Temporal and semantic analysis of richly typed social networks from user-generated content sites on the web

Zide Meng

UNIVERSITY COTE D'AZUR

DOCTORAL SCHOOL STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

P H D T H E S I S

to obtain the title of

PhD of Science

of the Université côte d'azur

Specialty : COMPUTER SCIENCE

Defended by

Zide MENG

Temporal and semantic analysis of richly typed social networks from user-generated content sites on the web

Thesis Advisor: Fabien GANDON and Catherine FARON-ZUCKER

prepared at INRIA Sophia Antipolis, WIMMICS Team

defended on Nov 07, 2016

Jury :

Reviewers :	Pr. Frédérique LAFOREST	-	Télécom Saint-Etienne
	Dr. John BRESLIN	-	National University of Ireland, Galway
Examiner :	Pr. Martin ARNAUD	-	Université de Rennes 1
Advisor :	Dr. Fabien GANDON	-	INRIA Sophia Antipolis
Co-Advisor:	Dr. Catherine FARON-ZUCKER	-	University Nice Sophia Antipolis
President :	Pr. Frederic PRECIOSO	-	University Nice Sophia Antipolis

Acknowledgments

I would like to express my sincerely thanks to my supervisors Fabien Gandon and Catherine Faron-Zucker for their great help, support, inspire and advice. I was very lucky to have them as my supervisors since they supported me not only on the research, but also on many aspects of my life.

I would also like to thank the rest of my thesis committee for their precious time, insightful comments and helpful suggestions. I could not finish my thesis without all the help.

I would like to thank the Octopus project (ANR-12-CORD-0026) for the financial support of my research. I would also like to thank our project partners for all the collaborations and meetings. It was a great pleasure to work with them.

I would also to thank the SMILK project for the financial support of my research and the chance of applying my work on additional real-world data set.

I would like to thank Stack Overflow, Flickr and Viseo for sharing their data which provided me with the chance to conduct my research project.

I would like to thank the Wimmics team. It is a friendly and international environment where I spent a great time with all my colleagues.

I would like to thank Christine Foggia for all her help and support.

I would like to thank my dear friends who always supported me and encouraged me. I also want to thank Sophie for her help and support. Especially, I would like to thank Fuqi Song for helping me to get over many difficult times. I am so lucky to have all of them.

I would like to thank shanshan for her love and support. I am happy and lucky to have her with me.

I would like to express my deep love and thanks to my family for their supporting and understanding whenever and wherever.

Abstract

We propose an approach to detect topics, overlapping communities of interest, expertise, trends and activities in user-generated content sites and in particular in question-answering forums such as StackOverflow. We first describe QASM (Question & Answer Social Media), a system based on social network analysis to manage the two main resources in question-answering sites: users and content. We also introduce the QASM vocabulary used to formalize both the level of interest and the expertise of users on topics. We then propose an efficient approach to detect communities of interest. It relies on another method to enrich questions with a more general tag when needed. We compared three detection methods on a dataset extracted from the popular Q&A site StackOverflow. Our method based on topic modeling and user membership assignment is shown to be much simpler and faster while preserving the quality of detection. We then propose an additional method to automatically generate a label for a detected topic by analyzing the meaning and links of its bag of words. We conduct a user study to compare different algorithms to choose a label. Finally we extend our probabilistic graphical model to jointly model topics, expertise, activities and trends. We performed experiments with real-world data to confirm the effectiveness of our joint model, studying user behaviors and topic dynamics.

Keywords:

social semantic web, social media mining, probabilistic graphical model, question answer sites, user-generated content, topic modeling, expertise detection, overlapping community detection

Résumé

Nous proposons une approche pour détecter les sujets, les communautés d'intérêt non disjointes, l'expertise, les tendances et les activités dans des sites où le contenu est généré par les utilisateurs et en particulier dans des forums de questions-réponses tels que StackOverFlow. Nous décrivons d'abord QASM (Questions & Réponses dans des médias sociaux), un système basé sur l'analyse de réseaux sociaux pour gérer les deux principales ressources d'un site de questions-réponses: les utilisateurs et le contenu. Nous présentons également le vocabulaire QASM utilisé pour formaliser à la fois le niveau d'intérêt et l'expertise des utilisateurs. Nous proposons ensuite une approche efficace pour détecter les communautés d'intérêts. Elle repose sur une autre méthode pour enrichir les questions avec un tag plus général en cas de besoin. Nous comparons trois méthodes de détection sur un jeu de données extrait du site populaire StackOverflow. Notre méthode basée sur le se révèle être beaucoup plus simple et plus rapide, tout en préservant la qualité de la détection. Nous proposons en complément une méthode pour générer automatiquement un label pour un sujet détecté en analysant le sens et les liens de ses mots-clefs. Nous menons alors une étude pour comparer différents algorithmes pour générer ce label. Enfin, nous étendons notre modèle de graphes probabilistes pour modéliser conjointement les sujets, l'expertise, les activités et les tendances. Nous le validons sur des données du monde réel pour confirmer l'efficacité de notre modèle intégrant les comportements des utilisateurs et la dynamique des sujets.

Mot-clés:

web social sémantique, l'analyse des médias sociaux, modèle graphique probabiliste, sites de questions-réponses, contenu généré par l'utilisateur, modélisation des thématiques, détection d'expertise, la détection de communautés recouvrantes

You can't connect the dots looking forward, you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future. –Jobs

Contents

CHAPTER 1

Introduction

Contents

1.1 Context: the rise of new content on the Web

One of the significant changes to the Web during the 2000s was a move from Web 1.0 to Web 2.0. A main attribute of Web 2.0 is that it allows users to interact and collaborate with each other in a social media platform as creators of user-generated content (Moens 2014) and members of (virtual) communities. In contrast, in Web 1.0 people were mostly limited to the passive viewing of content. Examples of Web 2.0 sites include social networking sites, blogs, forums, video, image or music sharing sites, etc. Web 2.0 does rely on this combination of contributing users and rich Web content. It is not limited to a network of relationships between users but is built on the common interests shared among users. Therefore, when analyzing Web 2.0 structures and activities, it is crucial to jointly study both users and user-generated content to really understand them. In other words, this analysis involves not only social network analysis (SNA) such as community detection or centrality calculation methods, but more generally social media mining techniques (e.g. topic detection from user-generated content). Also, users' behaviors and contributions are changing over time. Therefore it is also important to consider a temporal dimension when performing such an analysis.

In parallel, the Web has also evolved from a Web of Documents to a Web augmented with data readily available to software and machines. Following the W3C definition the "Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries".[1] However, most user-generated content on the Web is unstructured and isolated except for some classical hyperlinks.

Apart from some pioneering initiatives (Breslin 2006) (Breslin 2007) (Mika 2004) (Erétéo 2009) most user-generated content does not benefit from Linked Data and the models and formalisms of the Semantic Web. We need new methods and models in order to bridge social semantics and formal semantics on the Web (Gandon 2013). In particular, it is essential to formalize this information and transform it into knowledge.

In this thesis, we propose a framework, which combines social network analysis, social media mining and Semantic Web technologies, to help manage user-generated content-based websites. Figure 1.1 shows an overview of the proposed framework discussed in this thesis.

1.2 Our Scenario: managing question-and-answer sites

The main motivating scenario for this framework and our research questions is the case of question-and-answer sites (Q&A sites), which is a very rich (in terms of valuable and useful knowledge) type of user-generated content (UGC) website. Q&A sites were initially created to allow users to direct questions to a community of experts. But since these exchanges are archived as Web pages they become user-generated Web content, as formulated questions with submitted answers and comments, and they can be viewed and

[1] https://www.w3.org/2001/sw/ (accessed Feb 2016)

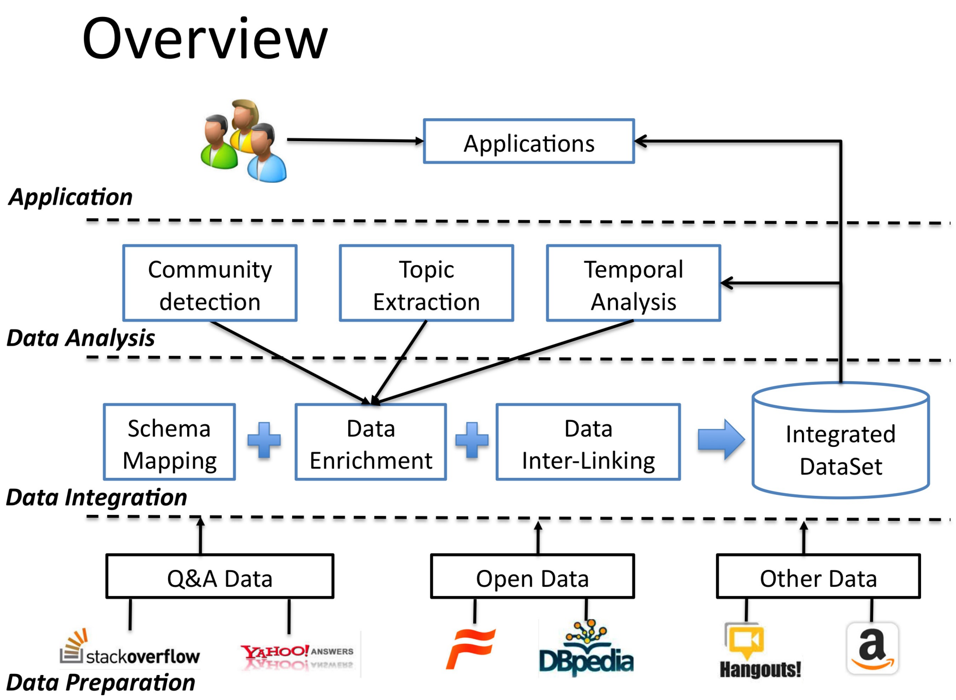


Figure 1.1: The overview of the framework proposed in this thesis to analyze Q&A sites content and communities

searched again later. People with the same or similar questions can find answers by browsing or searching the questions that were already answered. On one hand, Q&A sites have rapidly became huge repositories of question-answer content supporting highly valuable and highly reusable knowledge (Anderson 2012). On the other hand, Q&A sites also together a large number of users who keep contributing questions and answers. Most of these users are more likely to ask questions on topics they are interested in, and to answer questions on topics they are experts on. So in addition to hosting a semi-structured content network, Q&A sites have an implicit social structure and this is why Q&A sites are particularly illustrative of the need to jointly study both users' social structures and user-generated content as two sides of the same coin. Q&A sites are also known as Community Question Answering (CQA) sites, indicating the combination of the two key features of Q&A sites: a community (the users) and the questions and answers (the content).

Tags and folksonomies (for gathering and organizing tags) are quite common features in social networks, e.g. in Twitter [2], del.icio.us [3], Flickr [4], and also in some Q&A sites such as StackOverflow[5]. They are a special case of user-generated content and the activity of associating tags to content is known as collaborative tagging or social bookmarking. Tags enable users to classify and find resources via shared tags; they can help with creating communities, considering the fact that users sharing the same tags have common interests. Also, tags can directly reflect a users' vocabulary, and resources annotated with the same tags are often related to the same topics. Therefore, finding communities and topics from tags is a key question. We will more specifically focus on the analysis of tags associated with questions and answers in CQA sites.

Considering again the framework we propose, a first step is the design of schemas to formalize all of the meta-information we can export from a Q&A site. Second, the resulting dataset can be analyzed in three different ways: social structure analysis, content analysis and evolution analysis. Then the results of these analyse will be integrated with

[2] `https://twitter.com/` (accessed Feb 2016)
[3] `http://delicious.com/` (accessed Feb 2016)
[4] `https://www.flickr.com/` (accessed Feb 2016)
[5] `http://stackoverflow.com/` (accessed Feb 2016)

the original dataset to enrich its structure and support new usages. Third, based on this integrated dataset, we will provide several social applications, such as question recommendation, expert detection and user life-cycle management. This is the basic logic of the proposed framework and in this thesis we will focus on the export and analysis stages, and in particular on overlapping community detection, shared interest labelling and temporal analysis.

The reason why we conduct three kinds of analysis is because we believe that they address three needs linked to the two main resources in Q&A sites: the users' network and the Q&A content. Indeed, from a user's perspective, detecting communities of interests is useful to reveal the sub-structures of the user network and identify relevant peers. More precisely, obtaining this information can contribute to the question routing problem (Li 2010a)(Zhou 2012b), which is a very important Q&A sites optimization problem, for example, to forward a question to a user who is active in the corresponding topic and has the expertise needed to answer it. From the content's perspective, extracting topics is required to uncover the key subjects from massive amounts of content. It is extremely useful for instance to retrieve already posted answers to a re-submitted question. Moreover, both users and topics are changing over time, and therefore detecting such temporal dynamics is of prime importance to be aware of novelties. These indicators also are especially useful to community managers; they can also contribute to the community management, for instance by allowing one to track the interest evolution or community evolution in Q&A sites.

1.3 Research Question: topics, communities and trends in Q&A sites

In this section we summarize the main research questions that this thesis will address and answer.

RQ1. How can we formalize user-generated content?

The information in user-generated content is unstructured. The first issue is to formalize it. In addition, once an analysis has been performed, a second issue is to formalize the detected latent information and integrate it with the initial data in order to enrich it.

RQ2. How can we identify the common topics binding users together?

On user-generated content websites, users normally are creating information about their topics of interest. It is important to be able to detect these topics from the raw content generated by the users.

RQ3. How can we generate a semantic label for topics?

Until now in our research questions we have not characterized the representation of topics and in fact a topic consists mainly of a bag of words. One essential need is to automatically generate an adequate label for each topic to convey the meaning and coverage of the topic of shared interest it represents.

RQ4. How can we detect topic-based overlapping communities?

We address the problem of overlapping community detection. Unlike traditional graph-structure based methods, we try to solve this problem by relying directly on topic modeling. The advantage is that detected topics can be directly used to interpret the *raison d'être* of the communities. Another reason is that, regarding our scenario, Q&A sites support social networking, however, unlike networks such as Facebook, there are no explicit relationship-based links between their users. In fact, Q&A sites indirectly capture the connections made by users through the question-answer links or co-answer links. The users are neither mainly concerned with nor aware of the links existing between them. The social network is said to be implicit. Therefore, compared with other classical social networks, Q&A networks contain more *star-shape* structures (many users linked to a central user) than *triangle-shape* structures (users linked to each other). As a result, many community detection algorithms developed to discover sub-structures in social networks do not apply to Q&A implicit networks. Moreover, people may have multiple interests i.e. they may belong to several communities of interest. It is therefore important to be able to detect overlapping communities.

RQ5. How can we extract topics-based expertise and temporal dynamics?

The topics and the attention they attract change over time. We propose to address the problem of expert detection and temporal dynamics analysis together with topic modeling.

1.4 Contributions: models to identify shared interests and temporal dynamics

The major contributions of this thesis are as follows.

- To address research question **RQ1**, we designed a prototype system to formalize both implicit and explicit information in question and answer sites, to extract the implicit information from the original user-generated content, and to provide useful services by leveraging this detected information. Also, we proposed a vocabulary that can be used to formalize the detected information.

- To address research question **RQ2**, we present a topic tree distribution method to extract topics from tags. We also propose a first-tag enrichment method to enrich questions which only have one or two tags. We show the effectiveness and efficiency of our topic extraction method.

- To address research question **RQ3**, we propose and compare metrics and provide a method using DBpedia to generate an adequate label for a bag of words capturing a topic.

- To address research question **RQ4**, based on our topic extraction method, we present a method to assign users to different topics in order to detect overlapping communities of interest.

- To address research question **RQ5**, we present a joint model to extract topic-based expertise and temporal dynamics from user-generated content. We also propose a post-processing method to model user activity. Traditionally, this information has been modeled separately.

1.5 Thesis Outline: Social Semantic Web and CQA sites mining

This thesis contains a background and state of the art of related literature, an approach to detect topics from tags, an approach to detect overlapping communities and an approach to detect expertise and activities. The chapters in the rest of this thesis are organized as follows:

- Chapter 2 provides a background of related domains, and the state of the art on community detection, topic modeling, expert detection and temporal analysis. We identify the research trends in related areas, and outline the focus of this thesis.

- Chapter 3 describe QASM (Question & Answer Social Media), a system based on social network analysis (SNA) to manage the two main resources in CQA sites: users and contents. We also present the QASM vocabulary used to formalize both the level of interest and the expertise of users on topics.

- Chapter 4 describes an efficient approach for extracting data from Q&A sites in order to detect communities of interest. We also present a method to enrich questions with a more general tag when they only have one or two tags. We then compare three detection methods we applied on a dataset extracted from the popular Q&A site StackOverflow. Our method based on topic modeling and user membership assignment is shown to be much simpler and faster while preserving the quality of the detection.

- Chapter 5 describes an approach to automatically generate a label for a topic by analyzing the meaning and links contained in the bag of words. We conduct a user study to compare different algorithms to choose the label.

- Chapter 6 describes a probabilistic graphical model to jointly model topics, expertises, activities and trends for a question answering Web application. We performed experiments with real-world data to confirm the effectiveness of our joint model, studying the users' behavior and topics dynamics, again on the dataset extracted from the popular question-answer site StackOverflow.

- Chapter 7 summarizes our contributions and describes our perspectives on this work and future work.

1.6 Publications on the thesis contributions

The publications resulting from this thesis are the following:

- Journal

  1. Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker, Ge Song: Detecting topics and overlapping communities in question and answer sites. Social Network Analysis and Mining 5(1): 27:1-27:17 (2015)

  2. Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker: Overlapping Community Detection and Temporal Analysis on Q&A Sites. Web Intelligence and Agent Systems 2016.

- Conference Paper

  1. Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker: Joint model of topics, expertises, activities and trends for question answering Web applications. IEEE/WIC/ACM Web Intelligence 2016.

  2. Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker: Simplified detection and labeling of overlapping communities of interest in question-and-answer sites.

IEEE/WIC/ACM Web Intelligence 2015.

3. Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker, Ge Song: Empirical study on overlapping community detection in question and answer sites. IEEE/ACM ASONAM 2014: 344-348

4. Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker: QASM: a Q&A Social Media System Based on Social Semantic. International Semantic Web Conference (Posters & Demos) 2014: 333-336

CHAPTER 2

Background

Contents

2.1 Introduction

In this chapter, we review related topics for the background knowledge for this thesis and provide a state of the art review of related literature. First, we summerize background knowledge on social Web and semantic Web. Second, we provide an introduction to the collaborative project OCKTOPUS in which funded this Ph.D. We then discuss the state of the art approaches to community detection, topic modelling and temporal analysis in question-answering sites. We also detail classical tasks in the management of question-answering sites, and connected these tasks with our research questions. Finally, we define the focus of this thesis by identifying the research questions addressed and by positioning our contribution with regard to the state of the art.

2.2 Social Semantic Web: combine social network analysis and Semantic Web

2.2.1 Social Web: online communities and user-generated content

The term "social Web" was coined by Howard Rheingold in 1996. His Whole Earth Review article in 1987 introduced the notion of "Virtual Communities" and he was quoted in an article in Time magazine in 1996 introducing the term "Social Web". His website "Electric Minds", described as a "virtual community", listed online communities for users interested in socializing through the Web, stating that "the idea is that we will lead the transformation of the Web into a social Web" (Rheingold 2000). According to the World Wide Web Consortium (W3C), "the Social Web is a set of relationships that link together

Web 1.0		Web 2.0
DoubleClick	-->	Google AdSense
Ofoto	-->	Flickr
Akamai	-->	BitTorrent
mp3.com	-->	Napster
Britannica Online	-->	Wikipedia
personal websites	-->	blogging
evite	-->	upcoming.org and EVDB
domain name speculation	-->	search engine optimization
page views	-->	cost per click
screen scraping	-->	web services
publishing	-->	participation
content management systems	-->	wikis
directories (taxonomy)	-->	tagging ("folksonomy")
stickiness	-->	syndication

Figure 2.1: A comparison of examples of Web 1.0 and Web 2.0 sites, as in (O'really 2009)

people over the Web"[1]. The social Web is designed and developed to support social interaction (Porter 2010) on the Web. These on line social interactions include for instance online shopping, blogs, forums, video sharing and social networking websites. Today, hundreds of millions of people are using thousands of social websites to connect with friends, discover news and to share user-generated content, such as blogs, photos, microblogs, and videos. By the end quarter of 2008, Facebook reported 67 million members, YouTube had more than 100 million videos and 2.9 million user channels (Watson 2008), and these numbers are consistently growing, as today Facebook reports more than a billion active users.

2.2.1.1 Web 2.0

One of the significant changes for the World Wide Web was the move from the practices of Web 1.0 to the practices of Web 2.0. The term Web 2.0 was initially coined by Darcy DiNucci in 1999 (DiNucci 2012) and became popular through Tim O'Reilly in 2005 (O'really 2009). Web 2.0 techniques allowed users to interact and collaborate with each other and create user-generated content in online community sites, while users were mostly browsing content on Web 1.0 sites. A comparison of examples of traditional Web 1.0 sites and Web 2.0 sites is shown in Figure 2.1. Popular examples of Web 2.0 sites are Facebook (social networking service), Twitter (a microblog), YouTube (a video-sharing website), and Reddit (a user-generated news website).

[1] https://www.w3.org/2005/Incubator/socialweb/XGR-socialweb-20101206/(accessed Feb 2016)

With the evolution of web development technologies, such as Asynchronous JavaScript and XML (AJAX), Rich Internet Applications (RIA), Cascading Style Sheets (CSS), etc. Web 2.0 allowed users to create and share richly-typed user-generated content more easily. (Passant 2009a) argue that there are two main principles in Web 2.0, the first one is the *"Web as a platform"*, which implies the migration from traditional desktop applications (email clients, office suites, etc.) to Web-based applications. The second one is the *"architecture of participation"*, which represents how users change from data consumers to data producers in Web-based applications. For a more detailed description of the design principles of Web 2.0 websites, we refer the reader to (O'really 2009).

2.2.1.2 User-generated content

The OECD (Web 2007) considers that UGC applications have the following requirements: 1) content which is made publicly available through the Internet, 2) boasting a certain level of creativity and, maybe the most important point, 3) contents that are created outside of professional practices. UGC can be any form of content such as blog posts, photos, questions and answers, forum posts, tweets, videos, etc., created by users of online social media websites. (Moens 2014) The reasons why people contribute to user-generated content are many: connecting with people, self-expression and receiving recognition. For example: users connect with friends on Facebook; users express themselves on Twitter[2]; users share their photos on Flickr[3]; users ask and answer computer programming related questions on StackOverflow[4].

Nevertheless, there are some issues (Balasubramaniam 2009) with UGC, such as: the trust problem, since the content is written by non-professionals; the privacy problem, since the content often contains or reveals private information; the copyright problem, since more attention should be given to protecting rights in relation to user-generated content; etc. For more detailed information about the driving factors, the evolution of UGC, and the commercial influence of UGC, we refer readers to (Balasubramaniam 2009) and (Smith 2012).

[2] `https://twitter.com/`(accessed Feb 2016)
[3] `https://www.flickr.com/` (accessed Feb 2016)
[4] `http://stackoverflow.com/` (accessed Feb 2016)

2.2.1.3 Question-and-Answer (Q&A) sites

Question-and-Answer (Q&A) sites, also refered to as Community Question Answering (CQA) sites, were initially create to allow users to ask questions of a community of experts or, at least, a community of (shared) interest. Since this user-generated content, composed of questions and answers in this case, can be archived and later viewed and searched again, people with the same or similar questions can find answers by browsing or searching the questions that were already answered. For example[5], the first potential Q&A site Naver Knowledge Search [6] launched in 2002 in Korean, has accumulated 70 million questions and answers, and continues to receive over 40,000 questions and 110,000 answers per day (Sang-Hun 2007). Baidu Knows[7] and Zhihu [8] are the most popular Q&A sites in China. It is reported [9] that the number of registered users of Zhihu had exceeded 10 millions at the end of 2013, and reached 17 millions in May 2015 with 250 million page views monthly. Yahoo Answers, launched in 2005, offers Q&A sites localized in 26 countries and according to (Harper 2008) in September 2007 it was estimated to have 18 millions unique visitors monthly.

As the main means to access information on the Web are search engines, we compare the traditional keyword-based search engine to Q&A sites in terms of information retrieval tasks. In search engines, people choose some keywords to describe their problem, then look for related information in the result pages to solve their problems. In Q&A sites, people post their questions and wait for experts to solve them. Table 2.1 compares the two paradigms.

	Problem definition	Answer time	Results Precision	Problem Answers
Q&A	Well organized questions and background information	Until someone answers it	Specific to the question	Directly get the answers
Search Engine	Well chosen keywords or short question	Immediately get relevant information	Not specific to the question	Need to analyze the results

Table 2.1: Comparison of Q&A sites and Search Engines

[5] https://en.wikipedia.org/wiki/Comparison_of_Q&A_sites (accessed Feb 2016)
[6] http://naver.com (accessed Feb 2016)
[7] http://zhidao.baidu.com/ (accessed Feb 2016)
[8] http://www.zhihu.com/ (accessed Feb 2016)
[9] https://en.wikipedia.org/wiki/Zhihu (accessed Feb 2016)

In a Q&A site, people need to provide very detailed information about their questions, in order to let other users understand them. Providing additional details is even often asked by the experts in the first interactions. In a search engine, people have to wisely choose search keywords in order to look for solutions as the quality of keywords largely influences the results. When we pose a question to a Q&A site, it takes time to attract expert users and get the answers, but the search engine can immediately return relevant information. Once people get an answers from a Q&A site, normally it is very specific to the question and very precise. So a Q&A site can solve very complicated and precise questions. In search engines, people can get very relevant information about the keywords they provide but sometimes, the results are very general and not specific to the question. Users then have to find the solutions from the provided information by themselves. Beyond this comparison, it must also be stressed that as a Q&A site grows, providing an efficient search engine for its archive becomes a specific problem at the intersection of both paradigms. Moreover, a number of results found by major search engines come from Q&A Web archives.

On one hand, Q&A sites have become huge repositories of question-answer content which provide highly valuable and highly reusable knowledge (Anderson 2012), (Shah 2010). On the other hand, Q&A sites also contain a large number of users who keep contributing questions and answers. Also, most of them are more likely to ask questions on topics they are interested in and answer questions in topics they are experts on. This strong coupling of linked content and linked users is an aspect we will come back to.

Thus, we can consider this user-generated content is normally of high quality as it was generated by people with very strong domain knowledge and expertise. We list key features of some famous Q&A sites in Table 2.2. The column ’Category’ indicates the topics which are discussed in the websites. The column ’Reward’ indicates the rewarding system which is used to encourage users’ contribution. The column ’Tag’ indicates whether the website enables users to assign tags to questions. The column ’Vote’ indicates whether the website enables users to vote on questions, answers or both. The column ’Best Answer’ indicates whether the website enables users to choose a best answer.

	Category	Reward	Tag	Vote	Best Answer	Dataset availability
Yahoo Answer[a]	Multiple	Level and Points	no	answer	yes	Web access
StackOverflow[b]	Computer Programming	Reputation	yes	both	yes	Full access[c]
Baidu Zhidao[d]	Multiple	Level and Coins	no	both	yes	Web access
Zhihu[e]	Multiple	Vote and Like	yes	answer	yes	Web access
Quora[f]	Multiple	Views	no	answer	no	Web access

Table 2.2: Key features of famous Q&A sites

[a] https://answers.yahoo.com/ (accessed Feb 2016)
[b] http://stackoverflow.com/ (accessed Feb 2016)
[c] https://archive.org/details/stackexchange (accessed Feb 2016)
[d] http://zhidao.baidu.com/ (accessed Feb 2016)
[e] https://www.zhihu.com/ (accessed Feb 2016)
[f] https://www.quora.com/ (accessed Feb 2016)

StackOverflow is the most popular Q&A site that focus on computer programming topics. Its data is published every month. It includes all the detailed information, such as question answer contents, user profiles, temporal information. This is why we decided to use the StackOverflow dataset throughout this work. Figure 2.2 shows an example of a question and answer on StackOverflow[10]. Figure 2.3 shows the total number of visit since the StackOverflow.com is launched (data from [11]).

As already pointed, there are two main dimensions in Q&A sites, the coupling of which provide the power of these sites:

- Social dimension: A large number of people are very active and keep contributing answers to these sites. Most of them are more likely to answer questions about topics in which they are interested and specialized. Identifying interest groups of users in Q&A sites is an interesting indication of expertize in a Q&A site and community detection is a fundamental research topic for social network analysis. Many community detection algorithms have been developed to find sub-structures in social networks. Q&A sites are also social networks. However, unlike friendship networks such as Facebook, there are no explicit relationships between people on Q&A sites. Also, people are not aware of who they are interacting with, and normally they do not maintain a solid relationship. People are more like isolated nodes grouped by interests and the social network remains implicit. Therefore interest groups are an important implicit sub-structure to detect in such social sites. Moreover, people have multiple interests and therefore belong to several interest groups. Therefore an important aspect is the ability to detect overlapping communities of interest.

- Content dimension: Another important resource in Q&A sites are "question-answer" pairs. Questions cover different topics, and the fact that a user asks or answers a question can reflect the fact that he/she is interested in the topics touched by that question. Therefore, detecting topics of questions and identifying interests of groups

[10] http://stackoverflow.com/questions/3417760/how-to-sort-a-python-dict-by-value (accessed Feb 2016)

[11] https://www.quantcast.com/stackoverflow.com?qcLocale=en_US (accessed Nov 2016)

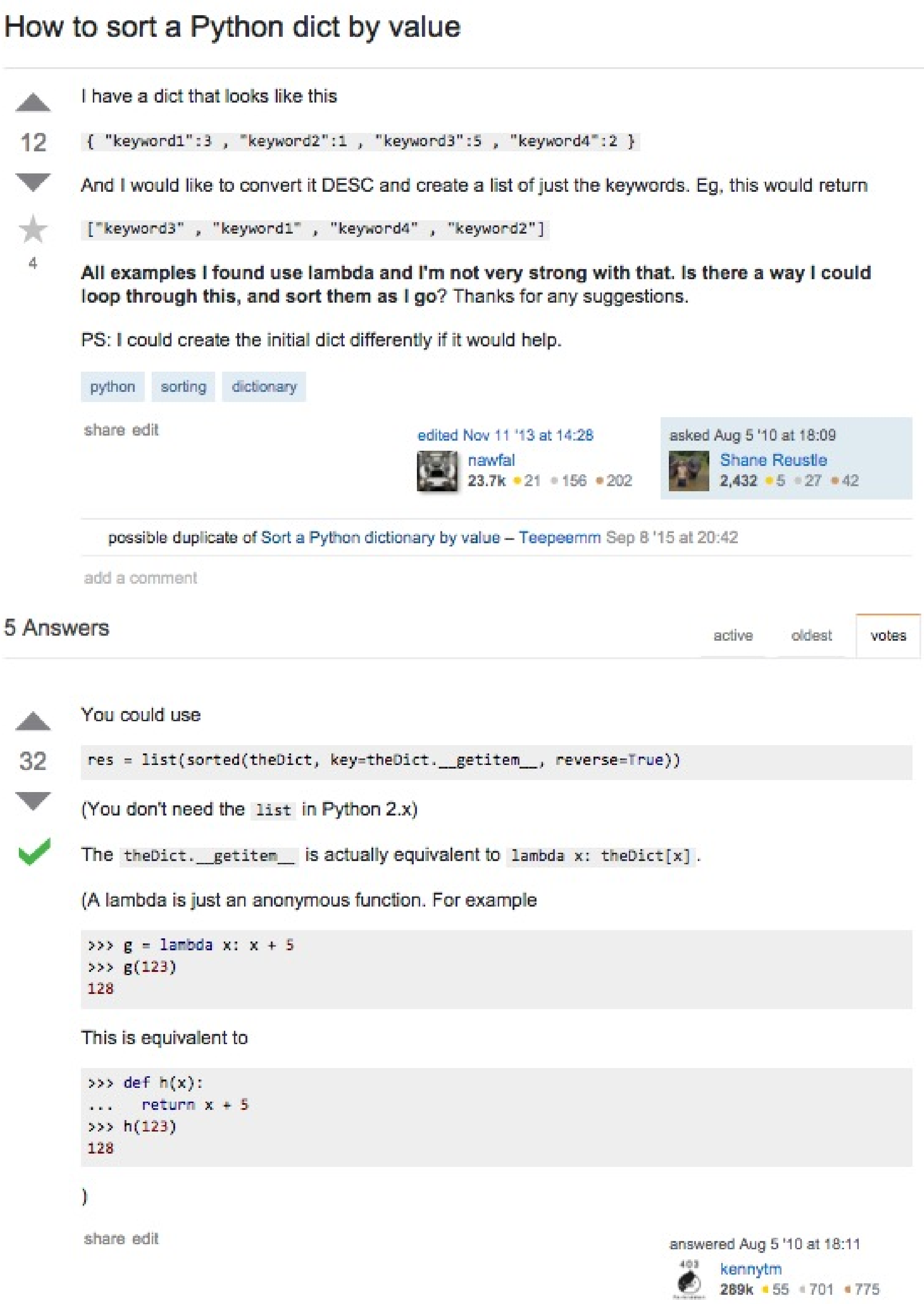


Figure 2.2: A example of question and answer on StackOverflow

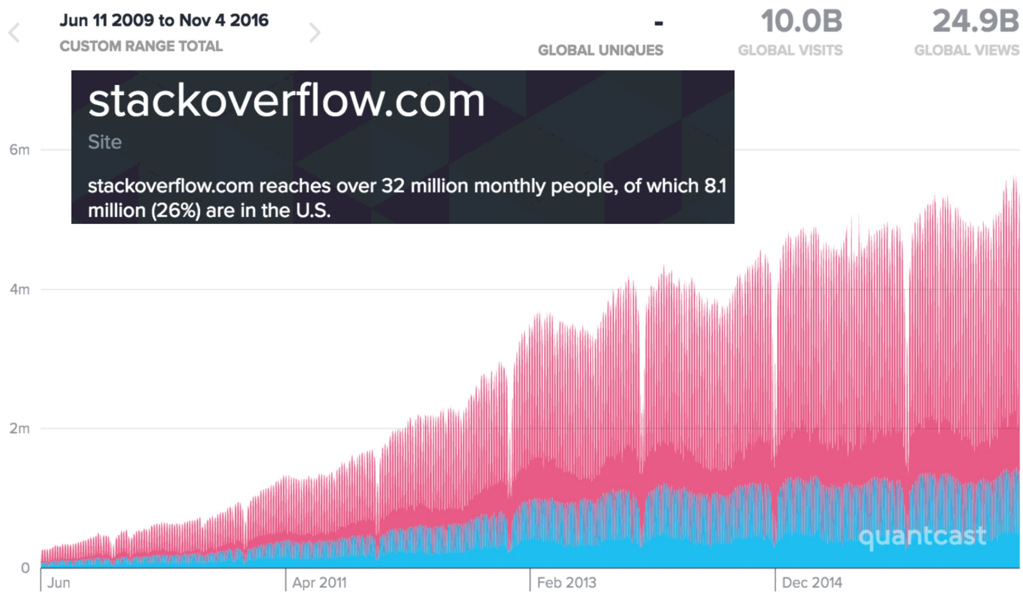


Figure 2.3: Total number of visit since the StackOverflow.com is launched

are related problems. We want not only to detect communities, but also to find their "raison d'être" i.e. to find the topic(s) of interest shared by each detected community. Topic extraction is a critical research problem in text analysis. Many topic extraction methods have been proposed to cluster textual resources by their topics. One of the reasons why we need such content analysis is that it enables systems, for instance, to use topics in recommending similar questions or in routing questions to experts, which are both very important functions in Q&A scenarios.

2.2.2 Semantic Web: formalizing and linking knowledge

According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries"[12] through the Web. Tim Berners-Lee (Berners-Lee 2001) also uses this term to refer to a Web of data that can be processed by machines. It is a change from a vision of a Web of documents to a Web that is also publishing and linking datasets. People generate and consume huge amounts of data every day. However, these data are kept in silos by each

[12] https://www.w3.org/2001/sw/ (accessed Feb 2016)

application or each website, and people have to manage and process the exchange of information by themselves. For example, in order to plan a trip, a user should check different websites to find information about a flight, hotel, weather, train schedule and so on. It is even not easy for a human to integrate them. For example, a small change of flight may cause the user to check and change all the other reservations. It is also not possible for applications to manage all these type of information from different websites. However, with a Web of data instead of a Web of documents, it becomes possible for applications to process and integrate data together. So, a key attribute of the Semantic Web is to enable content providers not only to publish human-readable Web documents, but also machine-readable data. With this vision, the Semantic Web allows applications to process data from different sources the same way people gather information from different Web pages. Later in 2006, Tim Berners-Lee (Berners-Lee 2006) proposed the Linked Data principles for publishing structured data on the Semantic Web. It is a method to share Semantic Web data using the Web architecture (Bizer 2011). An important development in this context is the W3C Linking Open Data (LOD) effort [13]. Figure 2.4 shows the LOD cloud diagram[14]. It shows the datasets that have been published as Linked Data. As of August 2014, the LOD cloud contains 1014 data sets classified into 8 domains while there are 520 datasets (corresponding to 51.28%) in the domain of Social Web and 48 datasets in user-generated contents (corresponding to 4.73%).

In the following subsections, we briefly introduce the RDF data model which is used to represent data on the Semantic Web and the related vocabularies to formalize social media datasets. For more details about the objectives and goals of the Semantic Web, we refer the readers to (Feigenbaum 2007) and (Berners-Lee 2001).

[13] https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData (accessed Feb 2016)

[14] Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

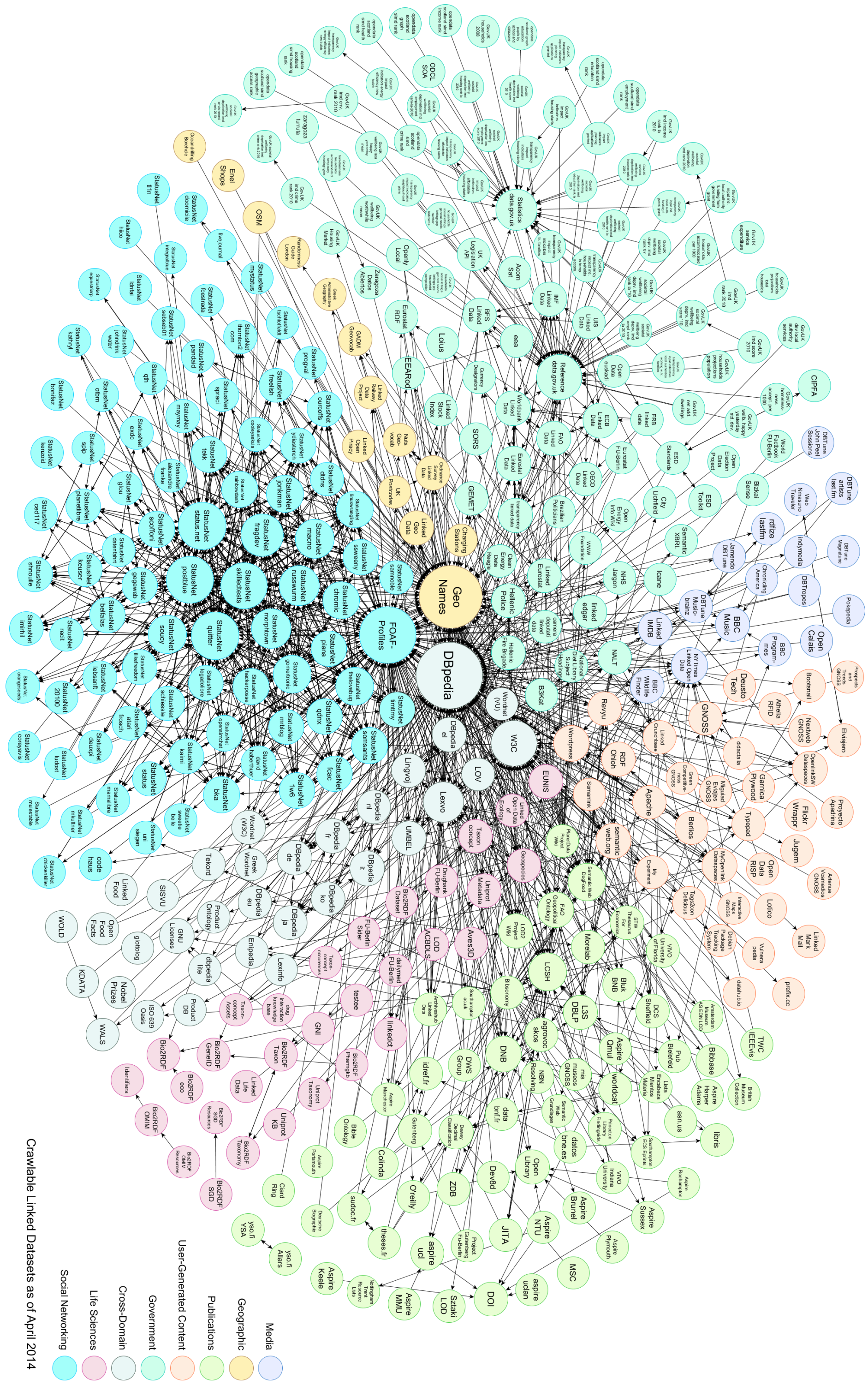


Figure 2.4: Linked Open Data cloud diagram.

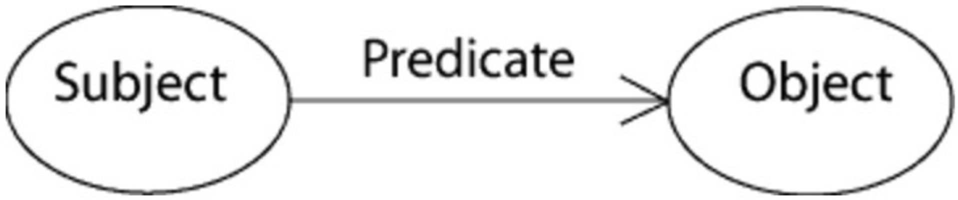


Figure 2.5: The triple as an arc in the graph data model of RDF.

2.2.2.1 RDF

The Resource Description Framework (RDF) data model is used to describe resources with the subject, the predicate and the object triple, which can be viewed as "a natural way to describe the vast majority of the data processed by machines". By considering RDF triples joined through shared URIs, one gets from a triple model to a graph data model[15], i.e., as show in Figure 2.5, each triple can be seen as a potentially distributed arc of an oriented labeled multi-graph.

The subject represents the described resource. The predicate represents the property used to describe the resource. The object represents the value of the property for the described resource. Any user can define and describe any resource with this model. For example, to formalize the fact that the user kingRauk is the owner of the question [16] from the Q&A site Stackoverflow, we can use a triple:

- whose subject <`http://stackoverflow.com/users/1214235/kingrauk`> is the URI that identifies the user who created the question,
- whose predicate <`http://rdfs.org/sioc/ns#owner_of`> is the URI that identifies the ownership property,
- whose object <`http://stackoverflow.com/questions/16772071/sort-dict-by-value-python`> is the URI that identifies the question.

[15] Resource Description Framework (RDF): Concepts and Abstract Syntax `https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-Concepts` (accessed Feb 2016)

[16] `http://stackoverflow.com/questions/16772071/sort-dict-by-value-python` (accessed Feb 2016)

Similarly, we can create another triple to formalize the fact that an answer is a reply to a question post:

- its subject <http://stackoverflow.com/questions/16772071/sort-dict-by-value-python/16772088#16772088> is the URI that identifies the answer to the question,
- its predicate <http://rdfs.org/sioc/ns#reply_of> is the URI that identifies the property *reply of*,
- its object <http://stackoverflow.com/questions/16772071/sort-dict-by-value-python> is the URI that identifies that was replied to question.

Alternatively, we can also create another triple to formalize the same fact where we can find the predicate 'reply_of' is the inverse relation of 'has_reply'. With respectively:

- its subject <http://stackoverflow.com/questions/16772071/sort-dict-by-value-python> is the URI that identifies that was replied to question,
- its predicate <http://rdfs.org/sioc/ns#has_reply> is the URI that identifies the property *has reply*, the inverse relation of *reply of*,
- its object <http://stackoverflow.com/questions/16772071/sort-dict-by-value-python/16772088#16772088> is the URI that identifies the answer to the question.

2.2.2.2 RDFS and OWL

An ontology is "a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application" (Liu 2009).

RDF enables people to describe resources and RDF Schema (RDFS) provides the basic primitives to define properties and classes. From the definition given by the W3C[17], RDFS is a language to define RDF vocabularies in order to represent RDF data. RDFS primitives extend the basic RDF vocabulary; they mainly enable us to declare classes, properties, hierarchies of classes and hierarchies of properties, and to associate labels and comments in Natural Language to classes and properties. The Web Ontology Language (OWL) builds on top of RDFS and provides a language for defining ontologies which enable richer integration and interoperability of data among descriptive communities. From the definition given by the W3C[18], OWL is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. While RDFS primitives enable us to *declare* atomic classes and properties, OWL primitives enables us to *define* classes and properties.

2.2.2.3 Vocabularies used in this thesis

In this thesis we needed to represent users, posted questions and answers, communities, and topics. Thus it is necessary to define an ontology for specific domain knowledge. We list the related and popular vocabularies used in our work.

SIOC[19] refers to the *Semantically-Interlinked Online Communities* ontology, which provides the main concepts and properties to describe online communitiy sites, such as weblogs, forums, message boards, wikis. These websites contain huge amounts of valuable information and the SIOC ontology tries to solve the problem that online community sites are like islands without bridges connecting them. It uses Semantic Web technologies to describe both the structure and content information in these online communities. It also allows us to link information to related online communities. Fig 2.6 shows an overview of

[17] `https://www.w3.org/TR/rdf-schema/` (accessed Feb 2016)
[18] `https://www.w3.org/OWL/` (accessed Feb 2016)
[19] `https://www.w3.org/Submission/sioc-spec/` (accessed Feb 2016)

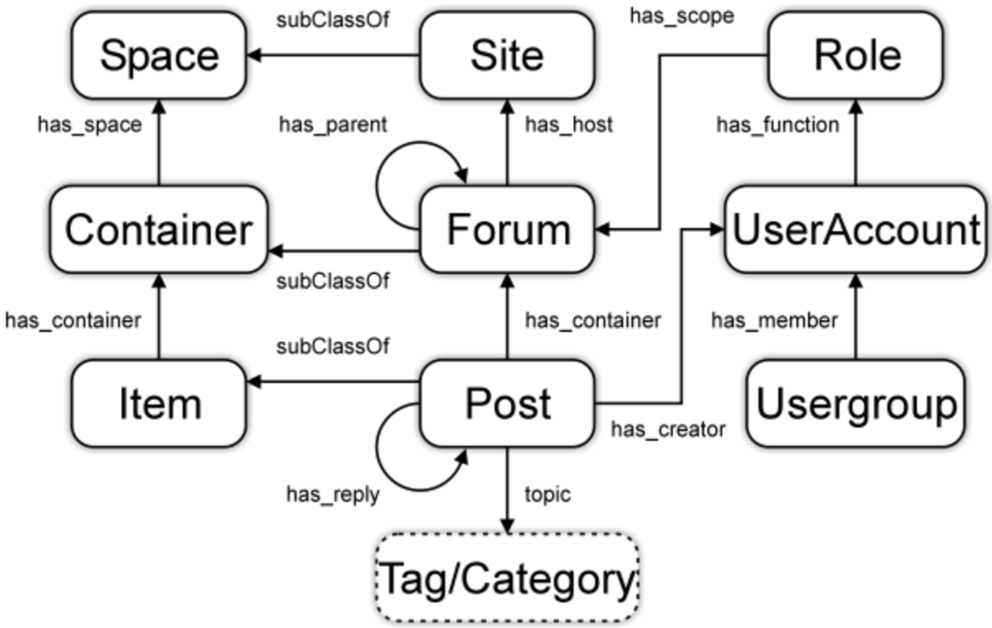


Figure 2.6: Overview of the SIOC ontology[20]

the SIOC ontology. It mainly formalizes community users and related activities in online communities.

The SIOC "UserAccount" primitive extends the OnlineAccount class from **FOAF**[21] ontology, which is another popular ontology to describe people and relationships between people. FOAF is a project devoted to link people and information using the Web. SIOC ontology mainly extends FOAF Core. It describes characteristics of people and social groups that are independent of time and technology. It includes classes such as 'OnlineAccount', 'OnlineGamingAccount' 'Organization', and 'Person'. Compared with SIOC, FOAF is not focusing on online communities and the user-generated content aspects.

Dublin Core[22] specification provides term definitions that focus on issues of resource discovery, document description and related concepts useful for cultural heritage and digital library applications. It is used to describe Web resources, such as Web pages, images, videos, but also physical resources such as CD, books. Dublin Core metadata may be used for multiple purposes, from simple resource description, to combining metadata vocabular-

[21] http://xmlns.com/foaf/spec/ (accessed Feb 2016)
[22] http://dublincore.org/documents/dcmi-terms/ (accessed Feb 2016)

ies of different metadata standards, to providing interoperability for metadata vocabularies in the Linked Data cloud and Semantic Web implementations. It is also not specific for online communities and user-generated content.

SKOS[23] stands for Simple Knowledge Organization Systems. It is a standard recommended by the W3C to formalize thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web. It enables the formalization of semantic relations between resources, such as 'narrower', 'broader' and related'. It also enables to describe concepts and labels which are often used in online communities. In our work on formalizing the latent information, which is beneath the data, we can reuse SKOS primitives to formalize a topic. Moreover, our work shows use cases that extend SKOS with new primitives enabling us to formalize to what extent a user is interested in a topic.

2.3 Context of the OCKTOPUS project: find the value of user-generated content

Over the past 15 years, along with the success of the Social Web, online communities have progressively produced massive amounts of user-generated content collaboratively. While some of these communities are highly structured and produce high-quality content (e.g., open-source software, Wikipedia), the level of discussions found within less structured forums remains highly variable. Coupled with their explosive growth, the lower quality of structure in online open forums makes it hard to retrieve relevant and valuable answers to users' search queries, and subsequently diminishes the social and economic value of this content.

The objective of the OCKTOPUS project[24] is to increase the potential social and economic benefit of this user-generated content, by transforming it into useful knowledge which can be shared and reused broadly. One of the targeted and easily-understandable outputs of the project is a demonstration platform which can be used to input a newly-

[23] https://www.w3.org/2004/02/skos/intro (accessed Feb 2016)
[24] https://alcmeon.com/ocktopus/ (accessed Feb 2016)

formulated question, search online forums for a similar already-answered question, and display a unique user-generated answer associated with these similar questions. This demonstration platform is built around the idea that finding relevant high-quality answers can be broken down into two steps:

- Triage user-generated content to extract gold (knowledge structured as pairs of questions and answers) from ore (random discussions).
- Given a newly-formulated question, retrieve relevant similar questions within the gold.

OCKTOPUS therefore investigates newer data mining techniques based on the proper assessment 1) of the organizational traits of online communities, 2) of the tree-structure of online discussions, and 3) of the temporal dynamics of large typed semantic user-user graphs to help improve the automatic classification and triage the unstructured online content.

2.4 Overlapping Community Detection

We distinguish between three kinds of approaches for community detection, depending on their characteristics: Graph-based methods relying on the network structure; Clustering methods based on the similarity of user profiles; Probabilistic graphical models based on network structure and/or user profiles.

2.4.1 Graph-based Methods

A first and direct solution for detecting communities from UGC data is to extract an implicit network structure (such as a question-answer network, a co-answer network, etc.) from interaction traces to come down to a traditional community detection problem on social networks. Since intuitively, users are grouped by interests, and most of their interactions are based on shared interests, it is reasonable to induce a network structure from these interactions and then run community detection algorithms on the network. Many classical algorithms have been developed such as (Xie 2013)(Ahn 2010). There are many constraints when adopting these methods. First, they do not take into account node attributes nor link attributes. Take a co-answer network as an example, where nodes represent users and links represent users answering the same questions. In case two users are connected, these methods can only indicate that they have answered the same questions many times. They cannot provide information on whether they have answered questions on the same topic or on different topics. Second, some of the works adopting this approach cannot detect overlapping communities, while other works such as (Xie 2013) address this problem.

2.4.2 Clustering Methods

Community detection can also be envisioned as a clustering problem. By computing similarities between user profiles, one can detect communities according to clustering results. The choice of which similarity metrics to use is quite important and influences clustering results. To find similar interests, we first have to define the distance between users' interests and the definition of this distance has a strong influence on the clustering results. For instance, we can consider a bag of weighted tags to represent an interest, then compute the weighted tag distance to define the interest distance between two users. Clustering methods, such as (Xu 2012)(Gargi 2011), group users according to their features. They do not take the network structure into consideration. Moreover, some clustering algorithms normally output hard-partition communities i.e. one user can only be assigned to one community. However, in the scenario we are interested in, a user often has more than one interest and should be assigned to more than one group simultaneously. This is a constraint for those hard-partition algorithms. (Chang 2013) uses spectral clustering to detect topics from the graph of tag co-occurrences. Compared to this, our approach is more efficient since we only run spectral clustering on a co-occurrence graph of selected tags (only 10% of all the tags). Also, (Chang 2013) does not give any details on how to compute the topic tag distribution and user topic distribution, while we use MLE(Maximum likelihood estimation) to compute the topic tag distribution.

2.4.3 Probabilistic Graphical Models

A third approach consists of using a probabilistic graphical model for both the user profiles and the network structure to solve the community detection problem. For example, (Zhang 2007a) transform links to binary node attributes, then uses a Latent Dirichlet Allocation (LDA) model to detect communities. (Sun 2013) uses a LDA-based method on social tagging systems where users label resources with tags, but they do not consider the problem of overlapping community detection. (Tang 2008) uses an extended LDA-based model to analyze academic social networks in order to find expert authors, papers and conferences. A problem with these LDA-based models is that they normally assume soft-membership (Yang 2013a) which means that a user cannot have a high probability of belonging to several communities simultaneously. That is to say, the more communities a user belongs to, the less it belongs to each community (simply because probabilities have to sum to one). Moreover, (McDaid 2010) and (Lancichinetti 2011) also use a statistic model to detect overlapping communities. The difference is that LDA-based models normally integrate topic detection which can be used to interpret detected communities while the two above cited methods only detect overlapping communities without any topic information on each detected communities.

2.4.4 Discussion on community detection alternatives

Table 2.3 summarizes the main features of the three approaches. The columns 'nodes' and 'links' indicate whether each method uses this information. The column 'overlap' indicates whether a user can belong to different communities i.e. if the approach detects overlapping communities. The column 'membership' indicates if the method provides a measure of "how much one user belongs to one community". The column 'topic' indicates if the method generates a bag of words to represent a topic, which can be used to explain the main aspects of contents generated by the users in the community.

Graph-based approaches normally use link information while ignoring node attributes. Some of them cannot detect overlapping communities or provide membership ratios which are weights denoting to what extent a user belongs to a community. Most of these methods cannot identify the topic in each detected community. Clustering approaches use node attributes to group similar users. Some of their results are hard-partition communities, with no overlapping and no membership information. LDA-based models overcome the shortcomings of graph-based and clustering approaches, using both node attributes and link information. Also, LDA-based models normally combine community detection with topic detection, which could be used to interpret detected communities. Our proposed method is similar to LDA-based methods, in that it also enables us to detect overlapping communities and identify the topics at the same time. It differs from LDA-based methods in that it enables us to consider a user that has a high probability of belonging to several communities simultaneously while these methods normally assume soft membership (Yang 2013a). In addition, our proposed method is much simpler and faster than LDA-based methods while preserving the quality of the detection. For more details about community detection algorithms in graphs, we refer readers to (Fortunato 2010) and (Xie 2013).

Table 2.3: Comparison of the main approaches and our method

	nodes	links	overlap	membership	topic
Graph-based methods	no	yes	few	few	no
Clustering methods	yes	no	few	few	no
Probabilistic graphical model	yes	yes	yes	yes	yes

2.5 Topic Modeling: Uncover the Hidden Thematic Structure

According to David M. Blei[25], "Topic models are a suite of algorithms that uncover the hidden thematic structure in document collections. These algorithms help us develop new ways to search, browse and summarize large archives of texts." For example, "guitar" and "music" will appear more often in documents about music, "law" and "lawsuit" will appear more often in documents about legal matters, and "the" and "is" will appear equally in both documents. A document normally contains multiple different topics in different propor-

[25] https://www.cs.princeton.edu/~blei/topicmodeling.html (accessed Feb 2016)

tions. For example, a document on music copyright lawsuit, could be 30% about music and 70% about legal matters.

Latent Semantic Analysis (LSA or LSI) (Deerwester 1990) (Landauer 1997) is an early topic model based on the factorization of document-word occurrence matrix. By using singular value decomposition (SVD), it can find a linear combination of topics for each document. Probabilistic Latent Semantic Analysis (PLSA), also known as Probabilistic Latent Semantic Indexing (PLSI) (Papadimitriou 1998)(Hofmann 1999b) is a generative statistic model to estimate a low-dimensional representation of the observed variables. Latent Dirichlet Allocation (LDA) (Blei 2003) is also a generative statistic model that uses observed variables to explain unobserved latent variables, which is a generalization of PLSI model. (Griffiths 2004) use Gibbs sampling to infer the latent variables in the LDA model and introduce some applications of LDA model. Many other topic models are extensions of the LDA model. For example, Hierarchical Latent Dirichlet Allocation (HLDA) (Thrun 2004) is a topic model that finds a hierarchy of topics. The structure of the hierarchy is determined by the data. Dynamic topic models (DTM) (Blei 2006b) discover topics that change over time and how individual documents predict that change. Correlated Topic Models (CTM) (Blei 2006a) discover correlation structures between topics, etc.

Topic modeling is an active field in text mining and machine learning and we refer the readers to (Blei 2012) for a high level view and summary of the topic modeling research area and also for several exciting future research directions. One of them deals with the development of evaluation methods, *"How can we compare topic models based on how interpretable they are?"*.

Another interesting research problem related to topic modeling is *how to automatically label the generated topics?* (Cano Basave 2014) (Hulpus 2013) (Aletras 2014) (Sun 2015) (Lau 2011) Typically, users of topic modeling approaches have to interpret the results and manually generate labels for topics for further processing, classification, visualization or analysis. Therefore, in this context, "labelling" means the problem of finding one or more phrases, or concepts, which can sufficiently cover, represent or describe the topic. The problem then is defined as the automation of the topic labelling.

2.6 Temporal Analysis: integrate temporal analysis within topic modeling

There is an increasing research interest into the temporal modeling of online communities and several methods have been proposed.

(Wang 2006) introduced *Topic Over Time* (TOT), which jointly models topics and timestamps by assuming that words and timestamps are both generated by latent topics. Therefore, the parameter estimation is able to discover topics that simultaneously capture word-word co-occurrences and word-timestamps co-occurrences. If some words co-occur for a short period, their approach will create a topic with a narrow time distribution. If some words co-occur over a long time, their approach will create a topic with broad time distribution. The novelty of TOT is that it treats time as an observed continuous variable rather than a Markov process. Besides, the meaning of topics remains constant while the topic themselves change over time. (Blei 2006b) proposed a dynamic topic model that treats the temporal dimension as a Markov process where the meaning of topics changes over time. (AlSumait 2008) also studied topic changes over time, but they focus on proposing an online method to extract topics from a stream of data. (Wang 2007) address the problem of mining correlated bursty topic patterns from coordinated text streams (e.g. the same news in different media or in different languages). They proposed a mixture model which is an extension of the PLSA (Hofmann 1999a) model to detect topic evolution from text streams by comparing topics in consecutive time intervals. (Yao 2010) and (Yao 2012) proposed a sliding window and graph partition based approach to detect a bursty event/topic in tags. (Diao 2012) proposed a TimeUserLDA model to find a bursty topic from microblogs. It considers both user personal topic trends and global topic trends and detects bursty topics from the extracted topics over time distribution. (Yin 2013) proposed a PLSA-based (Hofmann 1999a) model to separate temporary topics from stable topics. Temporal topics are based on popular real-life events, e.g. breaking news. It will lead to a burst in online community discussions with a large amount of user-generated content in a short time period. Stable topics are often users' regular interests and daily routine discussions which always exist and do not evolve a lot in a long time period. (Hu 2014) jointly model latent user groups and temporal topics to detect group-level temporal topics.

Compared with these works, our model not only captures topics and expertise, it can also detect topic dynamics both at the global community level and at the individual user level. Also, we propose a post-processing method to extract both topic-time and time-topic distributions. The time-over-topic distribution are usually ignored.

2.7 Q&A Sites Management

2.7.1 Expert Detection: find the "core" user

Research related on expert identification in Q&A sites is mainly based on link analysis and topic modeling techniques. The general purpose of expert detection is normally to support the question routing task which essentially consists of finding the most relevant experts to answer a newly submitted question.

(Zhang 2007b) is not specific to Q&A communities and focuses on a broader website category: help-seeking websites. It tested the PageRank and HITS algorithms to detect experts in such websites. PageRank and HITS are well known authority algorithms in directed graph analysis. By constructing a directed graph of the users'network, they could apply these algorithms to find the most important node in the graph according to these centrality metrics. In addition, they proposed the Z-score measure to evaluate expertise. Compared with simple statistic measures, for instance the number of best answers provided by a user, the Z-score measure uses both the number of questions and the number of answers posted by a user. Similarly, (Jurczyk 2007) use the HITS algorithm to discover users that are authorities. (Li 2010a) propose a probability model to estimate users' expertise for a question routing task.

(Zhou 2012a) address a core problem in applying the previous techniques to Q&A sites. They argue that most previous works in expert finding are based on link analysis while ignoring the topical similarity among users and user expertise and user reputation. They proposed a topic-sensitive probabilistic model to find experts in Q&A sites. This model is based on LDA. Then they generate a topic-similar graph based on the result of the topic model. Finally a PageRank algorithm is applied to find the experts. They compared their work with many state of the art link analysis algorithms and showed gains in the experiment.

(Pal 2010) on the other hand proposed a temporal pattern-based expert detection method. The temporal pattern is based on the reputation system of Q&A sites where a user having a high reputation is considered as an expert. Their approach uses a supervised learning algorithm to distinguish experts from normal users. The limitation of this work is that it cannot find in which topics people are specialized.

(Bouguessa 2008) proposed a method using link analysis techniques to find a list of expert users based on the in-degree of authority, which is computed based on the number of best answers provided by a user. Then they use the Bayesian Information Criterion (BIC) to estimate the authority score of a user. Therefore, experts are chosen according to their authority score. Their expertiment was done on Yahoo Answers.

Rather than detecting global experts, another kind of work uses topic models to detect topic level experts. (Guo 2008b) proposed a generative model by leveraging the category information of questions on certain Q&A sites. (Yang 2013b) jointly model topics and expertise by integrating a Gaussian Mixture Model to capture vote information. (Chang 2013) propose a spectral clustering-based topic model. (Ma 2015) propose a generative model to model the triple role of users (as askers, answerers, and voters). Our contribution extends this line of work.

There are also approaches applying machine learning techniques to perform expert detection. (Ji 2013) combine topic models outputs and statistic features and apply a pair-wise learning to obtain a ranked model and recommend expert users for a question.

(Pal 2011) apply machine learning algorithms to identify experts from their early behavior. (Anderson 2012) perform an in-depth study of StackOverflow and show that expert users tend to answer questions more quickly and gain high reputation by higher activity. Their work is based on features extraction and machine learning algorithms to predict whether a question has a long-term value and whether a question has been sufficiently answered. Their results show that votes information can indicate a user's expertise level while currently, this kind of work normally relies on the outputs of topic models.

2.7.2 Question Routing: recommend new questions to users

(Guo 2008a) try to solve the question routing problem, which we categorized as RQ5. They proposed an LDA-like probability model to find the latent topic of users and the latent topic of questions and answers. Then based on this topic information, they can route a new question to a user which has the same topic distribution. (Yang 2013b) proposed a Topic Expertise Model which is also an LDA-like probability model but combined with a Gauss Mixture Model (GMM) model to detect experts in Q&A sites. The probability model is mainly used for extracting topics from tags and words in Q&A and it contains two LDA processes: 'user-topic-tag' and 'user-topic-content'. The GMM is used for analyzing user's expertise on each topics. The output includes topic-tag distribution, user-topic distribution, topic-word distribution and the users' topic-expertise matrix. Then according to these outputs, they can identify the top tags for a topic, top users for a topic and top experts for a topic. The experiments show they can outperform the state-of-the-art probability model in Q&A sites.

(Chang 2013) proposed a recommendation model, which integrates topical expertise and availability of users, to recommend reactive answerers and commenters for a question. It constructs a similarity matrix between tags, and runs a spectral clustering algorithm over it. Then a cluster of tags can be viewed as a topic. But unlike LDA, spectral clustering can not output the topic-tag distribution which will limit the flexibility of subsequent applications. The paper proposed a question-topic distribution, but it dose not mention how to compute it. Therefore the conclusion that spectral clustering can out perform LDA is not clear. Spectral clustering is a hard partition of tags, while LDA can give probabilities that a tag belong to several topics.

2.7.3 Similar Question: find questions which have been answered

(Anderson 2012) investigates the general characteristics of the StackOverflow dataset. A contribution of this work is to predict the long-term value of a question. They find strong evidence that only 37% of the best answers for a question arrive within the time frame when the question is being answered. Actually the content in Q&A sites mainly serves two kinds of people: the people who ask questions and the people who search through previous questions. So, if a question has a long-term value, it is more likely to be searched for again. Finding out these questions could improve the results of searching for similar questions. They developed four categories of features for learning. They include: 4 questioner features which are related to questioner's behaviors; 8 activity and Q&A quality measures which are extracted from questions and answers; 8 community process features which are related to the reputation of answers; and 7 temporal process features which are generated from the time information of the Q&A activity. Then they treat this problem as a binary classification task and use a machine learning technique to predict whether a question has a long-term value. They compare their work with a baseline which only uses upvote and downvote features.

(Jeon 2005) discuss methods for question retrieval that are based on using the similarity between answers. It proposed a translation-based retrieval model to find similar questions. The experiment shows that it is possible to find semantically similar questions with relatively few overlapping words. They found that question titles can provide the best performance for retrieving a similar question. This work is based on the intuition that most of the people do not check whether their question has been already asked which leads to a situation where there can be many semantically identical questions. Therefore, they use the similarity between answers to group similar questions. A translation model based algorithm is proposed to calculate the similarity between answers. For example, this model can provide a similarity score between 'bmp' and 'jpg'. Experiments show that the model can outperform other language models and similarity metrics.

(Qu 2009) presents a probabilistic latent semantic analysis (PLSA) approach to compute the probability that a user will answer a question. They actually build a user-interest-question model. PLSA and LDA are quite similar and both are topic models, and LDA could be viewed as an extension of PLSA. The experiment shows that topic features-based similarity can outperform cosine distance-based similarity. (Wu 2008) also used PLSA to recommend questions.

Table 2.4 provides a comparison of the above described works.

2.8 Research Questions: the focus of this thesis

In this section we summarize the research questions we will address in this thesis and we position our contribution for each of them.

2.8.1 How can we formalize user-generated content?

Compared to state of the art approaches, we use social media mining techniques to extract topical dynamic, topical activity, topics and topical expertise from user-generated content. Then we integrate this extracted information into the original dataset in order to provide more functionality for further use. We will detail this work in Chapter 3.

2.8.2 How can we identify the common topics binding users together?

Compared to the state of the art approaches, we focus on the simplicity and efficient aspects of the proposed method. Based on a prefix-tree structure, our method can also extract sub topics from a topic. We detail this work in Chapter 5.

	Expert	Routing	SimilarQ	Method	Dataset	Topic
(Jeon 2005) 2005	no	no	yes	Probability Model	Naver[a]	no
(Zhang 2007b) 2007	yes	no	no	PageRank,HITS	Forum	no
(Guo 2008a) 2008	no	yes	no	LDA based	StackOverflow	yes
(Qu 2009) 2008,(Wu 2008)2008	no	yes	yes	PLSA	Yahoo,Wenda	yes
(Bouguessa 2008) 2008	yes	no	no	Link Analysis	Yahoo	no
(Pal 2010) 2010	yes	no	no	Supervised Learning	StackOverflow	no
(Anderson 2012) 2012	no	no	yes	Supervised Learning	StackOverflow	no
(Zhou 2012a) 2012	yes	no	no	LDA based	StackOverflow	yes
(Yang 2013b) 2013	yes	yes	yes	LDA based	StackOverflow	yes
(Chang 2013)2013	yes	yes	no	Spectral Clustering	StackOverflow	yes

Table 2.4: Comparison of several works in Q&A sites. ’Expert’ denotes ’Expert Detection’, ’Routing’ denotes ’Question Routing’, ’SimilarQ’ denotes ’Similar Question Finding’, ’Method’ denotes ’Proposed Algorithm’, ’Dataset’ denotes ’Experiment Data’, and ’Topic’ denotes ’Topic Detection’

[a] A leading Q&A sites in South Korea

	(Omitola 2015)	(Passant 2009b)	(Plumbaum 2015)	our work
Social media mining	yes	yes	no	yes
User behaviour modeling	yes	yes	yes	yes
User interesting modeling	no	yes	no	yes
User activity modeling	no	no	no	yes
User expertise modeling	no	no	no	yes
Topic based modeling	yes	no	no	yes

Table 2.5: Position of our work with regard to the first research question

	(Blei 2003)	(Chang 2013)	(Yang 2013b)	(Hu 2014)	our work
Model	PGM	SC	PGM	PGM	SC
Simplicity	no	yes	no	no	yes
Sub-topic	no	no	no	no	yes
Iterations	yes	no	yes	yes	no

Table 2.6: Position of our work with regard to the second research question, *PGM*: Probabilistic Graphical Model, *SC*: Spectral Clustering

	(Sun 2015)	(Hulpus 2013)	(Cano Basave 2014)	(Aletras 2014)	our work
Extra information	Probase[26]	DBpedia	no	Bing[27] results	DBpedia
Method	MDL	DR Graph	S	Words Graph	DR Graph
User study	no	yes	no	no	yes
Label type	word	DR	word	words	DR

Table 2.7: Position of our work with regard to the third research question, *MDL*: Minimum Description Length, *DR*: DBpedia Resource, *S*: Summarization Algorithm

2.8.3 How can we generate a semantic label for topics?

Compared to the state of the art approaches, we focus on extending our topic extraction method and on using DBpedia resources to automatically generate labels for bags of words composing topics. We also compare several graph centrality metrics to generate labels. We describe this work in Chapter 6.

2.8.4 How can we detect topic-based overlapping communities?

	(Raghavan 2007)	(Xie 2013)	(Girvan 2002)	(Yang 2013a)	(Hu 2014)	our work
Method	LPA	LPA	HC	PGM	PGM	SC
Info	Graph	Graph	Content	Graph, Content	Graph, Content	Graph, Content
Interpret	no	no	no	yes	yes	yes
Overlapping	no	yes	no	yes	yes	yes
Simplicty	yes	yes	yes	no	no	yes

Table 2.8: Position of our work with regard to the fourth research question, *LPA*: Label Propagation Algorithm *PGM*: Probabilistic Graphical Model, *HC*: Hierarchical Clustering *SC*: Spectral Clustering

Compared to the state of the art approaches, we focus on extending topic extraction methods to effectively detect overlapping communities. We describe this work in Chapter 5.

2.8.5 How can we extract topics-based expertise and temporal dynamics?

Compared to the state of the art approaches, we integrate topic dynamics, users' activity and topic-based expertise extraction together to solve several tasks related to Q&A site. We describe this work in Chapter 7.

	(Yang 2013b)	(Chang 2013)	(Guo 2008b)	(Blei 2003)	(Hu 2014)	(Diao 2012)	our work
Model	PGM	SC	PGM	PGM	PGM	PGM	PGM
Topic Dynamic	no	no	no	no	GL	GL,UL	GL, UL
Expertise	yes	yes	no	no	no	no	yes
User Activity	no	non-topical	no	no	topical	topical	topical

Table 2.9: Position of our work with regard to the fifth research question, *PGM*: Probabilistic Graphical Model, *SC*: Spectral Clustering, *GL*: Global Level, *UL*: User Level

CHAPTER 3

QASM: Question and Answer Social Media

Contents

3.1 Introduction: formalizing and linking knowledge on Q&A sites

Community Question Answering (CQA) services provide a platform where users can ask experts for help. Since questions and answers can be viewed and discussed and all these traces can be searched afterwards, Q&A sites form a special kind of social media.

In order to make the data of a social website available on the semantic Web we have to perform two steps:

- **extracting and formalizing:** to choose or provide suitable vocabularies or extensions to represent the social media data (content, users, interactions, etc.) and provide the extraction mechanism to produce the Semantic Web representation from the native structures and APIs of the social media platform. This is what we address in this chapter.

- **linking:** to weave a Web of data and allow the extracted data to be fully linked to other sources on the Web of data who will benefit from this enrichment and thereby contributing to the creation of new pathways in the linked data. This will be covered in Chapter 6.

We also differentiate between two kinds of information in our scenario.

- **Information explicitly generated:** this is the original user-generated content, for instance, a question, an answer, a comment, a tag, etc.

- **Information implicitly generated:** this information is generated as a side effect of the activity on the site. This is latent information extracted by data mining techniques, for instance, implicit social networks, detected community, traces and logs temporal information, etc.

It is important to formalize both kinds of information and to link the obtained representations in order to benefit from both aspects in the analysis. Among the available vocabularies (e.g. in the LOV directory) the SIOC[1] ontology is the most popular vocabulary to formalize social media, but it does not support the formalization of the latent information extracted by data mining techniques.

In this chapter, we propose the QASM (Question & Answer Social Media) vocabulary. We reuse existing vocabularies such as SIOC and FOAF[2] and extend them with the primitives needed to formalize explicit and implicit QA social media.

3.2 Overview of our modeling approach

Figure 3.1 presents an overview of QASM. We first use the SIOC ontology[3] to construct an RDF dataset from social media data extracted from a CQA site, namely StackOverflow.

[1] http://lov.okfn.org/dataset/lov/vocabs/sioc (accessed Aug 2016)
[2] http://lov.okfn.org/dataset/lov/vocabs/foaf (accessed Aug 2016)
[3] http://sioc-project.org/ontology (accessed Aug 2016)

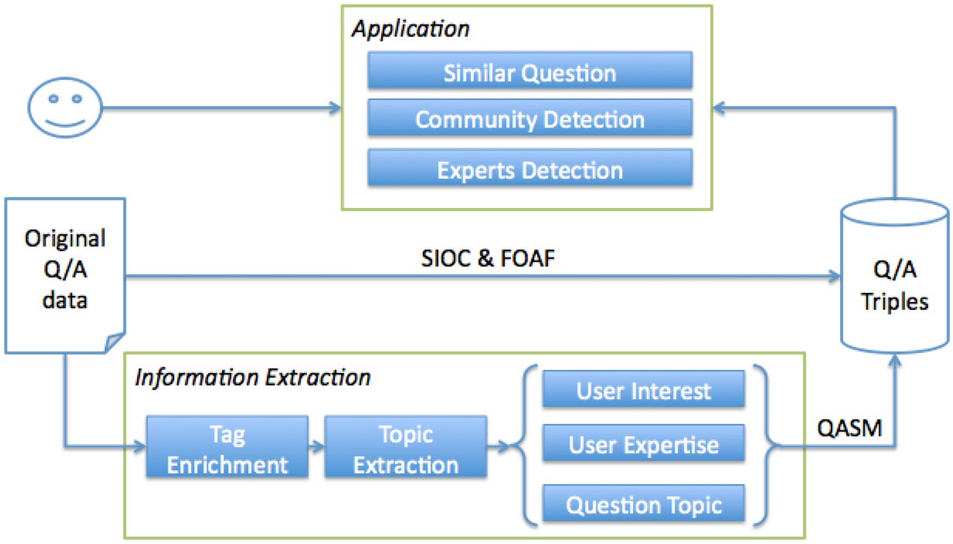


Figure 3.1: Overview of QASM

Then we use social media mining techniques to extract topics, interests and expertise levels and temporal dynamics from this dataset. We formalize them with the QASM vocabulary and enrich our RDF dataset with this latent information. As a result, we provide an integrated and enriched Q&A triple store which contains both user interests, user expertise, and temporal dynamics of users' profiles and of topics. Then, we link our dataset with DBpedia and use the resulting knowledge graph to generate labels for topics. Finally, based on the QASM RDF dataset, we can provide the users of the Q&A site with several services to find relevant experts for a question and to search for similar questions.

3.3 QASM Vocabulary: formalize Q&A information

As explained in the introduction, there are mainly two kinds of information to formalize. Part of it is explicit: the original user-generated content, such as Q&A contents, user profile, votes, and timestamps. Part of it is implicit and extracted by social media mining techniques: user interests, overlapping communities, user expertise, user activities.

Existing work mainly focuses on how to formalize the explicit information in Q&A sites. We are focusing on extending existing work to also formalize the implicit informa-

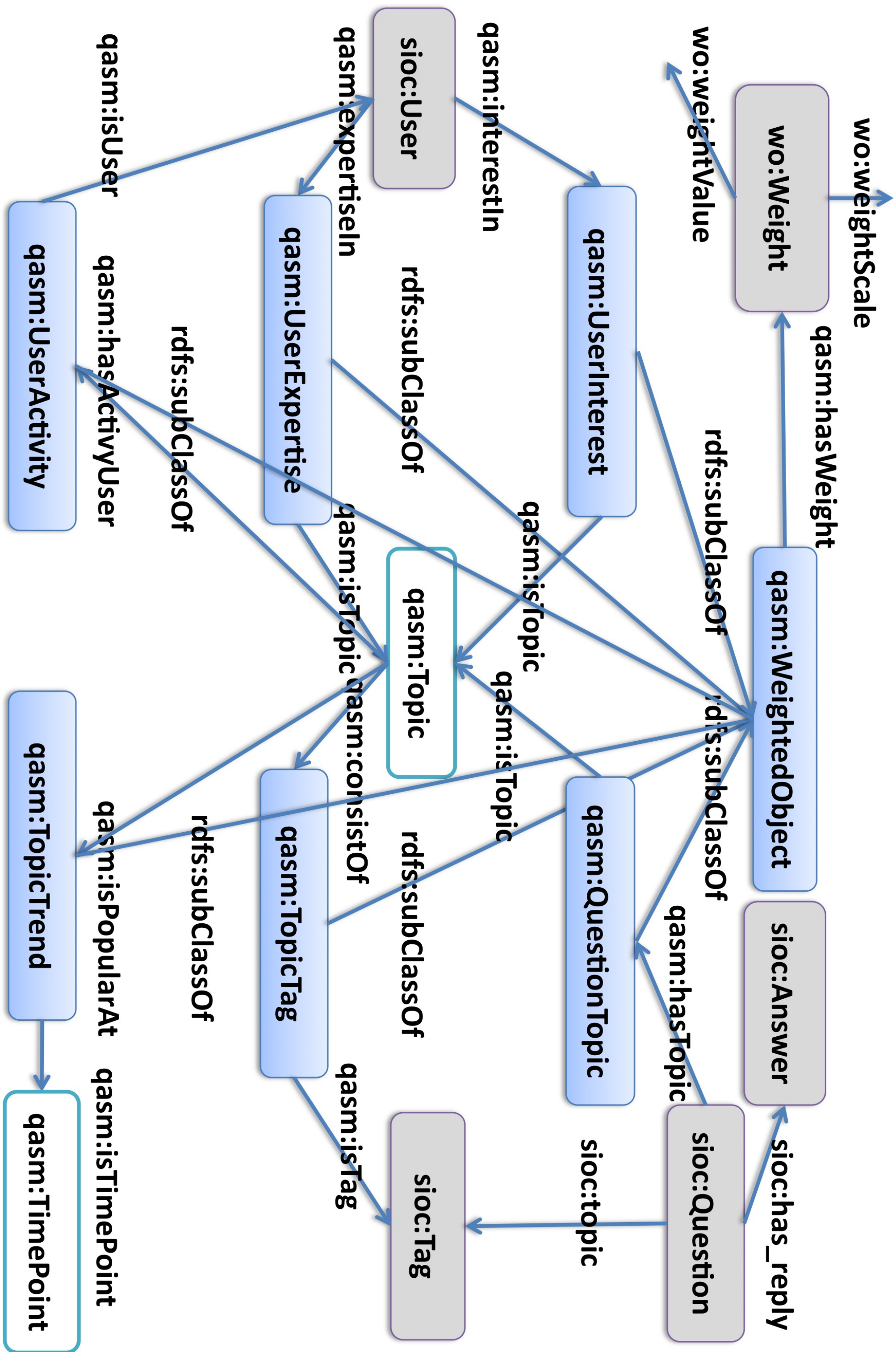


Figure 3.2: Overview of the QASM vocabulary

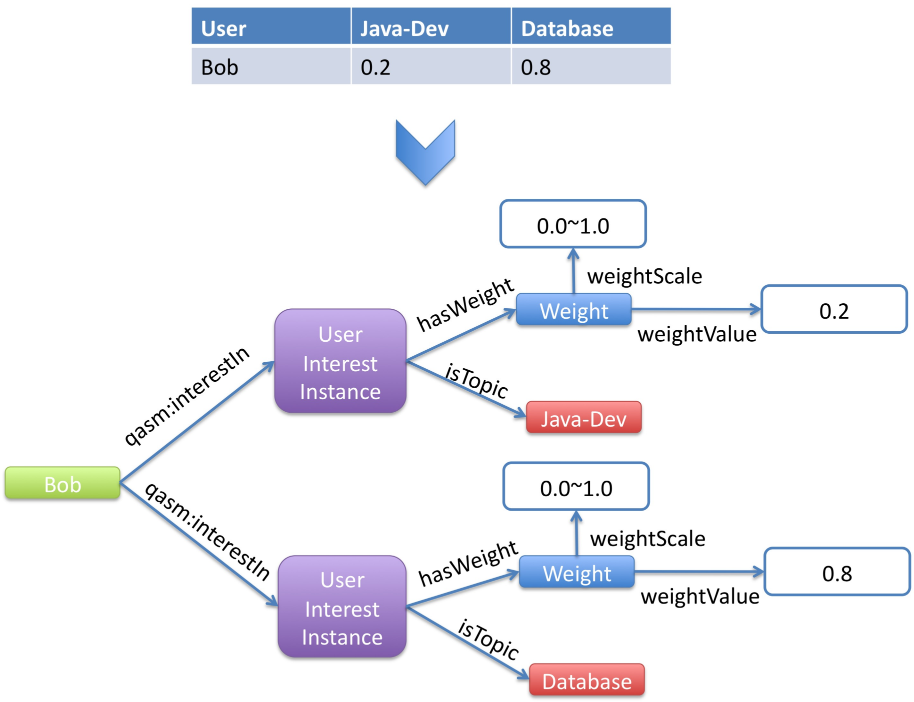


Figure 3.3: Example formalization of a distribution

tion. Thus, we proposed the QASM vocabulary[4]. Figure 3.2 provides an overview of our ontology. It reuses some class and property from SIOC (with soic: prefix), Dublin Core (with dcterms: prefix) and Weighted-Object (with wo: prefix)[5].

Table 3.1 shows its main classes. Table 3.2 shows several properties used in our work.

Since our work mainly generates distributions, we proposed a generic pattern to formalize these distributions. As an example, we show the formalization of a distribution in Figure 3.3.

Here are the main new classes and properties introduced in the QASM vocabulary:

- `qasm:WeightedObject` is used to describe the weight that a specified subject has with regard to a specified object. This class has four subclasses which represent question topics, users' interests, users' expertise and tag topics respectively. In fact, this class is used to model the distributions we extracted from the original data. For

[4]It is available online at `http://ns.inria.fr/qasm/qasm.html`

[5]`http://smiy.sourceforge.net/wo/spec/weightingontology.html` (accessed Aug 2016)

Ontology	Description	Type
sioc:User	active user	explicit
sioc:Question	questions post	explicit
sioc:Answer	answer post	explicit
sioc:Tag	tags used to label questions	explicit
qasm:Word	words used in Q&A content	explicit
dcterms:PeriodOfTime	time interval	explicit
sioc:Topic	bag of words/tags	implicit
qasm:UserInterest	User interest over topic distribution	implicit
qasm:UserExpertise	User expertise over topic distribution	implicit
qasm:TopicTag	Topic over tags distribution	implicit
qasm:TopicWord	Topic over words distribution	implicit
qasm:UserActivity	Topic over users distribution	implicit
qasm:TopicTrend	Topic over time distribution	implicit

Table 3.1: Vocabulary (classes) used in our work

Property	Description	Domain	Range
qasm:interestIn	links a user and a topic he is interested in	sioc:User	qasm:UserInterest
qasm:isTopic	links a UserInterest declaration with a topic	qasm:UserInterest	qasm:Topic
qasm:hasWeight	links a UserInterest declaration with a weight	qasm:UserInterest	wo:Weight
qasm:expertiseIn	links a user with as expertise	sioc:User	qasm:UserExpertise
qasm:isTopic	links a UserExpertise with a topic	qasm:UserExpertise	qasm:Topic
qasm:hasWeight	links a UserExpertise with a weight	qasm:UserExpertise	wo:Weight
qasm:consistOf	links a topic with tags	qasm:Topic	qasm:TopicTag
qasm:isTag	links a TopicTag with a tag	qasm:TopicTag	sioc:Tag
qasm:hasWeight	links a TopicTag with a weight	qasm:TopicTag	wo:Weight
qasm:consistOf	links a topic with a TopicWord declaration	qasm:Topic	sioc:TopicWord
qasm:isWord	links a TopicWord with a word	qasm:TopicWord	sioc:Word
qasm:hasWeight	links a TopicWord with a weight	qasm:TopicTag	wo:Weight
qasm:isPopularAt	links a topic with a trend declaration	qasm:Topic	qasm:TopicYearTrend
qasm:isTimePeroid	links a TopicTrend with a time interval	qasm:TopicTrend	dcterms:PeriodOfTime
qasm:hasWeight	alinks a TopicTrend with a weight	qasm:TopicTrend	wo:Weight
qasm:hasActiveUser	links a topic with an active user activity declaration	qasm:Topic	qasm:UserActivity
qasm:isUser	links a user activity declaration with a user	qasm:UserActivity	sioc:User
qasm:hasWeight	links a user activity declaration with a weight	qasm:UserActivity	wo:Weight

Table 3.2: the Vocabulary (properties) used in our work

example, topic-tag distribution, user-interest distribution.

- `qasm:interestIn` is used to describe the user-interest distribution. This property is different from `foaf:interest` for its range. In FOAF people are interested in documents, while in QASM a user is interested in a topic to a certain degree (a weight). In addition, our model of user interests to is quite similar to the WeightedInterest[6] ontology. The difference is that we mainly focus on formalizing the user-topic interest distribution on Q&A sites. We also formalize expertise, trend, activity distribution on Q&A sites.

- `qasm:expertiseIn` is used to describe the user-expertise distribution. A user has different weights for different topics. The FRAPO ontology[7] has a 'hasExpertise' property to describe a user having an expertise in a specified research area. Our model not only enables to describe a user having expertise on a topic, but also formalizes to what extent a user has expertise.

- `qasm:isPopularAt` is used to describe the topic-time distribution. A topic has different popularity levels at different time interval.

- `qasm:hasActiveUser` is used to describe the topic-user distribution. Different users perform different activities on a topic.

3.4 Formalizing StackOverflow data with the QASM vocabulary

We obtained the data dump of StackOverflow from the website[8]. It includes all user-contributed content on the Stack Exchange network. Each site is formatted as a separate archive consisting of XML files. The data set includes Posts (including all the questions

[6] http://smiy.sourceforge.net/wi/spec/weightedinterests.html (accessed Aug 2016)
[7] http://purl.org/cerif/frapo/hasExpertise (accessed Aug 2016)
[8] https://archive.org/details/stackexchange (accessed Aug 2016)

and answers), Users (including all the user profiles), Votes (including all the vote information for both questions and answers), Comments (including all the comments for both questions and answers) and the schema information (describing the content of each file).

A first step is to map the original dataset to the QASM vocabulary.

The original schema elements and mappings to QASM concepts are listed in Table 3.3.

Original Schema in Data dump	Mapping vocabulary in QASM
Id	sioc:id
PostTypeId(1: Question, 2: Answer)	tsioc:Question, tsioc:Answer
ParentID (only present if PostTypeId is 2)	sioc:reply_of
AcceptedAnswerId (only present if PostTypeId is 1)	qasm:acceptedAnswer
CreationDate	dcterm:created
Score	rev:totalVotes
ViewCount	sioc:num_views
Body	sioc:content
OwnerUserId	sioc:has_owner
LastActivityDate	sioc:last_activity_date
Title	dcterms:title
Tags	sioc:Tag
AnswerCount	sioc:num_replies
CommentCount	qasm:num_comments
FavoriteCount	qasm:num_favorites

Table 3.3: Mapping between original data dump of StackOverflow and QASM vocabulary

Here is a sample of question#9 in Posts.xml file. Each *row* contain a post with all the detailed information about this post. *PostTypeId* is 1 when the post is a question, and is 2 when the post is an answer. *Score* is equal to *UpVote* minus *DownVote*. *Tags* are the keywords which are assigned by users.

```
<?xml version="1.0" encoding="utf-8"?>
<posts>
    <row
        Id="9"
        PostTypeId="1"
        AcceptedAnswerId="1404"
        CreationDate="2008-07-31T23:40:59.743"
        Score="39"
        ViewCount="9011"
        Body="Given a DateTime representing their birthday, how do I calculate someone's age? "
        OwnerUserId="1"
        LastEditorUserId="56555"
        LastEditorDisplayName="Rich B"
        LastEditDate="2009-07-28T20:52:42.660"
        LastActivityDate="2009-07-28T20:52:42.660"
        Title="How do I calculate someone's age in C#?"
        Tags="c#,datetime," AnswerCount="22" CommentCount="0"
        FavoriteCount="21"
    />
</posts>
```

Here is an example of formalized question#9. We list reused schemas in line 3-15. The detailed information about question#9 is described in line 17-34. The mapping between the original post and RDF format are listed in Table 3.3.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rev="http://purl.org/stuff/rev#"
    xmlns:sioc_type="http://rdfs.org/sioc/type#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:dcterm="http://purl.org/dc/terms/"
    xmlns:qasm="http://ns.inria.fr/qasm#"
    xmlns:sioc="http://rdfs.org/sioc/ns#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:vocab="http://localhost:2020/"
    xmlns:dcterm="http://purl.org/dc/terms/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
>
    <rdf:Description rdf:about="post/9">
        <rdf:type rdf:resource="http://rdfs.org/sioc/type#Question"/>
        <rdfs:label>question #9</rdfs:label>
        <sioc:id rdf:datatype="xsd:integer">9 </sioc:id>
        <sioc:has_owner rdf:resource="user/1"/>
        <qasm:acceptedAnswer rdf:resource="post/1404"/>
        <qasm:num_comments rdf:datatype= "xsd:integer">0</qasm:num_comments>
        <sioc:num_views rdf:datatype="xsd:integer">9011</sioc:num_views>
        <dcterm:created rdf:datatype="xsd:dateTime"> 2008-07-31T23:40:59.743 </dcterm:created>
        <dc:title>"How do I calculate someone’s age in C#?"</dc:title>
        <rev:totalVotes rdf:datatype="xsd:integer">39</rev:totalVotes>
        <sioc:last_activity_date rdf:datatype="xsd:dateTime"> 2009-07-28T20:52:42.660 </sioc:last_activity_date>
        <sioc:num_replies rdf:datatype="xsd:integer">22</sioc:num_replies>
        <sioc:content>"Given a DateTime representing their birthday, how do I calculate someone’s age? "</sioc:content>
        <sioc:topic rdf:resource="tag/c#"/>
        <sioc:topic rdf:resource="tag/datetime"/>
        <qasm:num_favorites rdf:datatype="xsd:integer">21</qasm:num_favorites>
    </rdf:Description>
</rdf:RDF>
```

3.5 Modeling the latent knowledge in Q&A sites

Topics, interests, expertises, activities, trends are implicit in the available raw CQA data. We use social media mining techniques to extract this knowledge. In Chapter 5, we propose a Tag Tree Distribution method to efficiently extract topics from tags. In Chapter 7, we jointly model topic, interest, expertise and trend to extract the relations between them, such as user-topic, topic-time, user-expertise, user-interest etc. In Chapter 6 we propose a method using DBpedia to generate labels for the bags of words used to define a topic and therefore to provide a label for the shared interests of a community. In the following we summarize the main distributions that we will use in this thesis and give some examples of the latent knowledge extracted by our models. We also indicate the related vocabulary for each of them.

- **Topic**: A bag of words or tags which are closely related. Words are the content of questions or answers, tags are explicitly attached as such to questions. For example, the topic-tag distribution *Database*:{*mysql*: 0.5, *sql*: 0.3, *query*: 0.2}. expresses that topic *Database* is related to tags *mysql*, *sql*, and *query*. We use `qasm:TopicTag` and `qasm:TopicWord` to formalize this distribution.

- **User Topical Interest**: A user is interested in different topics with different levels. For example, the user-topic distribution *Alice*:{*Database*: 0.8, *Java*: 0.2} expresses that *Alice* prefers to answer questions related to *Database*, but also(to a lesser extent) about *Java*. We use `qasm:UserInterest` to formalize this distribution.

- **User Topical Activity**: Different users are interested in the same topic with different levels. For example, the topic-user distribution *Database*:{*Alice*: 0.8, *Bob*: 0.2} expresses that *Alice* prefers to answer questions related to *Database*, while *Bob* is less willing to contribute answers to it. We use `qasm:UserActivity` to formalize this distribution.

- **Topic Trend**: A topic is popular at different points in time with different levels. For example, the topic-time distribution *Database*:{*May/2013*: 0.2, *June/2013*: 0.3, *July/2013*: 0.5} expresses that the topic *Database* is increasingly popular. We use `qasm:TopicTrend` to formalize this distribution.

- **User Topical Expertise**: A user has expertise in different topics with different levels. For example, the topic-expertise distribution for *Alice ios*:{*High*: 0.2, *Medium*: 0.7, *Low*: 0.1} expresses that *Alice*'s expertise on topic *ios* is probably at a medium level. We use `qasm:UserExpertise` to formalize this distribution.

3.6 Summary: an effective way to manage Q&A sites

We presented QASM, a Q&A system and a vocabulary to combine social media mining and Semantic Web models and technologies to manage Q&A users and content in CQA sites. This chapter provided us with a general framework and vocabulary to capture user-generated content and extracted latent knowledge on Q&A sites. In the next chapters, we will focus on how to efficiently extract this latent knowledge, such as topics and communities. And how to extract more latent information such as topic based temporal dynamics, topic based expertise.

CHAPTER 4

Adapting Latent Dirichlet Allocation to Overlapping Community Detection

Contents

4.1 Introduction to the Latent Dirichlet Allocation Adaptation

In Natural Language Processing (NLP), Latent Dirichlet Allocation (LDA) (Blei 2003) is an increasingly popular document clustering method, a Bayesian network that models how documents in a corpus are topically related. It is used to detect latent topics from documents by constructing a three-layer probabilistic graphical model: document-topic-word. In this three-layer model, documents and words can be observed from a dataset, while topics are a hidden layer which has to be estimated from the observed data. In StackOverflow, a user submits a question, then assigns 1-5 tags to indicate the key topics touched upon this question. Other users who are interested in the question may provide answers to the question or comment on the question or others' answers. Therefore the main structuring graph in StackOverflow is the question-answer graph. As tags attached to a question reflect its scope and domain, users answering a question can be considered as interested in this domain. As a result, a first approach to detect user communities is to consider that a user answering a question acquires the tags attached to this question and that gradually, each user acquires a list of tags associated with frequencies. If we treat the user as a document and tags acquired by the user as words in a document, then community detection can be considered as a clustering problem where users with similar topics of interest are grouped into the same cluster forming a community of interest.

4.1.1 Problem Definition: mining topics and communities

The problem of mining topics and communities in Q&A platforms can be formalized as follows:

Let $U = \{u_1, u_2 ... u_n\}$ be the set of users, $Q = \{q_1, q_2 ... q_m\}$ the set of questions and $T = \{t_1, t_2 ... t_v\}$ the set of tags. We aim at:

1. extracting a topic distribution $Topic = \{topic_1, topic_2 ... topic_k\}$ from T, and for each $topic_k$, defining $topic_k = \{p_{k1}, p_{k2} ... p_{kv}\}$ where p_{ki} denotes the probability of tag t_i to be related to $topic_k$; and then

2. detecting user's interests. For a user $u_i \in U$, we define $I_i = \{I_{i1}, I_{i2} ... I_{ik}\}$ where I_{ik} denotes the probability of u_i to be interested in $topic_k$.

Similarly to (Li 2010b), we applied the classic LDA method to construct a users-topics-tags model to detect latent topics of interest from the tags acquired by users and then cluster users into different topics. The output of the model consists of two probability distributions:

1. a User-Topic distribution to describe to what extent a user is interested in the different topics.

2. a Topic-Tag distribution to describe to what extent a topic is related to the different tags.

The formalization of this model is given by equation 4.1:

$$P(t|u) = P(t|z) * P(z|u) \tag{4.1}$$

where t denotes a tag, z denotes a latent topic, u denotes a user. The probability of a tag for a user is the result of multiplying the probability of this tag for a topic and the probability of this topic for the user.

Probabilistic graphical models (PGM) express the conditional dependence structure between random variables as a graph. The plate notation of the PGM of our model is presented in Figure 4.1. The variables appear as white disks if the variable is observed and blue disks if the variable is hidden (guessed), the blue disks which are written α and β are hyper parameters of the model. The dependencies among the variables are captured by the direction of the edges. The boxes represent replicated variables, which are users, topics (interests) and tags. The Topic box represents different topic-tag distributions for each topic. The User box represents different user-topic distributions for each user. The Tag box represents one topic for each tag for each user.

The parameters of this model are explained in Table 4.1. M and V are given while K, α and β can be chosen. T is observed through the users' tag lists. Other variables are latent variables which have to be estimated.

The intuition behind this model is that users choose their topics and that these chosen topics drive the generation of the tags. The generative process can be summarized as follows:

We use the collapsed Gibbs Sampling method (Griffiths 2004) to sample the hidden variable z, then θ and ϕ can both be estimated. The inference process is as follows. We iteratively sample the topic indicator $z_{m,n}$ for each answer tag $t_{m,n}$ according to equation 4.2:

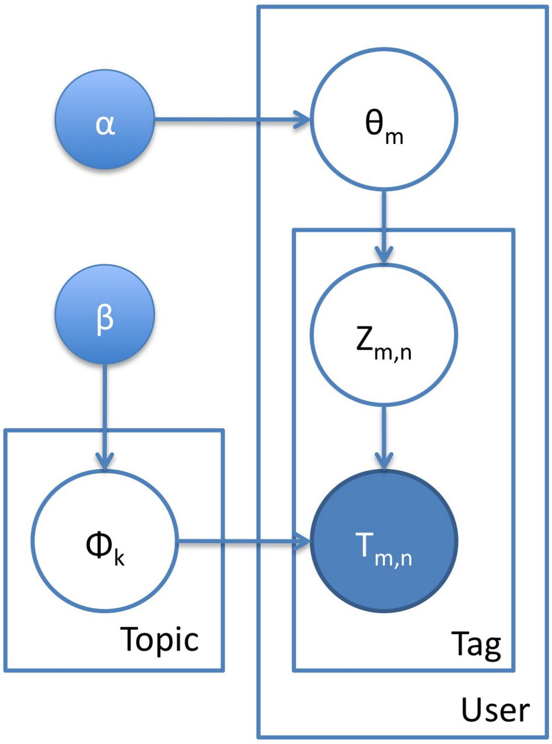


Figure 4.1: User-Topic-Tag (LDA) Model

Process of generating a user tag list
for *topic k* **in** *[1...K]* **do**
  draw topic-tag distribution $\phi(k) \sim \mathrm{Dir}(\beta)$
end for
for *user m* **in** *[1...M]* **do**
  draw a user-topic distribution $\theta(m) \sim \mathrm{Dir}(\alpha)$
end for
for *tag* $T_{m,n}$ **in** $n \in [1...N_m], m \in [1...M]$ ***do***
  draw topic $z_{m,n} \sim \mathrm{Multi}(\theta(m))$
  draw tag $t_{m,n} \sim \mathrm{Multi}(\phi(z_{m,n}))$
end for

Table 4.1: Model parameters

Parameter	Meaning
M	the total number of users
K	the total number of topics
V	the total number of tags
N_m	the total number of tags for user m
α	the parameter of the Dirichlet prior on the per-user topic distributions
β	the parameter of the Dirichlet prior on the per-topic tag distributions
θ_m	the topic distribution for user m
ϕ_k	the tag distribution for topic k
$z_{m,n}$	the topic for the n^{th} tag in m's tag list
$t_{m,n}$	the specified tag at the n^{th} position in m's tag list

$$\begin{aligned} p(z_i = z_{m,n} | u = u_m, t = t_{m,n}, Z, U, T_{\neg i}) \\ \propto \frac{C^{z_{m,n}}_{u_m,\neg i} + \alpha}{\sum_{k=1}^{K} C^{k}_{u_m,\neg i} + K * \alpha} \\ \cdot \frac{C^{t_{m,n}}_{z_{m,n},\neg i} + \beta}{\sum_{t=1}^{V} C^{t}_{z_{m,n},\neg i} + V * \beta} \end{aligned} \tag{4.2}$$

where $\neg i$ enforces that all the counters used are calculated with tag t_i excluded. $C^k_{u,\neg i}$ is the number of tags acquired by user u assigned to topic k, $C^t_{k,\neg i}$ is the number of tags t assigned to topic k.

Then with a Gibbs sampling, we can estimate θ and ϕ by equation 4.3 and 4.4:

$$\theta = \frac{C^k_u + \alpha}{\sum_{k=1}^{K} C^k_u + K * \alpha} \tag{4.3}$$

$$\phi = \frac{C^t_k + \beta}{\sum_{t=1}^{V} C^t_k + V * \beta} \tag{4.4}$$

where C^k_u is the number of tags assigned to topic k of user u , C^t_k is the number of tags t assigned to topic k.

4.2 First experiments: finding topics and communities with adapted LDA

We ran the above described model on a dataset from the popular Q&A site StackOverflow between 2008 and 2009, which is available online[1]. Some basic statistics of the dataset are given in Table 4.2. We see that the total number of users is around 100K and among them, 47K users submitted at least one question, and 54K users answered at least one question. The total number of tags attached to questions is 24K, and 20% of them are used more than 10 times. The frequency of tags follows a power law distribution. The total number of posts is 1.1M; among them there are 242K questions and 870K answers. Each question is attached with between 1 and 5 tags as a tag list. Each user can be represented by her tag lists.

Table 4.2: Basic statistics of the StackOverflow dataset

item	**description**
total number of users	103K (47K questioners, 54K answerers)
total number of tags	24K (20% used more than 10 times)
total number of posts	1.1M (242K questions, 870K answers)

We implemented the LDA algorithm in Python to create a user-topic-tag model as explained above. A first result when running the algorithm is the probability for each tag to belong to each topic. Table 4.3 shows eight examples of the detected topics of interest, each column showing one topic, and the ten rows giving the top 10 tags for each topic, sorted by descending weights. The weight of a tag is the probability that the tag belongs to the topic. This table shows that each topic has a clear and focused interest. For example, topic 1 has c-development related tags, topic 2 has java-development related tags, topic 3 has c#-development related tags, topic 4 has html-development related tags, topic 5 has iphone-development related tags, topic 6 has database related tags, topic 7 has linux-development related tags, topic 8 has non-programming related tags. Moreover, weights reflect the relevance of tags to each topic. For example, topic 5 is concerned with iphone-

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

development and its top 3 tags are 'iphone', 'objective-c' and 'cocoa' which are indeed very relevant.

The second result of the LDA algorithm is the probability for a user to belong to different topics of interest. Table 4.4 shows six randomly chosen users and their top 10 tags. The first row contains user ids, the second row contains their detected topics of interest with their probability. The next ten rows show the top 10 tags for each user. We replaced topic ids with topic names which we have assigned to them according to their associated tags.

4.3 Discussion: limitations and problems

The above experiments verified that, by applying users-topics-tags models on a Q&A website, we are indeed able to detect overlapping communities, and that the detected topics are meaningful and could be used to explain the shared interest of each corresponding community as in our work, we directly use each topic to represent a community of interest.

However, we found that there are three limitations when applying LDA models to our task:

- The first one is a lack of efficiency: the complexity of the probabilistic model was prohibitive. (Wei 2006) shows that the complexity of each iteration of the Gibbs sampling for LDA is linear with the number of topics and the number of documents, which is $O(kn)$, k representing the number of topics, n representing the number of posts. In addition, (Griffiths 2004) proved that LDA model requires a few hundreds-of iterations to obtain a stable topic distribution. Thus, it is necessary to improve the efficiency.

- The second limitation is that the original LDA model does not enable us to extract temporal and expertise information since the observed data in our LDA model are limited to users and tags/words. However, there is actually more information that can be observed in the dataset, such as temporal information and vote information. For expertise modeling, we could not use votes directly because (a) the vote scores

topic 1	topic 2	topic 3	topic 4
c++(0.225)	**java**(0.345)	**c#**(0.225)	**php**(0.117)
c(0.084)	**eclipse**(0.023)	**.net**(0.128)	**javascript**(0.115)
windows(0.020)	**swing**(0.015)	**asp.net**(0.059)	**html**(0.059)
stl(0.014)	best-practices (0.014)	**vb.net**(0.019)	**jquery**(0.056)
algorithm(0.014)	multithreading (0.011)	**linq**(0.018)	**css**(0.042)
c#(0.013)	xml(0.010)	windows-forms (0.016)	mysql(0.029)
win32(0.013)	**spring**(0.010)	**visual-studio** (0.015)	**ajax**(0.021)
linux(0.011)	performance (0.009)	**asp.net-mvc** (0.015)	**web-development** (0.019)
best-practices (0.011)	jsp(0.008)	wpf(0.012)	regex(0.018)
multithreading (0.011)	generics(0.008)	best-practices (0.011)	**asp.net**(0.015)
topic 5	**topic 6**	**topic 7**	**topic 8**
iphone(0.137)	**sql**(0.181)	**python**(0.181)	subjective(0.143)
objective-c (0.123)	**sql-server**(0.150)	**perl**(0.056)	**best-practices** (0.038)
cocoa(0.080)	**database**(0.062)	**regex**(0.031)	**language-agnostic** (0.035)
ms-access(0.062)	delphi(0.042)	**linux**(0.030)	programming (0.028)
cocoa-touch (0.056)	**sql-server-2005** (0.042)	**ruby**(0.027)	**not-programming-related** (0.019)
iphone-sdk (0.041)	**mysql**(0.039)	django(0.023)	**career-development** (0.018)
vba(0.035)	**tsql**(0.037)	ruby-on-rails (0.021)	**learning**(0.017)
excel(0.023)	**oracle**(0.028)	beginner(0.017)	polls(0.017)
vb6(0.022)	**database-design** (0.025)	git(0.013)	programming-languages (0.015)
xslt(0.021)	**stored-procedures** (0.017)	**bash**(0.013)	**design**(0.014)

Table 4.3: Top 10 related tags for 8 detected topics of interest

user_21886	user_14860	user_15401
html-development (0.284) **c-development** (0.275)	**c-development** (0.333) **linux-development** (0.196)	**database-related** (0.383) **non-programming-related** (0.290)
python(93)	c(152)	sql-server(108)
c++(64)	c++(148)	database(64)
javascript(45)	java(89)	sql(63)
html(34)	subjective(89)	subjective(45)
c#(33)	c#(68)	python(43)
css(32)	sql(68)	sql-server-2005(31)
visual-studio(29)	windows(67)	best-practices(27)
windows(27)	linux(54)	.net(25)
c(27)	bash(48)	c++(23)
.net(24)	regex(43)	c#(22)

user_78374	user_53897	user_23743
non-programming-related (0.493) **linux-development** (0.316)	**java-development** (0.835) **non-programming-related** (0.075)	**iphone-development** (0.683) **non-programming-related** (0.155)
subjective(35)	java(366)	objective-c(73)
python(32)	eclipse(24)	cocoa(71)
best-practices(16)	tomcat(20)	iphone(34)
c(13)	subjective(18)	cocoa-touch(21)
programming(13)	performance(18)	mac(19)
c++(10)	best-practices(16)	osx(17)
beginner(8)	j2ee(14)	iphone-sdk(13)
not-programming-related(8)	jar(13)	xcode(10)
language-agnostic(6)	logging(10)	subjective(8)
coding-style(5)	c#(9)	c(8)

Table 4.4: Detected topics of interest for 6 users

are sparse and noncontinuous, and (b) it is not reasonable to say that a vote score of 55 is better than a vote score of 50 if the vote scores are ranging from 0 to 5000. For temporal modeling, similar to (Wang 2006) (Hu 2014), we use time stamps directly. However, it is also important to extract temporal information from different point of view (year, month, day, hour). In addition, contrary to (Blei 2003) who applied the LDA model on long documents such as news articles and assumed that each word has a latent topic, we assume that each answer post has one topic: as with other social media with short contributions, e.g. Twitter, an answer post is normally short, each answer post is therefore suitable to be assigned to one single latent topic, and all the words in that post are considered to be generated by this topic. Some work (Zhao 2011)(Diao 2012) on microblogs also made this assumptions.

Therefore, we aim to extend the original LDA model to extract temporal and expertise information, which will be used to solve the question routing task, etc.

- The third limitation is that the detected probability distributions cannot be compared with each other. Let us explain this in detail. A three-layer LDA model (user-topic-word) generates two kinds of distributions, a user-topic distribution and a topic-word distribution, which describe to what extent a user is interested in different topics and to what extent a keyword or a tag is related to different topics. However, as shown in Figure 4.2, the same user-topic distribution could be generated by different training data (assume that the hidden variable topic is generated by Gibbs sampling (Griffiths 2004)), which means that user-topic distributions are incomparable among users. For the upper distribution of figure 4.2, *Alice* is more active in topic *music*, but for the lower one, *Bob* is more active.

Therefore, in the rest of this thesis we show how we extended our preliminary work in two directions:

1. First, we developed a more simple method to detect topics and overlapping communities to solve the efficiency problem: the TTD method is presented in Chapter5.

	Music	Sport
Alice	0.2	0.8
Bob	0.5	0.5

	Music	Sport
Alice	10	40
Bob	1	1

	Music	Sport
Alice	0.91	0.975
Bob	0.09	0.025

	Music	Sport
Alice	1	5
Bob	10	10

	Music	Sport
Alice	0.09	0.33
Bob	0.91	0.67

Figure 4.2: Different ways to estimate probabilities with topic assignment counts. The upper table: per-user topic distribution; the bottom table: per-topic user distribution

2. Second, we propose a more complex model to extract more information from user generated content to answer the two other limitations: we propose the TTEA method to extract more information and a post-process method to solve the incomparable problem. They are all described in Chapter 7.

CHAPTER 5

Topic Extraction: identifying topics from tags

Contents

5.1 Introduction

In Chapter 3, we mentioned that there are two kinds of information in user-generated content. One kind is latent information such as topics and communities, which do not explicitly exist in the original data set. In this chapter, we aim to extract this latent information from tags on Q&A sites.

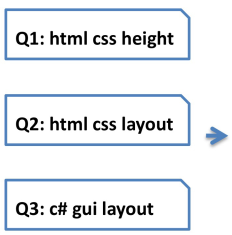
In Chapter 4, we applied the original LDA model and we found that it is complicated and slow to extract this latent information. In this chapter, we aim at proposing a much simpler and more efficient method to extract this information. This is described in section 5.2. Section 5.3 describes our experiments on StackOverflow.

5.2 Topic Trees Distributions (TTD)

5.2.1 First-Tag Enrichment: adding a more general tag when needed

When sorting the tags of a question by their global frequency, we found that normally the first tag of a question is much more general and indicates the domain of the question. For example, a question tagged with {*c#*, *iostream*, *fstream*} is related to *c#*; a question tagged with {*html*, *css*, *height*} is related to *html*. However, there are also some questions which have less tags and, in this case, the tags are less popular, like a question tagged with {*ant*} or a question tagged with {*qt*, *boost*}. For these questions, the main domain is implicit. Our experimental dataset shows that nearly 12% of the questions only have one tag, and nearly 25% of the questions only have two tags.

Therefore, we propose an approach to enrich a question with a first tag when needed. The first step of our approach consists in computing the first-tag distribution.



first->	html	c#
html	2	
css	2	
height	1	
layout	1	1
c#		1
gui		1

first->	html	c#
html	1.0	
css	1.0	
height	1.0	
layout	0.5	0.5
c#		1.0
gui		1.0

Figure 5.1: Example of computing a first-tag distribution

For example, as shown in Figure 5.1, let us consider the three tag lists, {*html*, *css*, *height*}, {*html*, *css*, *layout*}, and {*c#*, *gui*, *layout*}, respectively associated to questions Q1, Q2, Q3 . The first-tag frequency map for *html* is {*html*:2}, the first-tag frequency map for *css* is {*html*:2}, and the first-tag frequency map for *layout* is {*html*:1,*c#*:1}. Given a tag, the probability of its first-tag is computed by equation 5.1, which is the Maximum Likelihood estimation (MLE) of the probability $p(T_f|T_i)$, where $I(T_i)$ denotes the occurrence of tag T_i and $I(T_f, T_i)$ denotes the co-occurrence of first-tag T_f and tag T_i.

$$\begin{aligned} p(T_f|T_i) &= \frac{p(T_f, T_i)}{p(T_i)} \\ &\propto \frac{I(T_f, T_i)}{I(T_i)} \end{aligned} \tag{5.1}$$

We compute the probabilities just by normalizing the first-tag frequency map. In the previous example, the first-tag frequency map for *css* becomes {*html*:1.0} and the first-tag frequency map for *layout* becomes {*html*:0.5, *c#*:0.5}. In order to lower the probabilities of low frequency tags as first-tag, we use the squashing function 5.2:

$$\begin{aligned} p(T_f|T_i) &= \frac{I(T_f, T_i)}{I(T_i)} * \sigma(I(T_f)) \\ &\propto \frac{I(T_f, T_i)}{I(T_i)} * \frac{1}{(1 + e^{-k*I(T_f)})} \end{aligned} \tag{5.2}$$

where, $I(T_f)$ denotes the frequency of *first-tag*. $I(T_f, T_i)$ denotes the co-occurrence of *first-tag* and *tag*, $I(T_i)$ denotes the frequency of *tag* $\sigma(x)$ as a sigmoid function, which is used as a squashing function for numerical stability. The value of the sigmoid function is between 0 and 1, however the shape of this function is largely determined by parameter k. Considering the maximum value of tag frequency (tag c#:$31,801$) in our dataset, we chose k as 0.001 (dotted line), which will lower the probabilities of low frequency tags as first-tag while maintaining the probabilities of high frequency tags as first-tag. Figure 5.2 recalls the shape of the sigmoid function for different values of k.

For example, if the first-tag frequency map for *css* is {*html*:10, *jquery*:2}, then, when

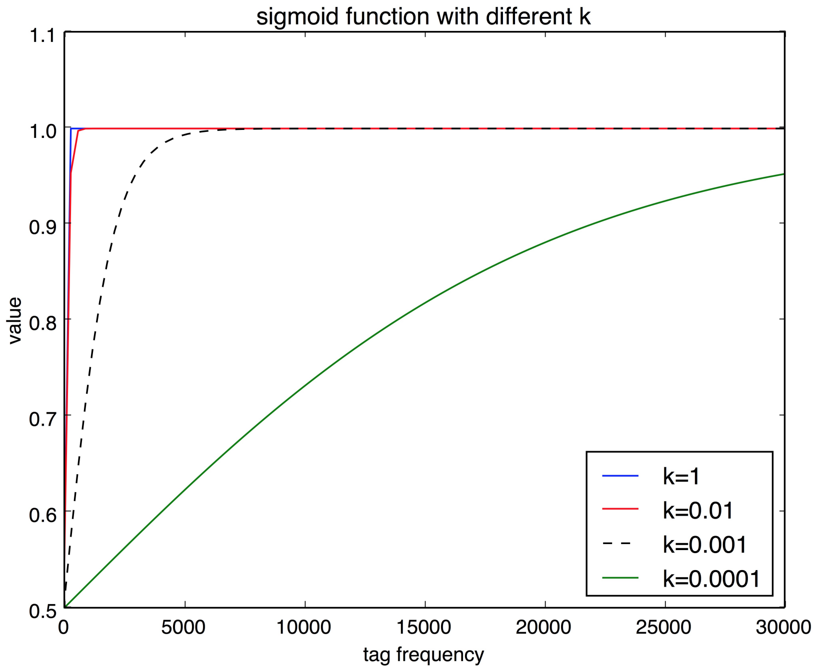


Figure 5.2: Shape of function $\frac{1}{(1+e^{-k*z})}$ for different values of k

normalizing first-tag *html*, $I(T_f, T_i) = 10$, $I(T_i) = 12$, $I(T_f) = 5,552$. As a result, $p(html|css) = 0.8301$. Similarly, for each tag, we provide a list of candidate first-tags with estimated probabilities.

The second step of our approach consists in choosing a first-tag to enrich each question. Given a question's tag list, we fetch the top 5 first-tags (with the highest probabilities). Then we accumulate the corresponding probabilities with a discount function taking into account the position of the tag in the tag list associated to the question, as shown in equation 5.3:

$$p_j = p_{1,j} + p_{2,j} * dis + ... + p_{k,j} * dis^{k-1} \tag{5.3}$$

where p_j denotes the probability of tag j to be the first-tag of a given question, $p_{k,j}$ denotes the probability for tag k to have tag j as its first-tag. The range of j and k are $[1, V]$ and $[1, K]$, where V denotes the number of all the first-tags, K denotes the number of tags in the given question and dis denotes the discount due to the position. There could be two kinds of discount function, linear or non-linear (e.g. exponential) discount. We discuss it in the experiment section.

Then we consider the first-tag with the highest probability as the enriching first-tag. If this first-tag already exists in the original tag list, we simply skip the insertion, or else we insert it at the first position of the question's tag list. We processed $242,552$ tag lists from the StackOverflow Q&A site, and our method enriched $33,622$ of them (13.5%).

Table 5.1 presents the results of the enrichment of 8 tag lists (enriched tags are in bold).

Table 5.1: Original and enriched tag lists

ant	**java**, ant
qt, boost	**c++**, qt, boost
django, hosting	**python**, django, hosting
xslt, dynamic, xsl	**xml**, xslt, dynamic, xsl
sql-server-2005, sorting	**sql**, sql-server-2005, sorting
tomcat, grails, connection	**java**, tomcat, grails, connection
cocoa, osx, mac, plugins	**objective-c**, cocoa, osx, mac, plugins
spring, j2ee, module, count	**java**, spring, j2ee, module, count

5.2.2 Efficient topic extraction from tags

From the observation of our dataset, we confirmed the natural intuition that high frequency tags are more generic and low frequency tags are more specific, and most of the low frequency tags are related to a more generic tag. A similar observation was also found in (Mika 2007). Also, (Yang 2013b) shows that tag frequency in Q&A sites also satisfies a power law distribution (Adamic 2000).

For example, for a question tagged with {*c++, iostream, fstream*} (with tags sorted according to their frequencies), we could find that it was related to *c++* and to the *iostream* topic of *c++*, and more specifically, that it focused on *fstream*. This inspired us to build a tag tree to represent it and compute the probability for a tag to be related to a topic. Figure 5.3 illustrates the process of building a tag tree. Figure 5.4 illustrates an example of *html*'s tree. Our topic extraction method is described in the below algorithm.

In the *build trees* process (lines 3-6), we build a tag tree according to the position of

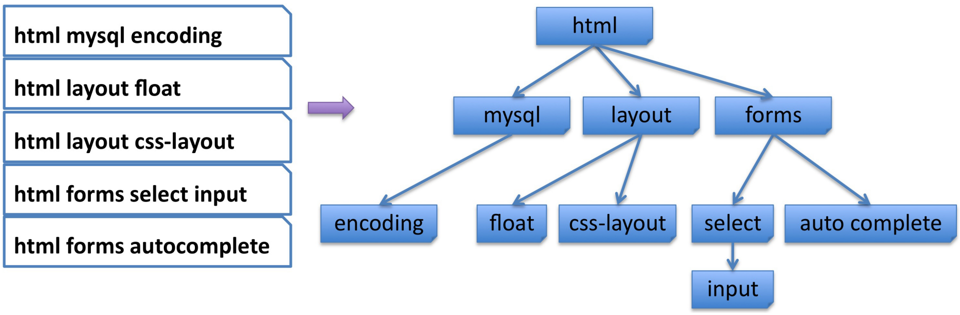


Figure 5.3: Example of a tag tree

Input: *enriched tag list of questions, topic number K*
Output: *topic-tag distribution*
*/*build trees process, shown in Fig 5.3*/*
trees = null */* initialize */*
for *tag* **in** *taglist* **do**
    trees.insert(taglist)
end for
*/*build affinity matrix for root_tags*/*
root_tags = trees.get_root_tags()
affinities_matrix = **build_affinity**(root_tags)
*/*run spectral-clustering on affinity matrix*/*
groups = **spectral**(*affinities_matrix,K*)
*/*combine tree according to groups*/*
new_trees = **combine_tree** (*trees,groups*)
*/*compute topic-tag distribution*/*
topic_distributions = **compute** (*new_trees*)
*** we perform a spectral clustering to divide these root tags into several groups*

tags in a question, and record the occurrence of each node. For example, let us consider again the tag lists of questions Q1, Q2, Q3 in Figure 5.1. Based on them, we construct two trees. The root of the first tree is *html*, the occurrence of this node is *2*, it has only one child *css*, which has *2* occurrences, and this node has two children, *layout* and *height*, and each one occurs *1* time. The root of the second tree is *c#* with *1* occurrence.

By processing all the tag lists, many trees are generated. We then construct an affinity matrix of the root nodes (lines 7-9). Since we applied our first-tag enrichment method, the number of root tags is not very large. The similarity of two root nodes is computed according to equation 5.4:

$$Simi(R_i, R_j) = \frac{I(R_i, R_j)}{(I(R_i) + I(R_j))} \tag{5.4}$$

where $I(R_i, R_j)$ denotes the co-occurrence of root tag R_i and root tag R_j, and $I(R_i)$ and $I(R_j)$ denote the occurrence of root tag R_i and root tag R_j respectively. Then we perform a spectral clustering (Ng 2001) on the affinity matrix to group these root nodes (line 10-11). Each group forms what we will call a topic. As spectral clustering requires that we select the desired number of topics, we choose the same number *30* as (Chang 2013), which has proved to be a reasonable setting for the StackOverflow dataset.

We then combine trees if their root nodes belong to the same topic (lines 12-13). This process leads to a forest where each tree represents a topic. Then, in the *compute topic-tag distribution* process (lines 14-15), for each topic tree, we compute $p(t|k)$, which denotes the probability of tag t belonging to topic k, by using the Maximum Likelihood estimation (MLE), according to equation 5.5:

$$p(t|k) = \frac{p(t,k)}{p(k)} = \frac{I(t) + 1}{\sum I(t) + N} \tag{5.5}$$

where $I(t)$ denotes the number of occurrences of tag t in the topic tree k, and $\sum I(t)$ denotes the total number of occurrences of all tag occurrences in the topic tree.

Compared with the LDA-based model, our model could have a zero-probabilities prob-

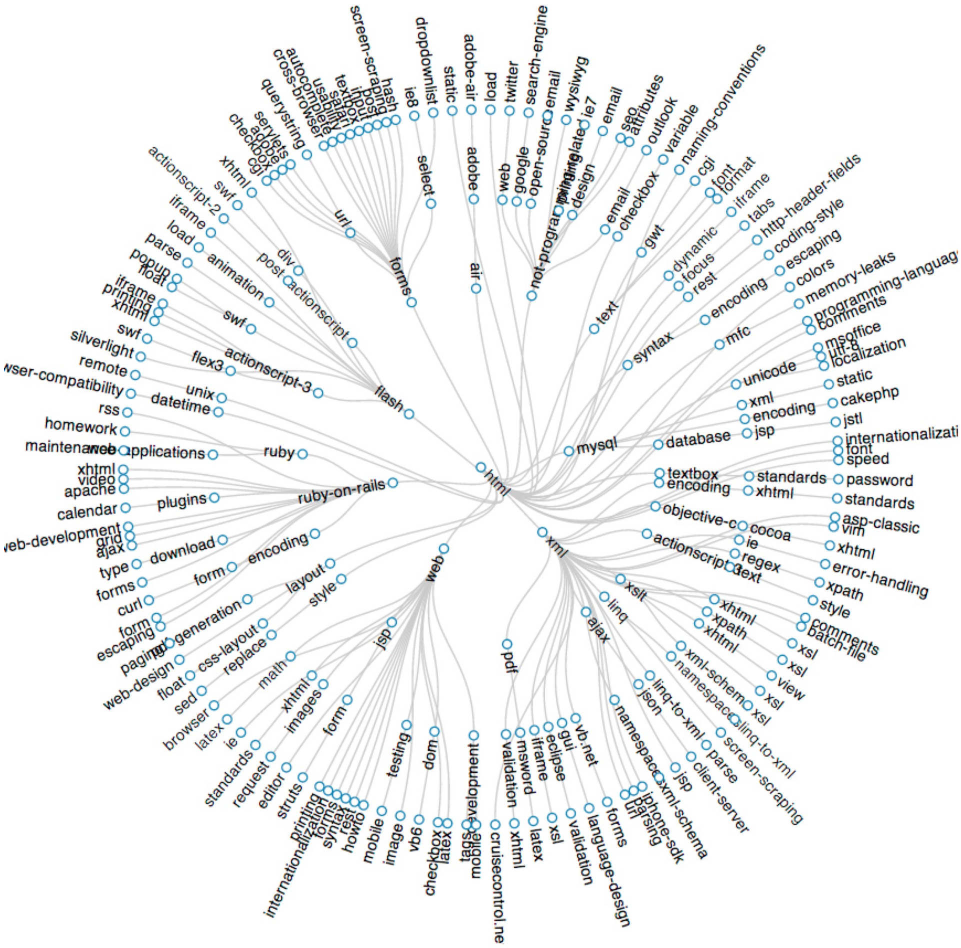


Figure 5.4: *html*'s tag tree

lem, with less popular or new tags related to some topics having a zero probability due to no evidence of co-occurrence. For example, if tag *zombie-process* never occurs in a *html-related* tag tree, then the probability of tag *zombie-process* to be related to *html-related* topics is zero, which could lead to some problems when dealing with young datasets. We avoid it by using the Laplace smoothing method, as shown in equation 5.5. Table 5.3 shows the top tags and their probabilities as detected by our method.

In addition, compared with an LDA-based model, our model is much simpler and faster. The probabilistic graphical model requires hundreds of iterations to get stable results (Griffiths 2004).

We used the spectral clustering implementation of scikit-learn toolkit[1]. We only run it on the set of root nodes, which has quite a small size (around 1175 nodes after the tag enrichment process), which means that we only need to build an affinity matrix on these root nodes and the overall cost therefor remains acceptable.

5.2.3 User Interest Detection: assigning users to topics

In StackOverflow, users answering a question can be considered as interested in the topics denoted by the tags of the question. As a result, a starting point for user interest detection is to model the initial situation as follows: a user answering a question acquires the tags attached to this question and gradually, each user acquires a list of tags.

So we represent a user by a tag list: $U = \{U_i | i = 1, ..., n\}, U_i = \{tag_i | i = m, n, ..., k\}$, and our goal is, for each user U_i, to find $I_i = \{I_{i1}, I_{i2}...I_{ik}\}$ where I_{ik} denotes the probability of user U_i to be related to $topic_k$. As we already have a topic-tag distribution we simply compute the user-topic distribution according to equation 5.6 where $P_{t,k}$ denotes the probability of tag t to be related to topic k. We then normalize the probabilities between 0 and 1 by dividing by the global max value. We use the log function for numerical stability. Here we do not apply normalization at the level of the user, because like

[1]Scikit-learn toolkit:
http://scikit-learn.org/stable/modules/clustering.html#spectral-clustering

(Yang 2013a), we believe that each user could have a high interest in two or more topics simultaneously, while most of the probabilistic graphical models including LDA and PLSA require that the sum of all the probabilities is 1, which means that a user cannot have high probabilities for many topics simultaneously. Our method does not have this limitation.

Then we identify users' communities of interests based on the user-topic distribution: a user having a high probability for a topic should be a member of the community represented by this topic.

$$I_{i,k} = log \left\{ \sum_{t=1}^{v} P_{t,k} + 1 \right\} \tag{5.6}$$

5.3 TTD Experiments and Evaluation on StackOverflow data

We conducted experiments on the dataset of activities in StackOverflow between 2008 and 2009, which is available online[2], to evaluate the performance of our TTD approach compared to three other community detection algorithms. Some basic statistics of the dataset are given in Table 5.2. We see that the total number of users in around 100K and among them, 47K users submitted at least one question, and 54K users answered at least one question. The total number of tags attached to questions is 24K, and 20% of them are used more than 10 times. The frequency of tags follows a power law distribution. The total number of posts is 1.1M; among them there are 242K questions and 870K answers. If two users answer the same question, then the two users are wired by a co-answer link. We filtered the co-answer links with a rule stating that a link is kept if two users answer the same questions more than 10 and 20 times. As a result, we obtained two noise-less datasets.

5.3.1 Performance of Topic Extraction: perplexity metric

We use the Perplexity (Blei 2003) metric to measure the topic extraction performance. It is a common metric in the topic modeling area, measuring how well the words in test

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

Table 5.2: Basic statistics of the stackoverflow dataset

item	description
total users	103K (47K questioner, 54K answerer)
total tags	24K (20% used more than 10 times)
total posts	1.1M (question 242K, answer 870K)
co_answer_10	902 users, 6746 co_answer link
co_answer_20	241 users, 1064 co_answer link
labeled user	902 users, 1-3 labels per user

documents are represented by the word distribution of extracted topics. The intuition is that a better model will tend to assign higher probabilities to the test dataset, corresponding to a lower perplexity value. We split the dataset of question tag lists randomly shuffled into a training set (80%) and a testing set (20%).

We run LDA and our method on the training set to get the topic distribution. Then for a test set of M questions' tag lists (N_d denotes the number of tags in the d^{th} question) the perplexity score is computed as shown in equation 5.7:

$$Perplexity(D_{test}) = exp\left\{-\frac{\sum_{d=1}^{M}\log p(t)}{\sum_{d=1}^{M} N_d}\right\} \tag{5.7}$$

In our model, $p(t)$ is equal to $p(k|q) * p(t|k)$. We compute the topic-question distribution $p(k|q)$ similarly to the user-topic distribution (see Section 5.2.3), by replacing a user's tag lists by a question's tag lists. The only difference is that we normalize the question-topic distribution to make sure that the sum of a question's topic distribution is 1. We show and compare the average perplexity score in Figure 5.5. *TTD* is our method, *TTD_noEnrich* represents our method without first-tag enrichment. We find that TTD can outperform the state-of-the-art LDA method. The reason is that, compared with traditional document topic modeling use cases, question tag lists in Q&A sites are very short, and LDA performs poorly in this situation. Also, our first-tag enrichment method can improve the performance when the number of topics is not very large.

We use different discount functions, which are included in equation 5.3, and compare the perplexity score. We found that the performance when using the discount is better than

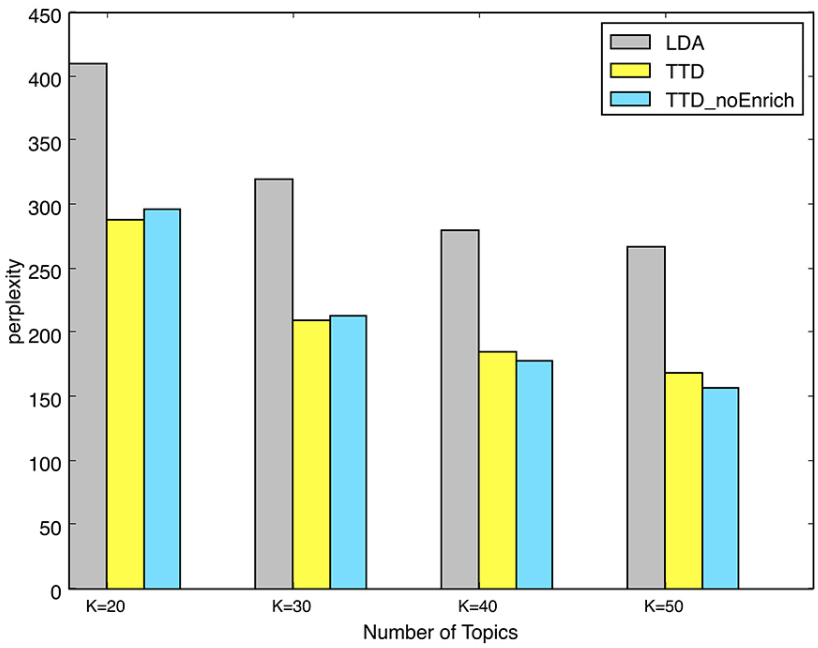


Figure 5.5: Comparison of topic extraction performances

when not using the discount. Also, the linear discount is better than exponential discount.

Another point is that, benefiting from a tree structure for topics, we can easily extract sub-topics from a given topic. In addition, TTD is based on a topic model, so extracting these sub-topics can help us find sub-communities within a detected community. Table 5.4 shows the top tags of *java*'s sub-topic *html* and of topic *html*. We can find that the differences are noticeable for topics: a user who is interested in the topic *html* is not necessarily interested in *java*'s sub-topic *html* and vice versa.

5.3.2 Performance of User Interest Detection: Similarity metrics

Traditional community detection algorithms are based on a network structure. As there is no explicit network in our dataset and in order to compare our work with other approaches on the same dataset, we extracted a network of interactions between users: a co-answer network inspired by the notion of a co-view network introduced in (Gargi 2011). The idea behind it is that if two users answer the same question they share some of their interests.

Table 5.3: Top tags and their probabilities for some topics computed with TTD

topic4		topic5		topic6	
iphone	0.203	git	0.198	sql	0.177
objective-c	0.112	svn	0.096	mysql	0.122
ios	0.109	version-control	0.045	sql-server	0.074
xcode	0.042	github	0.033	database	0.040
cocoa-touch	0.021	tfs	0.033	oracle	0.030
ipad	0.020	maven	0.029	sql-server-2008	0.029
cocoa	0.018	tortoisesvn	0.018	tsql	0.026
uitableview	0.012	msbuild	0.016	query	0.025
ios5	0.010	jenkins	0.015	sql-server-2005	0.019
core-data	0.009	tfs2010	0.014	database-design	0.011
topic12		**topic13**		**topic14**	
html	0.214	javascript	0.264	machine-learning	0.247
css	0.201	jquery	0.114	artificial-intelligence	0.130
xhtml	0.017	html	0.035	neural-network	0.062
web-development	0.016	ajax	0.031	classification	0.046
ie	0.012	css	0.016	data-mining	0.037
css-layout	0.010	firefox	0.013	svm	0.031
div	0.010	dom	0.011	weka	0.025
layout	0.010	php	0.011	libsvm	0.015
firefox	0.009	ie	0.010	nlp	0.024
ie6	0.009	web-development	0.008	bayesian	0.011

Table 5.4: Top tags for *java*'s sub-topic *html* and *mysql*, denoted by java_html, and java_mysql respectively, compared with topics *html* and *mysql*

java_html	jsp swing xml parsing jsf jeditorpane pdf applet dom
html	css xhtml web-development table div ie layout css-layout firefox
java_mysql	jdbc hibernate database tomcat prepared-statement spring connection-pooling connection security
mysql	database query mysql-query ruby-on-rails database-design performance stored-procedures innodb optimization

So, the co-answer network, to some extent, can reflect the common interests between users. We filtered the co-answer links with a rule stating that a link is kept if two users answer the same questions more than 10 times and 20 times, which are the co-answer-10 data set and co-answer-20 data set.

Based on the noise-less dataset obtained, we implemented three well known community detection methods in order to compare our approach with them.

In order to evaluate the results of overlapping community detection, for each user, a method should output 1-3 community labels with corresponding probabilities to indicate to what extent the user is interested in the community. Then we define three levels of interest in a community: *High*, *Medium*, *Low* according to the probabilities. In addition, we empirically set the number of communities to 30 for all the evaluated methods. These are the methods we evaluated:

- SLPA (Xie 2013): An overlapping community detection method inspired by a classical Label Propagation Algorithm (LPA). SLPA algorithm can evaluate to which extent a user belongs to a community by the received propagated label (a 'post-process' in the SLPA algorithm). Therefore, it can output more than one community label according to these frequencies.

- LDA: Similar to (Yang 2013b), we run LDA to build a user-topic-tag model on the given dataset, where users are represented by their tag list. As the output contains a user-topic distribution, we just sort the distribution for each user and choose the top 3 topic labels as community labels together with their probabilities.

- Clustering: We used the implementation of hierarchical clustering from scikit-learn toolkit[3]. As clustering algorithms are hard-partitioned, it can only generate one group label for each user.

- TTD: this is our method. We sort the results of user interest detection (section 5.2.3) and choose the top 3 as community label together with their probabilities.

[3] http://scikit-learn.org/stable/modules/clustering.html

Our aim was to evaluate the similarity between users within a detected community of interest. We mainly used the *jaccard similarity* and *cosine similarity* of two user's tag lists to evaluate the similarity of two user's interests. We used a modified modularity metric to compute the difference between the average similarity between the users within a community (avg_inner) and the average similarity between the users in a community and some user randomly chosen from the whole dataset (avg_rand). This is captured in Equation 5.8, where N represents the number of users in a community C, and $Simi$ denotes the similarity function. $Rand\_U$ represents users that are randomly chosen from the whole data set. A higher value of Avg_inner denotes that users within a community are very similar. A lower value of Avg_rand denotes that users of a community are not very similar to random users. So a higher value of $modularity$ means a larger difference between Avg_inner and Avg_rand, which is considered as a better partition of communities. As the metric has random variables, we run the experiments 10 times and each time we used different random users. In addition, we created a *center* user in each community by averaging all users' tag lists and frequencies, then we computed the average similarity between each user in a community and denote this *center* user as Avg_center. As introduced before, each method gives $1 \sim 3$ community labels for each user to indicate their level of interest. So we evaluated each level of interest respectively.

$$M(C) = \frac{Avg\_inner(\sum_{i=1}^{N}\sum_{j=1}^{N} Simi(U\_i, U\_j))}{Avg\_rand(\sum_{i=1}^{N}\sum_{j=1}^{50} Simi(U\_i, Rand\_U))} \tag{5.8}$$

Experiment results are shown in Table 5.5 and 5.6. We run each method on the co-answer-10 and co-answer-20 dataset 10 times, and listed the average value. We found that our method is better than the three other methods in detecting users' *High* level of interest with both metrics. The reason why our method is not very efficient to detect users' *Low* level of interest is that our method allows users to belong to more than one community with high probabilities, since our method does not have the sum-to-one constrain. For example, a user could be interested in a topic with a probability of 0.7 (High) and interested in several topics with a probability of 0.3 (Low), where the sum of these probabilities is not equal to 1. Then this user will be in many *Low* level of interest communities. This puts in some irrelevant users with *Low* levels of interest which decreases the similarity between community members.

Table 5.7 shows some users and their interests detected with TTD and their top 10 tags. The first row contains user ids, the second row contains their detected communities of interests with their probabilities. The following ten rows show the top 10 tags for each user. We replaced community labels by names assigned according to the tags associated to each topic of interest.

5.3.3 User Study: ranking users' interested topics

In order to evaluate the quality of whether a user is correctly assigned to the right interest group, and to which extent the user belongs to the interest group, we conducted a user study on the dataset by inviting 2 volunteers as annotators. We asked a volunteer to manually label 902 users (in the co_answer_10 dataset) by assigning each user up to 3 labels out of eight group labels, chosen from *c-development* group, *java-development* group, *c#-development* group, *web-development* group, *ios-development* group, *database* group, *linux-development* group and *other-topic* group. For example, if user A sequentially has three group labels, *java-development*,*web-development*,*ios-development*, it means that user A has a big interest in the group *java-development*, a medium interest in the group *web-development*, a lower interest in the group *ios-development*. Since each user has an ordered label list, we have to evaluate both the correctness of detected groups and the correctness of the order. We asked another volunteer (who was not involved in labeling the 902 users) to label the results of the methods with the same 8 labels. As SLPA algorithm can detect overlapping communities, she was asked to assign an interest group name, from the 8 labels, to each community according to users' tag lists in each community, then each user gets at least one interest group name. In addition, the SLPA algorithm can evaluate to which extent a user belongs to a community by the frequency (a 'post-process' in SLPA algorithm). Combined with the interest group name we assigned for each community, the

Table 5.5: Comparison of the performances of the methods of user interest detection on co_answer_10 dataset

Similarity	Jaccard Similarity											
Level	High Interest				Medium Interest				Low Interest			
Metric	avg_inner	avg_rand	*modularity*	avg_center	avg_inner	avg_rand	*modularity*	avg_center	avg_inner	avg_rand	*modularity*	avg_center
TTD	**0.162**	**0.033**	**4.909**	**0.218**	**0.135**	**0.039**	**3.462**	0.171	0.107	0.042	2.548	0.131
LDA	0.147	0.035	4.200	0.178	0.131	0.039	3.359	**0.177**	**0.144**	0.041	**3.512**	**0.193**
SLPA	0.131	0.040	3.275	0.166	0.129	0.040	3.225	0.159	0.121	**0.039**	3.103	0.155
Clustering	0.130	0.041	3.171	0.161	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Similarity	Cosine Similarity											
Level	High Interest				Medium Interest				Low Interest			
Metric	avg_inner	avg_rand	*modularity*	avg_center	avg_inner	avg_rand	*modularity*	avg_center	avg_inner	avg_rand	*modularity*	avg_center
TTD	0.736	**0.574**	**1.282**	0.857	0.573	**0.602**	0.952	0.761	0.475	0.629	0.755	0.695
LDA	**0.836**	0.660	1.267	**0.917**	**0.900**	0.612	**1.471**	**0.948**	**0.757**	**0.600**	**1.262**	**0.865**
SLPA	0.749	0.624	1.200	0.854	0.590	0.621	0.950	0.687	0.702	0.625	1.123	0.844
Clustering	0.763	0.622	1.226	0.875	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 5.6: Comparison of the performances of the methods of user interest detection on co_answer_20 dataset

Similarity	Jaccard Similarity											
Level	High Interest				Medium Interest				Low Interest			
Metric	avg_inner	avg_rand	*modularity*	avg_center	avg_inner	avg_rand	*modularity*	avg_center	avg_inner	avg_rand	*modularity*	avg_center
TTD	**0.175**	0.034	5.121	**0.232**	**0.153**	0.038	4.025	**0.189**	0.115	0.043	2.695	0.132
LDA	0.160	**0.028**	**5.742**	0.198	0.139	0.030	**4.623**	0.188	**0.174**	0.037	**4.697**	**0.224**
SLPA	0.126	0.029	4.290	0.167	0.057	**0.014**	4.104	0.078	0.064	**0.015**	4.394	0.087
Clustering	0.164	0.038	4.303	0.206	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Similarity	Cosine Similarity											
Level	High Interest				Medium Interest				Low Interest			
Metric	avg_inner	avg_rand	*modularity*	avg_center	avg_inner	avg_rand	*modularity*	avg_center	avg_inner	avg_rand	*modularity*	avg_center
TTD	0.676	**0.594**	1.138	0.818	0.548	0.621	0.883	0.745	0.471	0.644	0.730	0.690
LDA	**0.858**	0.627	**1.367**	**0.926**	**0.888**	**0.608**	**1.462**	**0.939**	**0.755**	**0.610**	**1.237**	**0.865**
SLPA	0.729	0.632	1.152	0.845	0.695	0.630	1.104	0.834	0.679	0.632	1.074	0.826
Clustering	0.762	0.627	1.216	0.875	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 5.7: Examples of user interests detected with TTD

user_10224	user_103043	user_113570
database (0.805)c#-dev (0.081)	java-dev (0.664)database (0.105)	c#-dev (0.393)web-dev (0.328)
sql-server (21)	java (135)	c# (107)
sql (21)	swing (28)	jquery (89)
tsql (6)	oracle (27)	javascript (56)
performance (4)	sql (23)	.net (47)
database (4)	subjective (15)	asp.net (27)
stored-procedures (3)	windows (13)	css (23)
sql-server-2005 (3)	eclipse (12)	regex (20)
.net (3)	best-practices (12)	html (20)
mysql (2)	plsql (10)	iphone (12)
sql-server-2000 (2)	regex (10)	string (10)
user_24181	user_34509	user_30461
web-dev (0.743), database (0.072)	c-dev (0.663), linux-dev (0.083)	ios-dev (0.885), linux-dev (0.020)
php (304)	c++ (703)	cocoa (333)
javascript (193)	c (187)	objective-c (184)
mysql (116)	templates (62)	iphone (47)
html (86)	stl (53)	cocoa-touch (39)
css (57)	linux (48)	osx (35)
regex (40)	subjective (45)	mac (34)
jquery (37)	pointers (44)	iphone-sdk (20)
sql (27)	java (42)	xcode (18)
ajax (26)	bash (40)	cocoa-bindings (18)
apache (23)	boost (31)	core-graphics (18)

SLPA algorithm now can output an ordered interest group name list for each user. Clustering algorithms can only generate one cluster id for each user, so she was asked to assign an interest group name, from the 8 labels, for each cluster. The LDA method can give the probability membership to each topic. A high probability indicates that a user is more interested in that group. The volunteer associated the detected 30 topics to the 8 group labels. Then we ordered the interest group name list for each user, sorting them by their probabilities. Our approach is treated just like LDA. Here, she just chose the top 3 group names for each user. The Normalized DCG (NDCG) is introduced to compare different ranking lists. The value of NDCG is between 0.0 and 1.0. In our scenario, a NDCG@p value of 1.0 means detected interests and their order are totally the same as the labeled data untill position p, while a NDCG@p value of 0.0 means that the detected interests are completely different from the labeled data. For values between 0.0 to 1.0, it means that the detected interests are partially correct or ordered incorrectly. Here, we evaluate NDCG@1, NDCG@2, and NDCG@3. The ideal ranking list of each user is the ground-truth and the corresponding score is 10, 8 and 6. Figure 5.6 shows the result of NDCG performance for each method. NDCG@1 reflects the prominent interest detected by each algorithm compared with the ground-truth of a user's prominent interest. We noticed that our empirical method is partially better than LDA, and outperforms SLPA and hierarchical clustering. We also mention that with the dataset becoming less noisy (for people who have prominent and clear-intention interests), all methods' performance increase. The same phenomenon is also observed in NDCG@2,3. As hierarchical clustering algorithms give a hard partition there are no performance comparisons for hierarchical clustering algorithms in NDCG@2,3. Although there is a limitation in the user study because the ground-truth is the human judgement label, which may have some bias. It still worth doing this experiment because that the similarity experiment is focused more on the community, but this user study experiment was focused more on each user.

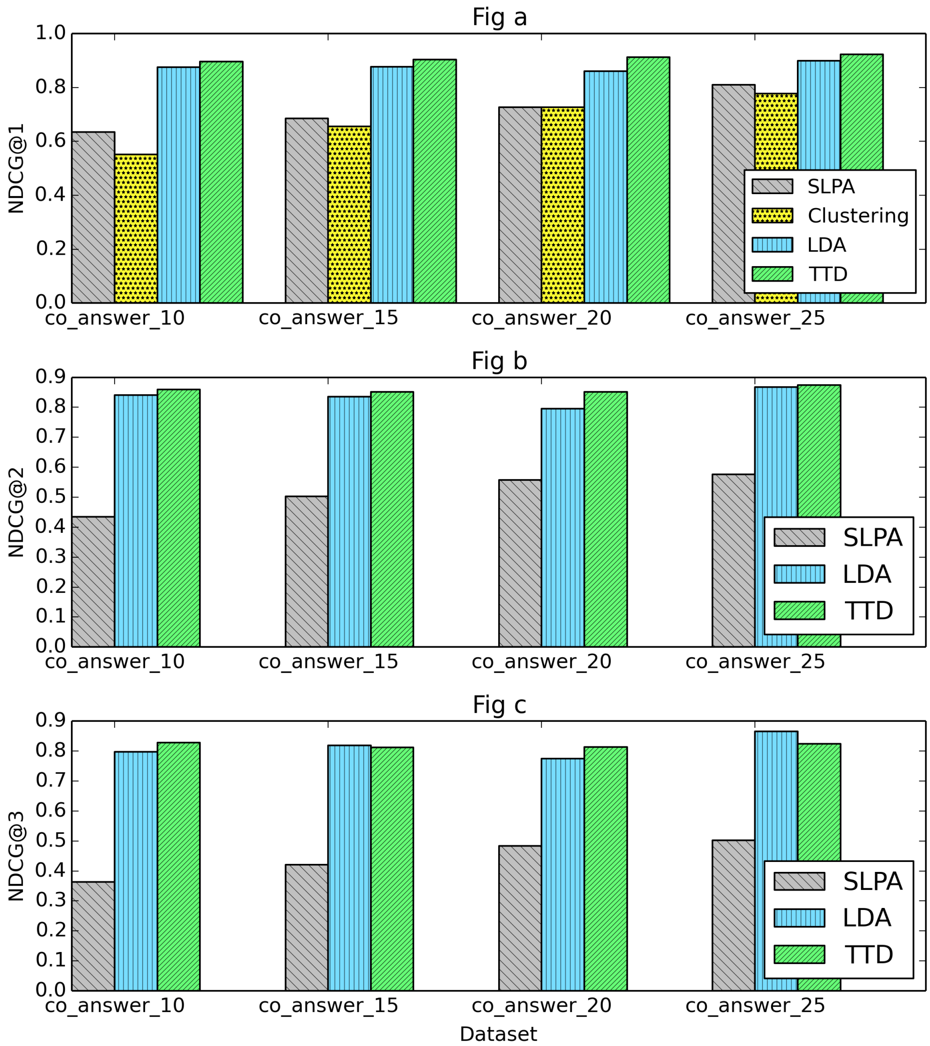


Figure 5.6: NDCG results comparaison

5.3.4 Scalability of topic based user assignment

We also evaluated the scalability of each method. However, as these methods are written in different programming languages, it is not fair to consider this as a precise evaluation; it is just an indication. To increase the stability of the comparison, we run experiments 10 times, and listed the average values. We used a Java implementation of LDA algorithm. All the other methods were implemented in Python. For our method, the time for topic detection was also included. For LDA and SLPA, we set the iteration number to 100. We run the experiments on a computer with a 3GHz Intel i7 CPU and 8GB RAM. From the experiment, we could find that LDA, SLPA and our method are linear in terms of the number of users. Although LDA algorithm is theoretically $O(nm)$ in each iteration, with n representing the number of users, and m representing the number of tags for each user, when we test it on large datasets, it clearly appears that only n actually has an impact; m has a very low impact. Therefor the LDA could be regarded as linear. Also, (Griffiths 2004) proved that the LDA model requires a few hundred iterations to obtain a stable topic distribution. Our model does not have this limitation.

5.3.5 Genericity of the proposed Topic Extraction Method

In order to test whether our proposed topic extraction methods is generic, we collected a dataset from Flickr[4] which contains 1211499 photos attached with tags. For instance, a photo tagged with {*china pinyao*} indicates the location information. A photo tagged with {*night people bar*} describes the time and content information. We run our topic extraction method on this dataset, and we list some results in Table 5.8. We can find that the detected topics are interesting. For example, topic 3 includes photos which contains airplanes, topic 24 includes photos which contains bicycles, and topic 23 includes photos taken in cities of Italy.

[4]Flickr website: https://www.flickr.com/

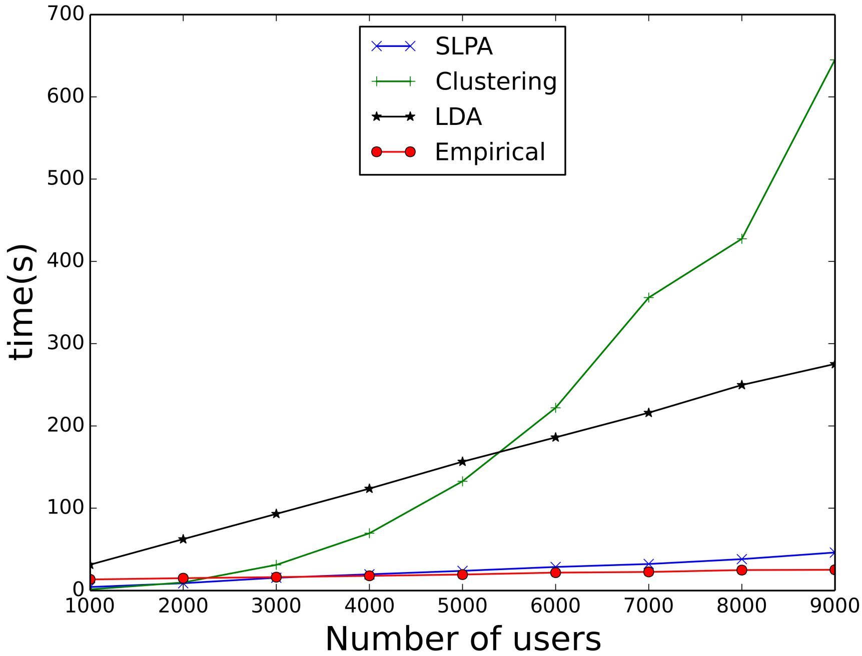


Figure 5.7: Scalability of the compared user interest detection methods

Table 5.8: Top tags and their probabilities on the Flickr dataset

topic3		topic4		topic5	
airplane	0.074	tshirt	0.216	music	0.077
airport	0.053	shirt	0.154	rock	0.040
aircraft	0.029	shirts	0.112	concert	0.036
flying	0.028	threadless	0.109	live	0.025
plane	0.027	tshirts	0.009	band	0.022
aviation	0.022	tee	0.008	singing	0.019
flight	0.014	clothing	0.007	guitar	0.018
aeroplane	0.012	media	0.006	festival	0.017
jet	0.010	models	0.006	show	0.014
boeing	0.009	camiseta	0.004	livemusic	0.010

topic23		topic24		topic25	
italy	0.179	bike	0.114	portrait	0.049
italia	0.053	motorcycle	0.052	girl	0.029
rome	0.028	racing	0.033	woman	0.014
florence	0.021	bicycle	0.028	smile	0.014
venice	0.014	race	0.027	model	0.010
tuscany	0.014	motorbike	0.024	sexy	0.009
roma	0.011	sport	0.019	face	0.008
europe	0.011	speedway	0.011	fun	0.008
firenze	0.010	500cc	0.010	man	0.008
milan	0.007	methanol	0.010	love	0.008

5.3.6 Discussion: community detection in Q&A social network is particular

To sum up, most community detection algorithms work well on real-life social networks which contain many *triangle-shape* structures. The interactions between the users in these networks are mainly based on their relationships. It is also noticeable that the relationships which a user in such a network can maintain are limited and most likely restricted by the location (co-author networks in academia is also exhibit this), so the overall structure of the network is *flatter*, *scattered* and with many *triangle-shape* structures. Comparatively, in Q&A sites, such as StackOverflow, there are no fixed relationships between users. Users interact with each other based on their own interests. Also they are less aware of whom they are interacting with, so they will not maintain explicit relationships. Besides, a user can interact with any other user and mainly interacts with the *"gurus"* (most of questions are answered by a small group of people). So the overall structure of the network is *octopus-shape* (Leskovec 2008) with less *triangle-shape* structures. According to (Park 2013), the average number of *triangle-shape* structures per user in Twitter dataset is around 35714, while in our co-answer-10 dataset, the number of *triangle-shape* structures per user is around 30 which is far less. Therefore, graph-based community detection methods fail in such situation. The result of the SLPA algorithm shows that it outputs one or two giant groups, together with many tiny groups that only contain a small number of users as depicted in Figure 5.8, where each color represents a detected community. We can also see that the network contains less *triangle-shape* structures and a high-density *core*. It also indicates that the network has huge overlaps. However, in the co-answer-25 dataset, the graph structure is more *flatter* and contains many *triangle-shapes*. Therefore, as shown in Figure 5.9, the result of the SLPA algorithm outputs several medium sized groups.

Since clustering methods normally generate hard-partition communities, they cannot detect the overlapping communities which are typical in our case. Concerning the LDA-based methods, on one hand, in our dataset, question tag lists are quite short, and the experiment shows that our topic extraction method gives better results in this situation. On the other hand, the probabilistic graphical model requires hundreds of iterations to get

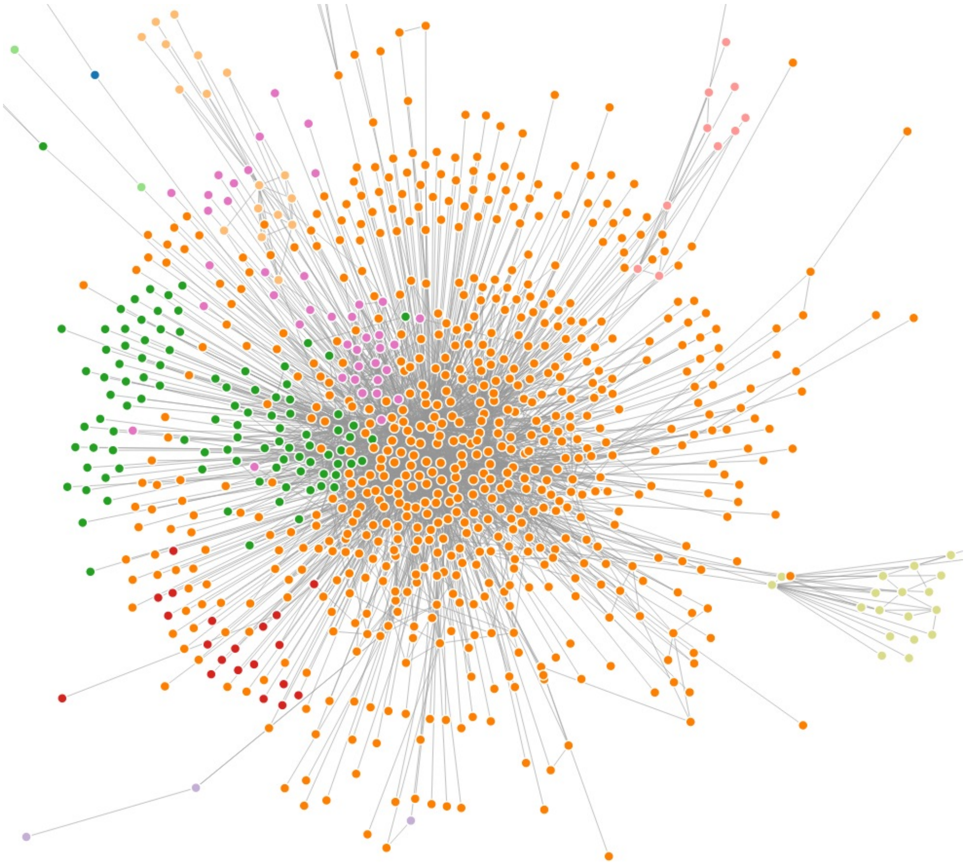


Figure 5.8: Illustration of co-answer-network-10, different colours indicate detected communities

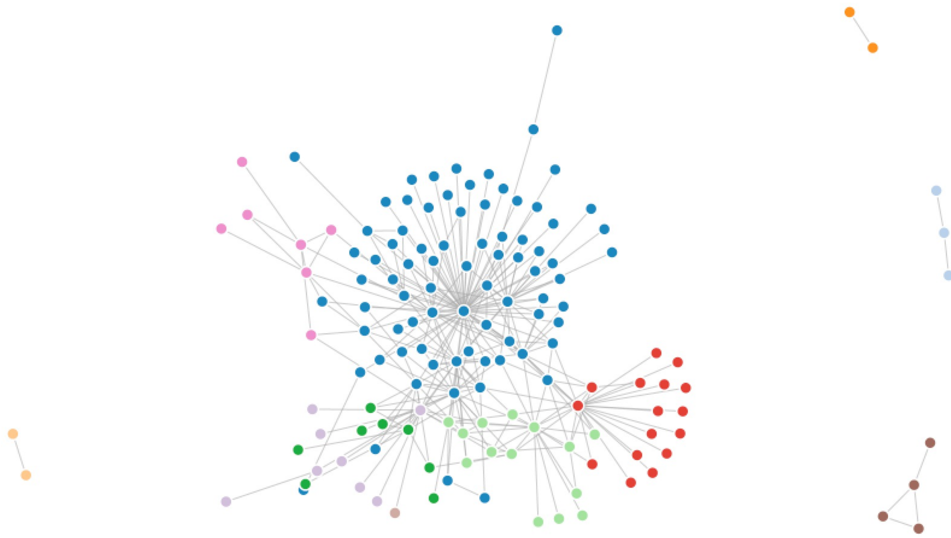


Figure 5.9: Illustration of co-answer-network-25, different colours indicate detected communities

stable results (Griffiths 2004) which is more complicated and slower than our method.

5.4 Summary: an efficient user topic extraction method

Recalling our research questions (How can we detect communities of interests in Q&A sites? How can we also identify the topics that attract them?) we believe that we have proposed a topic detection method which is very suitable for Q&A datasets and an efficient user interest detection method to discover topic based overlapping communities of interests. As we found in the topic extraction result, the output is just bags of words with labels such as "topic 15", "topic 30". Since it is not easy to understand the meaning of the topic using these labels, we try to tackle this problem in the next chapter. The goal will be to automaticaly generate a label for a bag of words.

CHAPTER 6

Automatic generation of labels for topics' bags of words

Contents

6.1 Introduction: finding labels to represent a topic

In natural language processing and information retrieval, topic modeling classically uses bags of words to represent the meaning of a text. However, this is not sufficient to support user interactions as bags of words require an effort from the user to go through the lists of the most important words in order to get an idea of the topic these words represent when considered together. In Chapter 5 we discussed a method that extracts topics from tags and the outputs of topic model are indeed bags of words, each of them representing a detected topic of interest. At this stage we could only attach meaningless labels for each topic, such as *topic 3*, *topic 5*.

Let us now consider examples of such topics:

- italy, florence, venice, tuscany -> *Italy*
- git, svn, tfs, maven -> *version-control*
- machine-learning, artificial-intelligence, neural-network, classification -> *artificial-intelligence*

The labels, (e.g. *Italy*, *version-control*, *artificial-intelligence*), on the right hand side are good candidates to summarize the overall topics captured by the bags of words on the left hand side. Those labels are at least more informative than labels such as *topic 3* and *topic 5* and can be used in interfaces and graphical representations of the results of the detection of communities of interest.

So an interesting task we consider in this chapter is how to automatically generate a general label for the bags of words representing a topic, which can best represent the meaning of that topic. (Sun 2015) introduces the task of conceptual labelling (CL), which aims at generating a minimum set of conceptual labels that best summarize a bag of words. Our work is similar to this one, but the main difference is that we use DBpedia as our external knowledge and we use graph centrality based algorithms to help generate labels to represent a bag of words. (Hulpus 2013) also proposes using DBpedia and graph centrality based algorithms to choose labels. Our approach differs from it in that rather than using existing graph centrality based algorithms to generate labels, it is a hybrid method.

6.1.1 Problem definition: words, topics and labels

Our previous work on topic modeling can generate topics from words or tags. Each topic consists of several tags or words. Table 6.1 and Table 6.2 list some detected topics from a Flickr dataset and StackOverflow dataset. The topic extraction algorithms are able to put closely related words or tags into the same topic, however, they can only use meaningless IDs (e.g. topic 3) to represent a topic. Our goal in this chapter is to find a label (e.g. aviation) to replace the original label (e.g. topic 3).

Table 6.1: Top tags and their probabilities in the Flickr dataset

topic3		topic4		topic5	
airplane	0.074	tshirt	0.216	music	0.077
airport	0.053	shirt	0.154	rock	0.040
aircraft	0.029	shirts	0.112	concert	0.036
flying	0.028	threadless	0.109	live	0.025
plane	0.027	tshirts	0.009	band	0.022
aviation	0.022	tee	0.008	singing	0.019
flight	0.014	clothing	0.007	guitar	0.018
aeroplane	0.012	media	0.006	festival	0.017
jet	0.010	models	0.006	show	0.014
boeing	0.009	camiseta	0.004	livemusic	0.010
topic23		**topic24**		**topic25**	
italy	0.179	bike	0.114	portrait	0.049
italia	0.053	motorcycle	0.052	girl	0.029
rome	0.028	racing	0.033	woman	0.014
florence	0.021	bicycle	0.028	smile	0.014
venice	0.014	race	0.027	model	0.010
tuscany	0.014	motorbike	0.024	sexy	0.009
roma	0.011	sport	0.019	face	0.008
europe	0.011	speedway	0.011	fun	0.008
firenze	0.010	500cc	0.010	man	0.008
milan	0.007	methanol	0.010	love	0.008

6.2 Proposed approach: using DBpedia information

6.2.1 Linking to DBpedia

DBpedia[1] is a crowd-sourced community effort to extract structured information from Wikipedia[2] and make this information available on the Web. It allows users to link their own dataset to Wikipedia data and to augment it with this huge amount of additional data, documents and links. The DBpedia knowledge base now plays an important role in enhanc-

[1] `http://dbpedia.org/about` (accessed Feb 2016)
[2] `https://www.wikipedia.org/`(accessed Feb 2016)

Table 6.2: Top tags and their probabilities on stackoverflow dataset

topic4		topic5		topic6	
iphone	0.203	git	0.198	sql	0.177
objective-c	0.112	svn	0.096	mysql	0.122
ios	0.109	version-control	0.045	sql-server	0.074
xcode	0.042	github	0.033	database	0.040
cocoa-touch	0.021	tfs	0.033	oracle	0.030
ipad	0.020	maven	0.029	sql-server-2008	0.029
cocoa	0.018	tortoisesvn	0.018	tsql	0.026
uitableview	0.012	msbuild	0.016	query	0.025
ios5	0.010	jenkins	0.015	sql-server-2005	0.019
core-data	0.009	tfs2010	0.014	database-design	0.011
topic12		topic13		topic14	
html	0.214	javascript	0.264	machine-learning	0.247
css	0.201	jquery	0.114	artificial-intelligence	0.130
xhtml	0.017	html	0.035	neural-network	0.062
web-development	0.016	ajax	0.031	classification	0.046
ie	0.012	css	0.016	data-mining	0.037
css-layout	0.010	firefox	0.013	svm	0.031
div	0.010	dom	0.011	weka	0.025
layout	0.010	php	0.011	libsvm	0.015
firefox	0.009	ie	0.010	nlp	0.024
ie6	0.009	web-development	0.008	bayesian	0.011

ing the intelligence of Web applications and in supporting information integration. Among the advantages of the DBpedia knowledge base are the fact that it covers many domains and that it automatically evolves with Wikipedia changes. It currently describes 38.3 million things in total and contains 3 billion RDF triples (2014 version).

In order to use the DBpedia knowledge base, a basic step is to link the bag of words to DBpedia. For example, the word *javascript* could be linked to DBpedia resource *http://dbpedia.org/resource/JavaScript*, the word *beer* could be linked to DBpedia resource *http://dbpedia.org/resource/Beer*. However, in some cases, several resources or entities may correspond to the same word (homonymy). For instance, *java* could be linked to the *Java* island but it could also be linked to the *Java* programming language. Therefore, we have to deal with a disambiguation problem when linking words to DBpedia resources. This is a well-known problem now and extensively studied by researchers working on entity recognition, named entity detection and entity linking. Babelfy (Moro 2014) is a unified graph-based approach to solve Entity Linking (EL) and Word Sense Disambiguation (WSD) problems. Their experiments show the state-of-the-art performances on both tasks on 6 different datasets. Moreover they provide an online webservice[3]. So we directly used their web API to retrieval DBpedia links for the words in our dataset. In addition, we used classical similarity metrics to solve the disambiguation problem, as detailed in the next subsection.

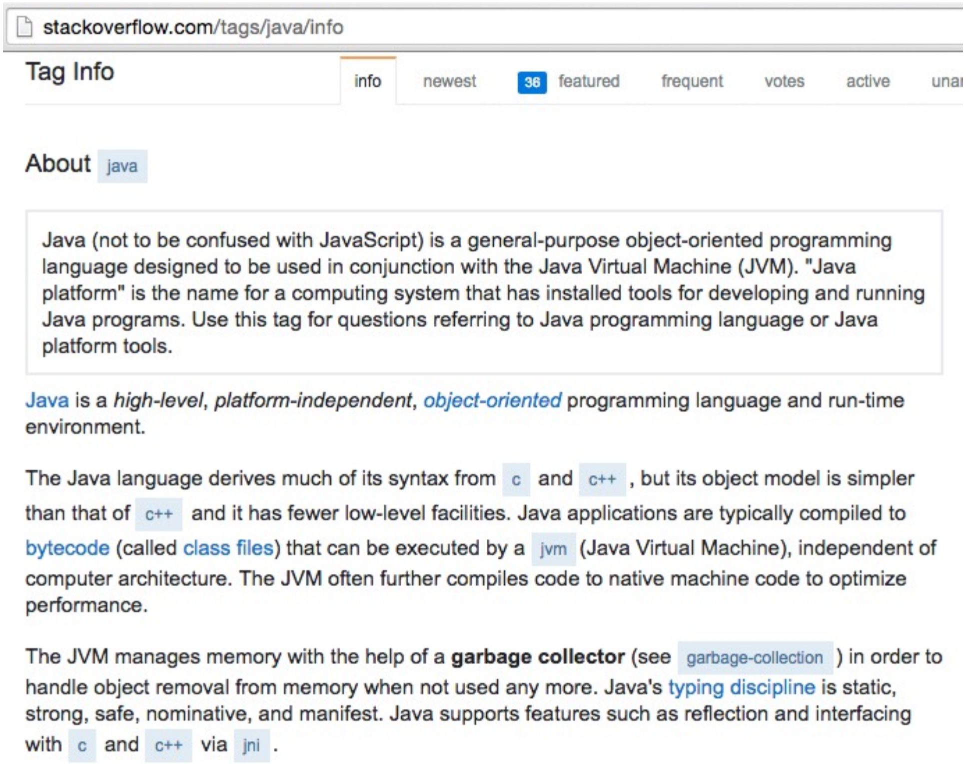
6.2.2 Using descriptions' cosine similarity for disambiguation

Our main dataset is from the StackOverflow website and we found that there are detailed descriptions for each tag on the website, as shown in Figure 6.1.

Also, each resource in DBpedia has a description. We used the DBpedia keyword lookup service [5] to retrieve related resources for each tag. As shown in Figure 6.2, the result of a call to the lookup service is a list of resources related to the given keyword.

In order to link *java* to the correct DBpedia resource, we compute the cosine similarity

[3] http://babelfy.org (accessed Feb 2016)
[5] http://dbpedia.org/projects/dbpedia-lookup(accessed Feb 2016)



stackoverflow.com/tags/java/info

Tag Info

info newest 38 featured frequent votes active unai

About java

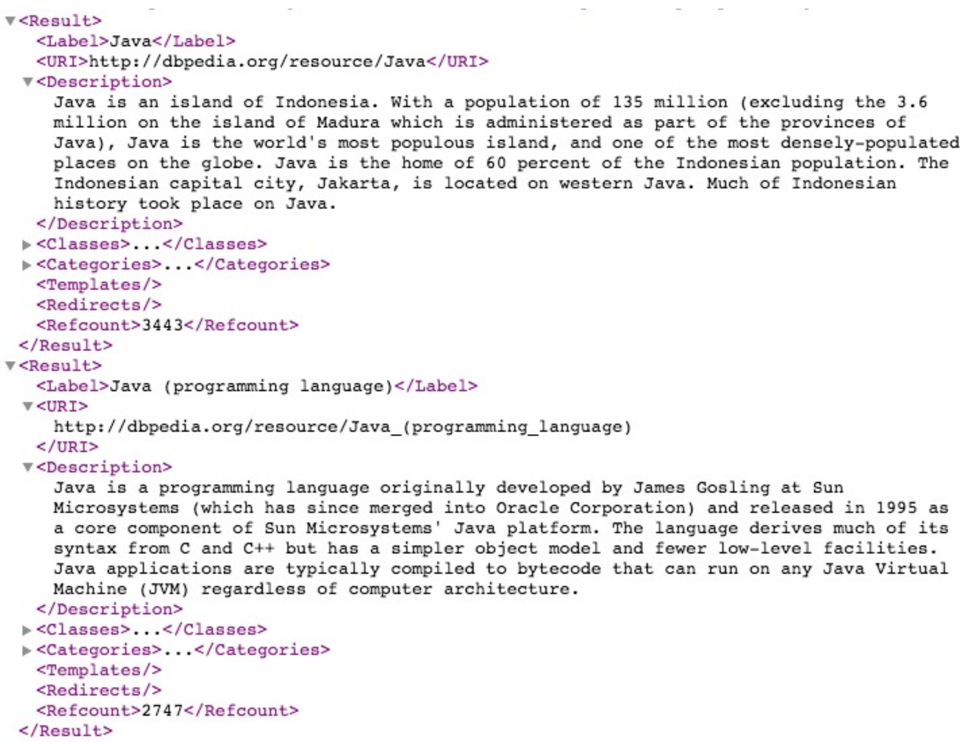
Java (not to be confused with JavaScript) is a general-purpose object-oriented programming language designed to be used in conjunction with the Java Virtual Machine (JVM). "Java platform" is the name for a computing system that has installed tools for developing and running Java programs. Use this tag for questions referring to Java programming language or Java platform tools.

Java is a *high-level*, *platform-independent*, *object-oriented* programming language and run-time environment.

The Java language derives much of its syntax from c and c++ , but its object model is simpler than that of c++ and it has fewer low-level facilities. Java applications are typically compiled to bytecode (called class files) that can be executed by a jvm (Java Virtual Machine), independent of computer architecture. The JVM often further compiles code to native machine code to optimize performance.

The JVM manages memory with the help of a **garbage collector** (see garbage-collection) in order to handle object removal from memory when not used any more. Java's typing discipline is static, strong, safe, nominative, and manifest. Java supports features such as reflection and interfacing with c and c++ via jni .

Figure 6.1: Description of *java* on StackOverflow dataset [4]




```
▼<Result>
   <Label>Java</Label>
   <URI>http://dbpedia.org/resource/Java</URI>
  ▼<Description>
     Java is an island of Indonesia. With a population of 135 million (excluding the 3.6
     million on the island of Madura which is administered as part of the provinces of
     Java), Java is the world's most populous island, and one of the most densely-populated
     places on the globe. Java is the home of 60 percent of the Indonesian population. The
     Indonesian capital city, Jakarta, is located on western Java. Much of Indonesian
     history took place on Java.
   </Description>
  ▶<Classes>...</Classes>
  ▶<Categories>...</Categories>
   <Templates/>
   <Redirects/>
   <Refcount>3443</Refcount>
 </Result>
▼<Result>
   <Label>Java (programming language)</Label>
  ▼<URI>
     http://dbpedia.org/resource/Java_(programming_language)
   </URI>
  ▼<Description>
     Java is a programming language originally developed by James Gosling at Sun
     Microsystems (which has since merged into Oracle Corporation) and released in 1995 as
     a core component of Sun Microsystems' Java platform. The language derives much of its
     syntax from C and C++ but has a simpler object model and fewer low-level facilities.
     Java applications are typically compiled to bytecode that can run on any Java Virtual
     Machine (JVM) regardless of computer architecture.
   </Description>
  ▶<Classes>...</Classes>
  ▶<Categories>...</Categories>
   <Templates/>
   <Redirects/>
   <Refcount>2747</Refcount>
 </Result>
```


Figure 6.2: Result of the DBpedia lookup service for keyword *java*

between the two descriptions from the two websites (StackOverflow and DBpedia) to solve the disambiguation problem. We show an example in Figure 6.3. The entire procedure is described as follows.

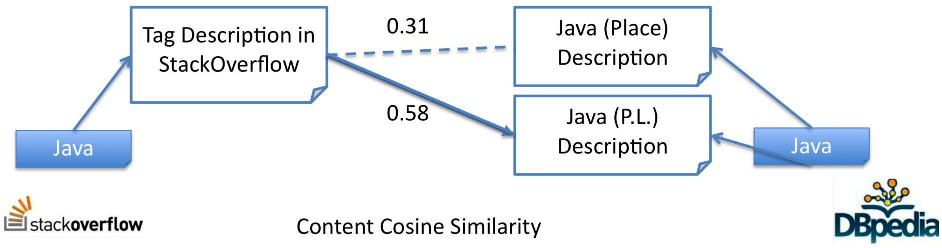


Figure 6.3: The example of disambiguation by computing the cosine distance of the descriptions

```
Input: tag
Output: tag-DB link
//crawl the tag description from StackOverflow
tagSO=getTagDescriptionSO( tag )
//retrieve DBpedia resources by the lookup service
tagResouces=getResourecesDB( tag )
//compute the cosine distances between the description from StackOveflow and the
description of each retrieved resource
DBLink=NULL
maxDis=-1.0;
for tagResource in tagResources do
    dis=consieDistance( tagSO, tagResource.Descrition )
    if ( dis > maxDis ) then
        //link the tag to the resource with the highest similarity
        DBlink=tagResource.Link
        maxDis=dis
    end if
end for
return tag,DBLink
```

6.2.3 Creating graphs: retrieving potential links between resources

After linking tags to their corresponding DBpedia resources, we then perform several SPARQL queries to retrieve the potential relations among the resources found for each topic.

```
/***DBpedia graph extaction queries***/
procedure EXTRACTGRAPH(ra,rb)
    /***Depth=1:***/
    select ?relation
    where{
        ra ?relation rb.
    }
    /***Depth=2:***/
    select ?r1, ?relation1, ?relation2
    where{
        ra ?relation1 ?r1.
        ?r1 ?relation2 rb.
    }
    /***Depth=3:***/
    select ?r1, ?r2, ?relation1, ?relation2, ?relation3
    where{
        ra ?relation1 ?r1.
        ?r1 ?relation2 ?r2.
        ?r2 ?relation3 rb.
    }
```

where, *ra*, *rb* are the resources for which we want to retrieve the potential relations and $?r1$, $?r2$, $?relation1$, $?relation2$, $?relation3$ are the potential relations and the intermediate resources. $Depth$ denotes the hops between the resources *ra* and *rb*. We vary this parameter by $1, 2, 3$. Figure 6.4 shows the retrieved graph for the linux related topic.

To remain compatible with SPARQL 1.0 we did not use the path operator that would support a much more synthetic way of writing these queries. The general idea behind these queries is to reconstruct a small connected graph around the detected resources for a topic in order to obtain a space where we can analyze their relations.

Once we have these relation graphs, we perform several graph algorithms to choose one or several resources as candidates to label the bag of words of the topics. We mainly used the following algorithms/metrics:

- InDegree (ID)

  In a directed graph, for a node, the number of head ends adjacent to a node is called the indegree of the node.

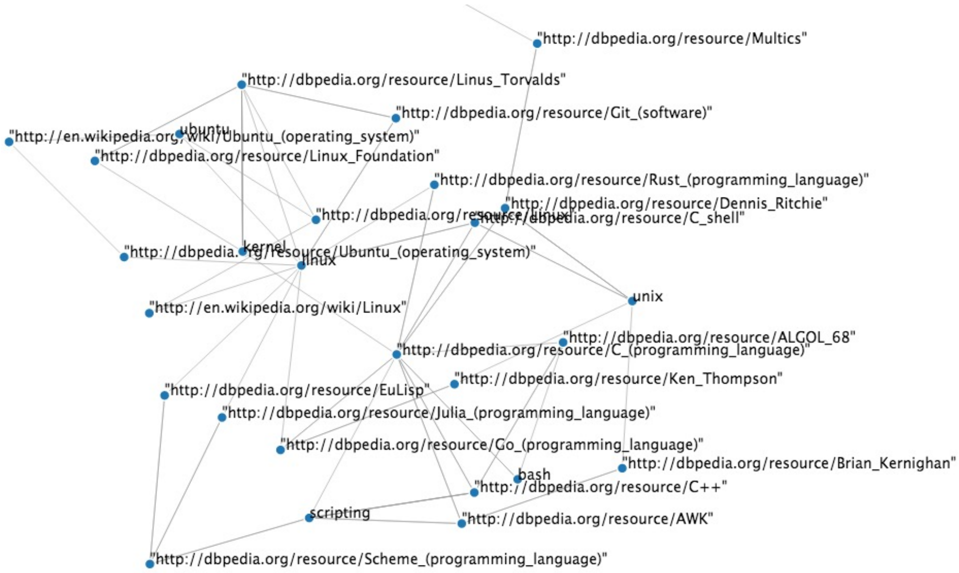


Figure 6.4: The example graph structure for the linux related topic

- Betweenness Centrality (BC)

  It is an indicator of a node's centrality in a network. It is equal to the number of shortest paths from all vertices to all others that pass through that node.

- Degree Centrality (DC)

  It is defined as the number of links incident upon a node, which is equal to indegree plus outdegree for a directed graph.

- PageRank (Page 1999) (PR)

  PageRank is an algorithm used by Google Search to rank websites in their search engine results. It can be applied on other kinds of graphs to estimate the importance of the nodes.

- Random

  We just randomly choose one node from the graph.

- Top tags (Top)

  The topic modeling algorithm generates a topic-word distribution to indicate to what extent a word is related to a topic. By sorting words' corresponding probabilities, we can obtain a ranked word list for each topic, which are the top related words in each topic. A naive approach can use the first tag or first two tags to label each topic.

We proposed a method called "Most+Top" which consists of creating a list of the most recommended labels by all of the above algorithms and then getting a label from this list by using the above "Top tags" algorithm.

6.3 Experiments: A survey study

In order to evaluate the performances of the different ways to generate labels we conducted user studies on the results. We designed two surveys for the user study. Table 6.3 shows the structure of the survey we used. For each survey, we listed 30 topics, half of them are from the StackOverflow dataset, half of them are from the Flickr dataset. The only difference for survey A and B is the linking (disambiguation) method for the StackOverflow dataset. As mentioned in section 6.2, we use both cosine similarity and Babelfy to link tags with DBpedia resources.

Table 6.3: Survey description and corresponding linking method

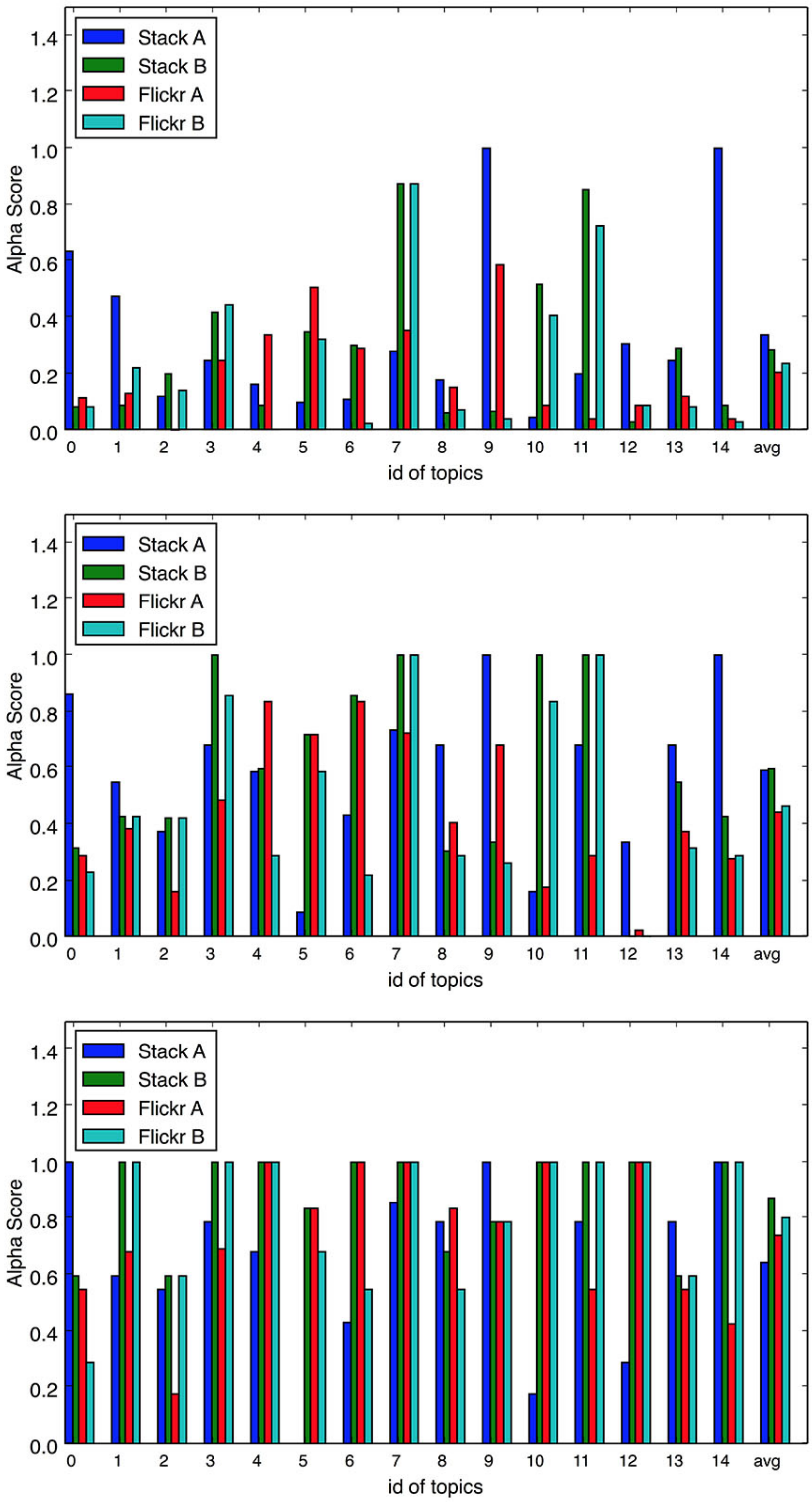
	15 StackOverflow Topics	15 Flickr Topics
Survey A	Cosine Similarity	Babelfy
Survey B	Babelfy	Babelfy

6.3.1 Users' agreement

We use Krippendorff's Alpha[6] score to evaluate the degree of agreement among users. The score indicates the homogeneity, or consensus, in the ratings given by users. The score is always smaller than 1, $\alpha = 1$ indicates the judges reach a perfect agreement and $\alpha = 0$ indicates the judges do not agree at all. When $\alpha < 0$ this means that judges reached a disagreement exceeding what can be expected by chance. Figure 6.5 illustrates the alpha score for 15 topics in each dataset and the average alpha score. We evaluate this score in three levels which correspond to the three sub figures. If we consider the top voted label as

[6] https://en.wikipedia.org/wiki/Krippendorffś_alpha

Figure 6.5: Agreement Alpha Score on the top X labels



the best label, the first figure shows the agreement score among users. When we lower this limitation and consider the set of the top two voted labels as the best label, we can find that for most of the topics users could reach a good agreement. When we keep lowering this limitation and consider the top three voted labels as the best label, we find that they reach an excellent agreement.

In addition, we calculate the proportion of top voted labels. Figure 6.6 shows the number of topics for which top voted labels take a certain proportion. The proportion of top voted labels are plotted on the X-axis, and numbers of topics are plotted on the Y-axis. For instance, a data point (50%,6) in the first sub figure means that there are 6 topics for which first voted labels corresponds to 50% percent of all voted labels. A data point (50%,6) in the second sub figure means that there are 6 topics which top two voted labels take 50% percent of all voted labels. Similarly, a data point (50%,6) in the third sub figure means that there are 6 topics for which top three voted labels corresponds to 50% percent of all voted labels. These three figures show that most of the labels chosen by judges are actually highly voted labels, which means all judges tend to agree on the top two or three labels.

6.3.2 Quality evaluation: NDCG measurement

We use the NDCG metric to evaluate all the algorithms listed in Section 6.2. The Normalized DCG (NDCG) is introduced to compare different ranking lists. The value of NDCG is between 0.0 and 1.0. As before, a NDCG@p value of 1.0 means detected interests and their order are totally the same as the labeled data until position p, while a NDCG@p value of 0.0 means that the detected interests are completely different from the labeled data. Values between 0.0 to 1.0 mean that the detected interests are partially correct or ordered incorrectly. Here, we evaluate NDCG@1, NDCG@2, and NDCG@3. The algorithm can generate a ranked label list. We sort the labels according to the number of votes from the survey as ideal ranking list. We find that most of the algorithms can predict good labels for a topic. In In particular, if we consider giving two labels for a topic, our proposed method

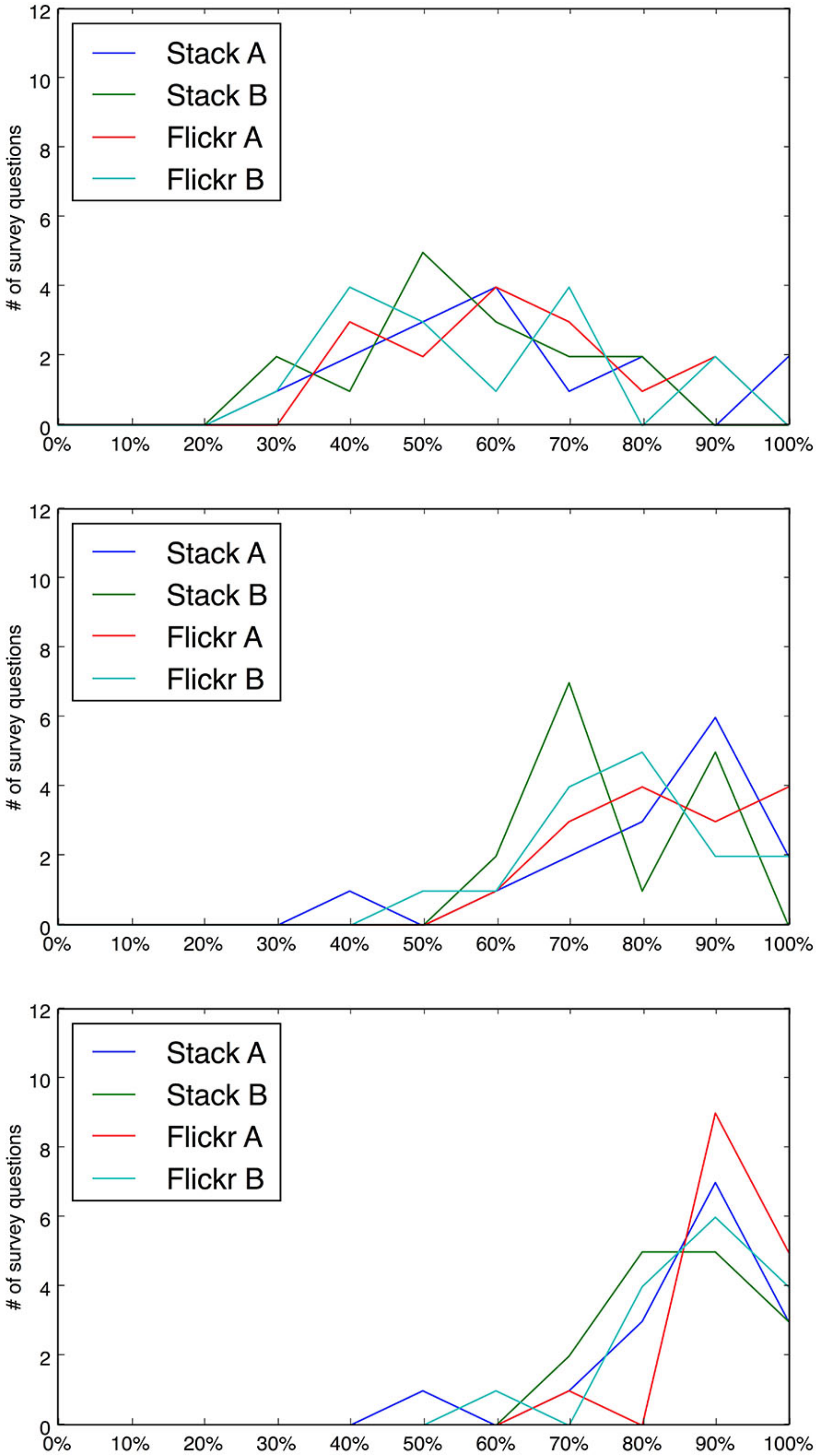


Figure 6.6: The proportion as a function of the top X voted labels

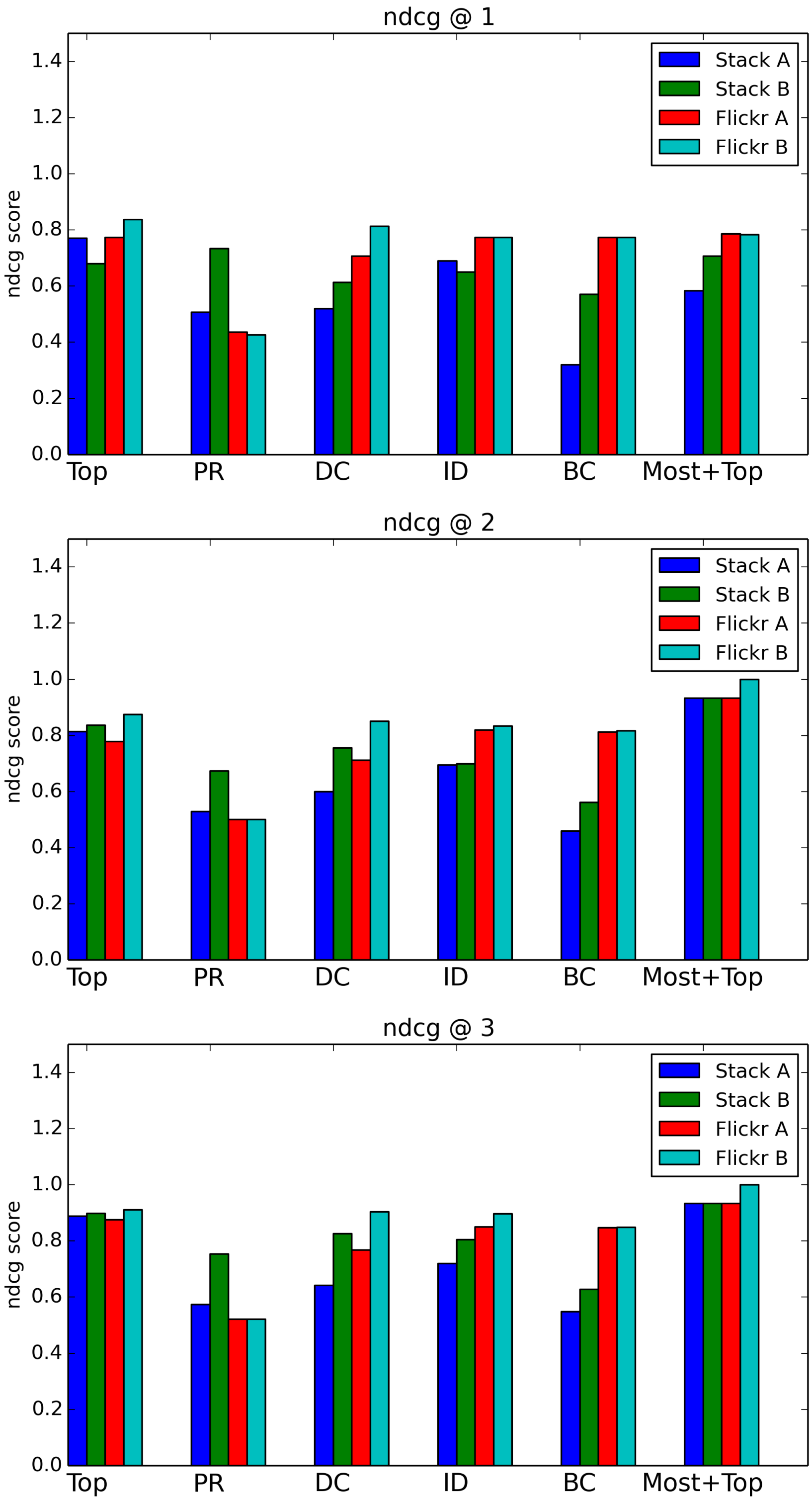


Figure 6.7: NDCG score at position X

"Most+Top" has very good results on all the datasets, which means for all the topics, the method can generate two good labels to represent the meaning of the topic.

6.4 Summary: representing a topic with labels

In this chapter, we discussed how we used DBpedia as external knowledge to help with choosing labels to turn bags of words into meaningful topics. From the user survey we found that users can reach a good agreement on composite labels. Therefore, it is more reasonable to have more than one keyword to label the bag of words of a topic. We also proposed a hybrid solution by combining results from different algorithms to generate composite labels to represent a topic.

In the next chapter, we will focus on how to extract more sophisticated social information such as expertise, activity and trends.

CHAPTER 7

Temporal Topic Expertise Activity (TTEA)

Contents

7.1 Introduction: Mining expertise and temporal information

Chapter 3 proposed a method to formalize the latent information in user-generated content. The key point was how to extract this information. Chapter 4 introduced the use of the original LDA model to extract topics and communities. In this chapter, we extend the results of Chapter 4 to extract topic based expertise and topic based temporal knowledge.

Let us consider StackOverflow for an example of the problem we address. In StackOverflow, for instance, *Alice* posts a question at *08/11/2015*, and adds the tags {*html*, *css*, *height*}. Her question then gets *30* votes, and *Bob* gives an answer to this question at *10/11/2015*, that gets a voting score of *35*. Based on this original information, we want to propose a model to extract more latent information from it.

7.1.1 Joint extraction of topics, trends, expertise, and activities

The Temporal Topic Expertise Activity (TTEA) model we propose aims towards jointly modeling topics, their trends, users' expertise, and their activities. More precisely, we aim at extracting the indicators listed in Table 7.1.

Table 7.1: Output distributions of our model and their functionality

Notation	*Functionality of distribution*
θ_{uk}	detect the topic a user is most interested in
θ_{ku}	detect the most active users for a topic
θ_{kv}/θ_{kw}	detect the most relevant tags/words in a topic
θ_{kt}	detect the trends of a topic
θ_{tk}	detect the most popular topic at point in time
θ_{ukt}	detect a user's activity pattern in a topic
θ_{uke}	detect the topic a user has most expertise in

7.1.2 Fundamental Notions in Defining a TTEA

Let us now define the basic notions later used in the description of TTEA:

Topic (θ_{kw}/θ_{kv}): A bag of words or tags which are closely related. Words are the content of questions or answers, tags are attached to questions. For example, the topic-tag distribution *Database*:$\{$*mysql*: 0.5, *sql*: 0.3, *query*: 0.2$\}$ expresses that topic *Database* is related to tags *mysql*, *sql*, and *query*.

User Topical Interest(θ_{uk}): A user is interested in different topics with different levels of interest. For example, the user-topic distribution *Alice*:$\{$*Database*: 0.8, *Java*: 0.2$\}$ expresses that *Alice* prefers to answer questions related to *Database*, but less so about *Java*.

User Topical Activity(θ_{ku}): Different users are interested in the same topic with different levels. For example, the topic-user distribution *Database*:$\{$*Alice*: 0.8, *Bob*: 0.2$\}$ expresses that *Alice* prefers to answer question related to *Database*, while *Bob* is less willing to contribute answers to it.

Topic Trend(θ_{kt}): A topic is popular at different points in time with different levels. For example, the topic-time distribution *Database*:$\{$*May/2013*: 0.2, *June/2013*: 0.3, *July/2013*: 0.5$\}$ expresses that the topic *Database* is increasingly popular.

Topic Temporal Activity(θ_{tk}): Topics are active at a point in time with different levels. For example, the time-topic distribution *Sept/2013*:$\{$*Ios*: 0.8, *Database*: 0.2$\}$ expresses that *ios* related questions are popular in Sept. 2013, while *Database* related questions are not specially popular.

User Topic Temporal Dynamics(θ_{ukt}): A user is interested in different topics at different points in time with different levels. For example, the topic-time distribution for *Alice* *ios*:$\{$*May/2013*: 0.2, *June/2013*: 0.3, *July/2013*: 0.5$\}$ expresses that *Alice*'s interest to topic *ios* is increasing.

User Topical Expertise(θ_{uke}): A user has expertise in different topics with different levels. For example, the topic-expertise distribution for *Alice ios*:$\{$*High*: 0.2, *Medium*: 0.7, *Low*: 0.1$\}$ expresses that *Alice*'s expertise on topic *ios* is probably in medium level.

7.2 TTEA Model and Computation

7.2.1 TTEA Probabilistic Graphical Model

The TTEA model we propose is based on LDA. Figure 7.1 represents it using the plate notation. The original LDA model is in red with dotted line style, and our extension is in blue with solid line style. Compared with original LDA, we not only model the word (W_i) in a post, but also model the tag (Ta_i), time (Ti_i), vote (V_i) to extract temporal and expertise information all together. Let $u_i \in \{1, 2, ..., U\}$ be the set of users, $p_i \in \{1, 2, ..., P\}$ the set of answer posts, which are generated by these users, $w_i \in \{1, 2, ..., W\}$ the set of words in answers posts, $ta_i \in \{1, 2, ..., Ta\}$ the set of tags which are attached to posts, $v_i \in \{1, 2, ..., V\}$ the set of votes for each answer post, $ti_i \in \{1, 2, ..., Ti\}$ the set of points in time which could be months or days depending on the requirements, and $z_i \in \{1, 2, ..., K\}$ the set of topics for the posts. Here, U, P, W, Ta, V, Ti and K denote the total number of users, posts, words, tags, votes, points in time, and topics. α, β, δ, γ, η, and λ are Dirichlet priors. The notation and description of distributions θ_{uk}, θ_{kv}, θ_{kw}, θ_{kt}, and θ_{uke} are listed in Table 7.1.

Contrary to (Blei 2003) who applied LDA model on long documents such as news articles and assumed that each word has a latent topic, we assume in TTEA that each answer post has one topic: like in other social media with short contributions, e.g. Twitter, an answer post is normally short, each answer post is therefore suitable to be assigned with one single latent topic, and all the words in that post are considered to be generated by this topic. Some work (Zhao 2011)(Diao 2012) on microblog also made this assumptions.

For expertise modeling, we do not use votes directly because (a) the vote scores are sparse and noncontinuous, and (b) it is not reasonable to say that a vote score of 55 is better than a vote score of 50 if the vote scores are ranging from 0 to 5000. Since the vote scores' counts distribution follows a log distribution (Yang 2013b), we use the logarithmic value of vote score, and separate them into several expertise levels, which is one of the parameters: the expertise level.

For temporal modeling, like (Wang 2006) (Hu 2014), we use time stamps directly. In

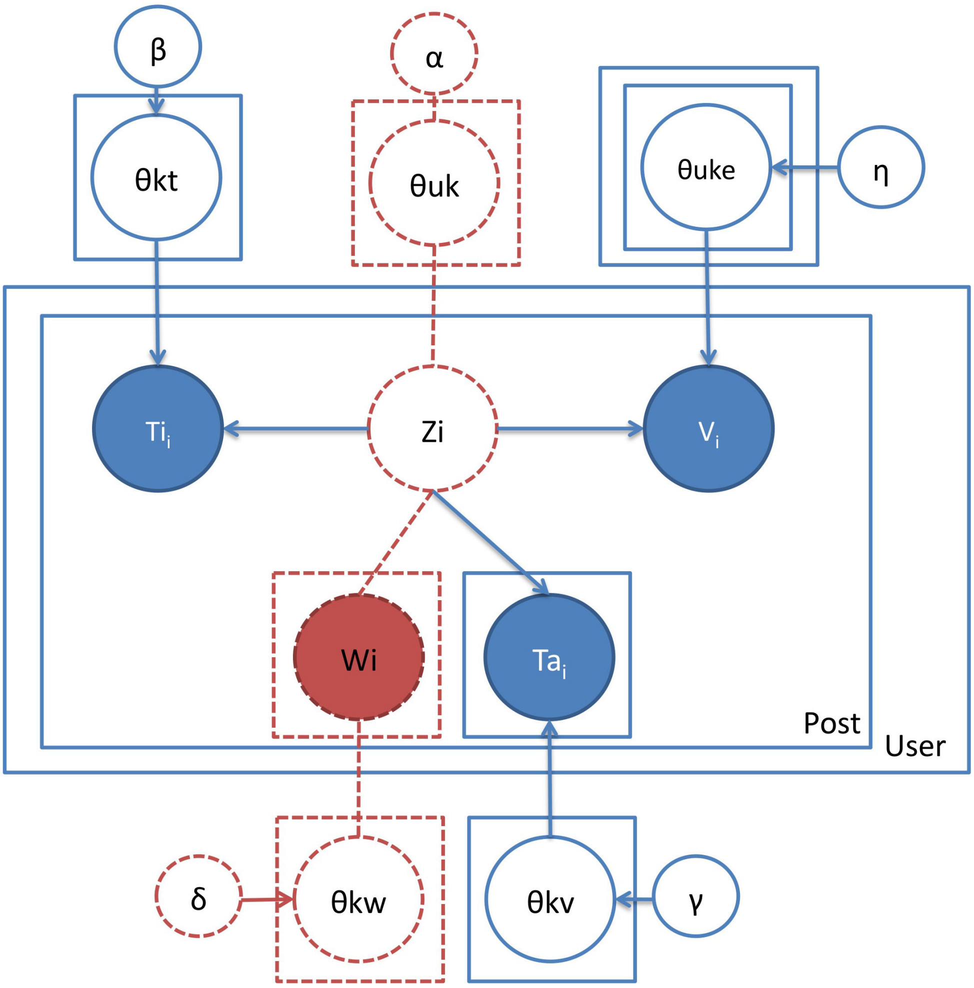


Figure 7.1: TTEA Model

order to model time at different levels, we simply split time stamps into different parts (month, day, and hour) and use them separately depending on the demands.

The generative process of TTEA model is: Let us consider a user u who wants to answer a question. She first selects a topic k according to her user-topic distribution θ_{uk}. Then she writes an answer post p. The words of p are generated from topic k's topic-word distribution θ_{kw}. Since only the questions have tags, we consider that the answers automatically acquire all the tags of the question they respond to. Then the answer post p acquires its tags according to the topic-tag distribution θ_{kv} of topic k. Meanwhile, the answer post p gets a time-stamp ti according to the topic-time distribution θ_{kt} of topic k. This procedure is described as follows:

*/*The generative process*/*
for the u-th user u **in** U **do**
  draw user topic distribution $\theta_{uk} \sim$ Dir(α)
end for
for the k-th topic k **in** K **do**
  draw topic tag distribution $\theta_{kv} \sim$ Dir(γ)
  draw topic word distribution $\theta_{kw} \sim$ Dir(δ)
  draw topic time distribution $\theta_{kt} \sim$ Dir(β)
end for
for the u-th user u *in* U **do**
  for the k-th topic k *in* K **do**
    draw user topic expertise distribution $\theta_{uke} \sim$ Dir(η)
  end for
end for
for the u-th user u *in* U **do**
  for the n-th q&a post p *in* P **do**
    draw topic $z \sim$ Multi(θ_{uk})
    draw time point $t \sim$ Multi(θ_{kt})
    draw expertise level $v \sim$ Multi(θ_{uke})
    for the i-th word w *in* W **do**
      draw word $w \sim$ Multi(θ_{kw})
    end for
    for the j-th tag ta *in* Ta **do**
      draw tag $t \sim$ Multi(θ_{kv})
    end for
  end for

7.2.2 TTEA Model Inference: using collapsed gibbs Sampling

Like (Hu 2014), we use the collapsed Gibbs Sampling algorithm (Griffiths 2004) to sample the hidden variable z, based on which the unknown probabilities $\{\theta_{uk}$, θ_{kv}, θ_{kw}, θ_{kt}, and $\theta_{uke}\}$ can be estimated.

The TTEA inference process is as follows. We iteratively sample the topic indicator z_i for each answer post p_i according to equation 7.1. The intuition behind this equation is to combine two parts of possibilities: (1) the possibilities to generate the topic indicator z_i and (2) the possibilities generated by the topic indicator z_i. Also, the intuition behind each part in Equation 7.1 are corresponding to Equations 7.2, 7.3, 7.4, 7.5 and 7.6. As explained before, each question/answer post will have one topic assignment.

$$\begin{aligned} & p(z_i = k | z_{\neg i}, \mathbf{U}, \mathbf{Ti}, \mathbf{Ta}, \mathbf{W}) \\ & \quad \propto \frac{C^k_{u,\neg i} + \alpha_1}{\sum_{k=1}^{K} C^k_{u,\neg i} + K * \alpha_1} \\ & \quad \cdot \frac{\prod_{ta=1}^{Ta} \prod_{q=0}^{C_{ta}-1} (C^{ta}_{k,\neg i} + q + \gamma)}{\prod_{p=0}^{\sum C_{ta}-1} \sum_{ta=1}^{Ta} (C^{v}_{k,\neg i} + p + Ta * \gamma)} \\ & \quad \cdot \frac{\prod_{w=1}^{W} \prod_{s=0}^{C_w-1} (C^{w}_{k,\neg i} + s + \delta)}{\prod_{t=0}^{\sum C_w-1} \sum_{w=1}^{W} (C^{w}_{k,\neg i} + t + W * \delta)} \\ & \quad \cdot \frac{C^{ti}_{k,\neg i} + \beta}{\sum_{ti=1}^{Ti} C^{ti}_{k,\neg i} + Ti * \beta} \\ & \quad \cdot \frac{C^{e}_{u,k,\neg i} + \eta}{\sum_{e=1}^{E} C^{e}_{u,k,\neg i} + E * \eta} \end{aligned} \tag{7.1}$$

where $\neg i$ enforces that all the counters used are calculated with the answer post p_i excluded. $C^k_{u,\neg i}$ is the number of posts by user u assigned to topic k, C_{ta} is the number of tags ta in p_i, therefore, $\sum C_{ta}$ is the total number of tags in p_i, $C^{ta}_{k,\neg i}$ is the number of tags ta assigned to topic k. Similarly, C_w is the number of words w in p_i, $\sum C_w$ is the number of words in p_i, $C^{w}_{k,\neg i}$ is the number of words w assigned to topic k. $C^{ti}_{k,\neg i}$ is the number of posts assigned to topic k and posted at time ti. $C^{e}_{u,k,\neg i}$ is the number of posts which are assigned to topic k and got a vote score in the range of expertise level e.

Then, with the result of the Gibbs sampling algorithm, we can make the following parameter estimation:

$$\theta_{uk} = \frac{C_u^k + \alpha}{\sum_{k=1}^{K} C_u^k + K * \alpha} \tag{7.2}$$

$$\theta_{kv} = \frac{C_k^{ta} + \gamma}{\sum_{ta=1}^{Ta} C_k^{ta} + Ta * \gamma} \tag{7.3}$$

$$\theta_{kw} = \frac{C_k^w + \delta}{\sum_{w=1}^{W} C_k^w + W * \delta} \tag{7.4}$$

$$\theta_{kt} = \frac{C_k^{ti} + \beta}{\sum_{ti=1}^{Ti} C_k^{ti} + Ti * \beta} \tag{7.5}$$

$$\theta_{uke} = \frac{C_{u,k}^e + \eta}{\sum_{e=1}^{E} C_{u,k}^e + E * \eta} \tag{7.6}$$

7.2.3 Post Processing: Extracting activity indicators

The previous model can only generate the distributions {θ_{uk}, θ_{kv}, θ_{kw}, θ_{kt}, and θ_{uke} }. To generate the other distributions, e.g. θ_{ku}, θ_{tk} and θ_{ukt}, we directly use the sample results at each iteration and keep recording the corresponding counters. Therefore, C_k^u is the number of posts assigned to topic k and posted by user u, C_{ti}^k is the number of posts posted at time ti and assigned to topic k. $C_{u,k}^{ti}$ is the number of posts by user u, assigned to topic k and posted at time ti. Then, we estimate θ_{ku}, θ_{tk}, θ_{ukt} according to the following equations:

$$\theta_{ku} = \frac{C_k^u + \alpha_2}{\sum_{u=1}^{U} C_k^u + U * \alpha_2} \tag{7.7}$$

$$\theta_{tk} = \frac{C_{ti}^k + \beta_1}{\sum_{k=1}^{K} C_{ti}^k + K * \beta_1} \tag{7.8}$$

$$\theta_{ukt} = \frac{C_{u,k}^{ti} + \lambda}{\sum_{ti=1}^{T} C_{u,k}^{ti} + T * \lambda} \tag{7.9}$$

7.3 TTEA Model Experiments and Evaluation on StackOverflow data

7.3.1 Basic statistic of StackOverflow Dataset: an overview

We conducted experiments on a dataset from StackOverflow. This site releases its whole content every three months. For our experiments, we used the data dump from July 2008 to March 2013.

Table 7.2 and Figure 7.2 provide basic statistics on the dataset.

Table 7.2: Basic statistics on the dataset

number of tags	32,379
number of questions	4,592,961
number of users asking questions	833,041
number of users providing answers	8,585,113
number of questions having accepted answers	2,808,825

Here are some general observations about the dataset:

- nearly half of the questions do not have accepted answers;
- nearly half of the questions only have one answer and it may be inadequate;
- more than a third of the questions only have one or two tags;
- nearly half of the users only answer one question so question routing and incentives are important problems;
- nearly 10% percent of the questions do not have answers.

7.3.2 Experiment Dataset and Compared Methods

In the experiments described in Chapter 5, we only used a part of this data set (from 2008 to 2009), and we mainly focused on several co-answer graphs. In addition, we also labeled this small dataset. Considering the large volume of the dataset over 3 years, the processing time is extremely long. (Wei 2006) shows that the complexity of each iteration of the

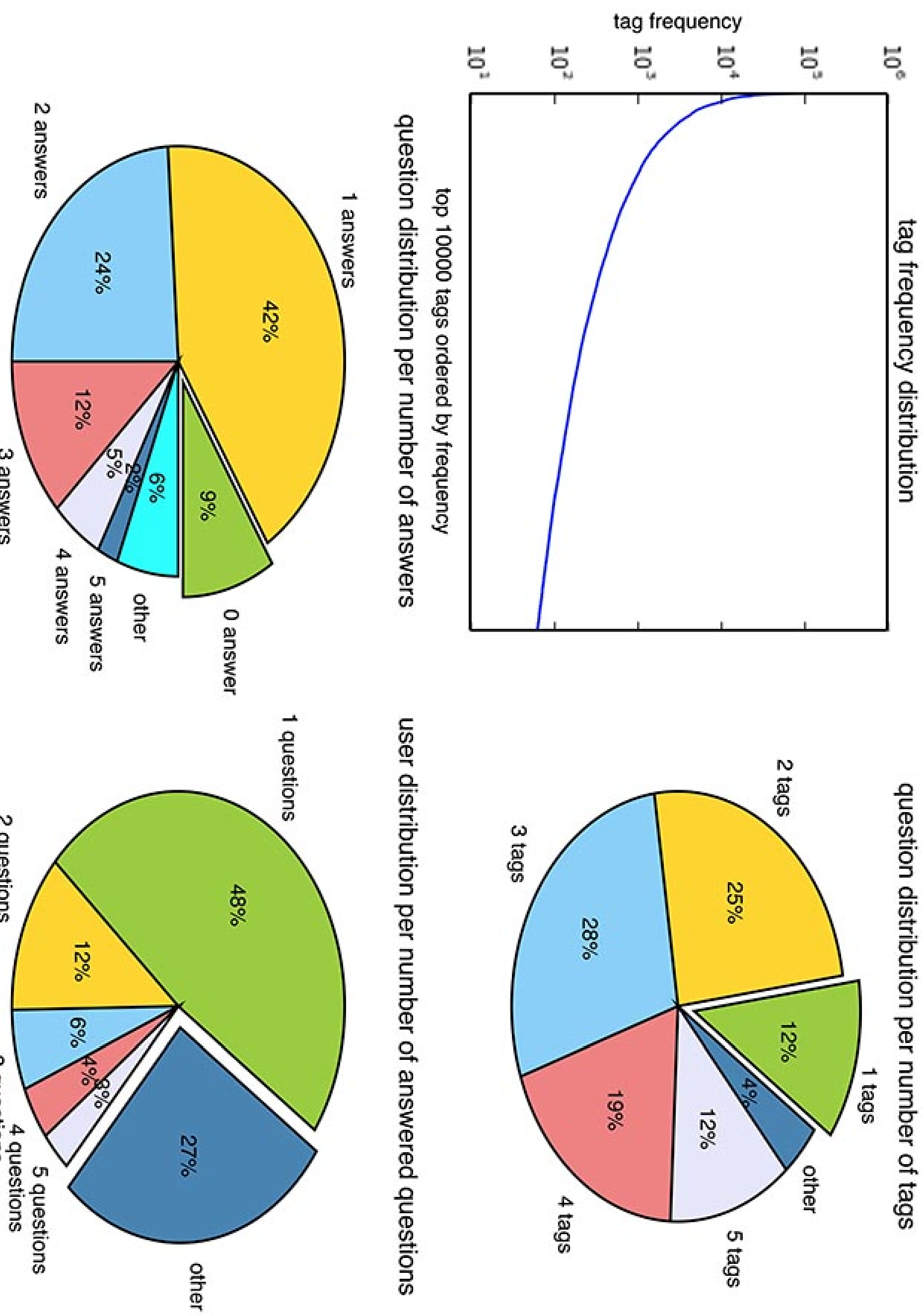


Figure 7.2: Basic perspectives of the dataset

Gibbs sampling for LDA is linear with the number of topic and the number of documents, which is $O(KN)$. In the experiments described in this chapter and aiming at evaluating the effectiveness of our model, in order to simplify the processing, we chose two continuous months from the dataset (From Jan 2011 to June 2011, from July 2011 to Jan 2012), with no bias to the selections.

To evaluate the effectiveness of our model, we compared it with several related works:

- TTEA is our method for modeling user, topic, temporal and expertise in Q&A sites. In addition, we also model activities by adding virtual nodes. We can generate the user-topic distribution and topic-activity distribution simultaneously.

- TEM: (Yang 2013b) proposed a model for users, topics and expertise in Q&A sites. It integrates a Gaussian Mixture Model to model expertise, which is time consuming. We simplify this process by directly modeling votes information. Also, it does not model temporal information and user topic activities.

- UQA: (Guo 2008b) proposed a User-Question-Answer model for modeling users and topics in Q&A sites. In certain Q&A sites, questions have category information which have proved to be very useful. The category in their model is similar to tags in TTEA model and TEM model. However we allow multiple tags for each posts while they can only set a single category.

- GrosToT: (Hu 2014) proposed a User-Group-Topic-Time model for modeling users, groups, topics and time in social media sites. It introduces a group level between users and topics compared with other models. It does not directly generate user-topic distributions, it computes it as the user-group distribution and group-topic distribution.

- LDA: based on (Blei 2003) we apply the LDA model to create a User-Topic-Post model for modeling users and topics. It can generate the user-topic distribution and topic-words distribution.

We chose the same number of topics K=*30* as (Chang 2013) and the same number of expertises E=*10* as (Yang 2013b), which have proved to be a reasonable setting for the StackOverflow dataset. We empiricaly set the Dirichlet hyper parameters α_1=α_2=50/K, β_1=β_2=0.01, δ=λ=η=0.01, γ=0.001 according to suggestions in (Griffiths 2004).

7.3.3 Performance of Topic Extraction: perplexity score

In Chapter 5, we have evaluated the perplexity score for both the TTD and LDA models. The evaluation aimed to check whether or not our model can have a similar or better performance on topic extraction than the much more complicated probabilistic graphical model. In this section, we re-evaluate the perplexity score only among those probabilistic graphical models as our TTEA model is a probabilistic graphical model. Also, we evaluate on a much larger dataset compared with Chapter 5.

Table 7.3 and Table 7.4 show the top tags and words detected by our model. We use again the Perplexity (Blei 2003) metric as a quantitative way to measure the performance of topic extraction.

We include in our training dataset all the posts in the two months from August 1^{st} 2011 to October 1^{st} 2011, from users having more than 80 posts (as in (Yang 2013b)). The resulting training dataset contains 87516 Q&A posts by 674 users. For data preprocessing, we tokenized the texts and removed the stop words. For the testing dataset, we used all the posts of the same set of users than the training data but this time from October 1^{th} 2011 to January 1^{th} 2012. The training and testing datasets have no overlap but concern the same community. We varied the number of topics: 10, 30, 50, and 100. For a testing set of M posts, N_i denotes the number of words in the i^{th} post and the Perplexity score is computed according to equation 7.10.

$$Perplexity(D_{test}) = exp\left\{-\frac{\sum_{i=1}^{M}\log p(W_i)}{\sum_{i=1}^{M}N_i}\right\} \tag{7.10}$$

where $p(W_i)$ is the probability of the words in the test document d_i. In our model, $p(W_i)$

is computed according to equation 7.11.:

$$p(W_i) = \sum_k \theta_{u_i k} \prod_w \theta_{k w_i} \tag{7.11}$$

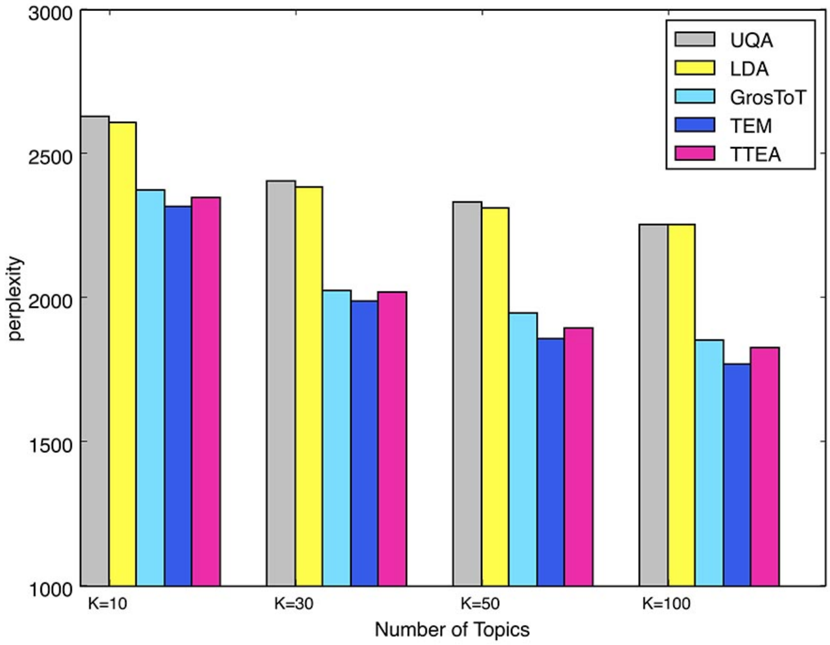


Figure 7.3: Comparison of topic extraction performances

Figure 7.3 shows the perplexity results for our TTEA method and other state-of-the-art methods. TTEA is almost as good as TEM. However TEM integrates a Gaussian Mixture Model, which is time consuming. The training process of TEM is nearly three times longer than the other models.

Table 7.3: Top tags for different topics generated by the TTEA model

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
php	c	iphone	c++	javascript	android	sql	java	jquery	git
xslt	.net	objective-c	c	jquery	java	mysql	spring	javascript	svn
xml	linq	ios	pointers	php	android-layout	sql-server	eclipse	html	version-control
xpath	generics	xcode	templates	ajax	listview	php	jsp	css	github
mysql	asp.net	cocoa-touch	stl	html	activity	query	.htaccess	jquery-selectors	mercurial
html	vb.net	ipad	arrays	json	android-intent	tsql	servlets	jquery-ui	eclipse
arrays	c-4.0	uitableview	vector	asp.net	sqlite	sql-server-2008	jsf	dom	tortoisesvn
jquery	reflection	iphone-sdk-4.0	string	jquery-ajax	layout	join	mod-rewrite	php	linux
javascript	entity-framework	cocoa	function	forms	android-widget	select	maven	javascript-events	clearcase
foreach	list	xcode4	c++11	asp.net-mvc-3	xml	sql-server-2005	apache	ajax	ssh

Table 7.4: Top words for different topics generated by the TTEA model

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
xsl	aspx	view	std	jquery	android	select	html	jquery	git
td	msdn	reference	const	ajax	activity	join	java	div	branch
tr	microsoft	nsstring	pointer	script	html	group	file	click	commit
template	library	apple	char	javascript	view	order	spring	element	file
select	select	html	template	page	developer	table	jar	event	svn
row	linq	library	vector	html	intent	key	apache	input	repo
echo	system	documentation	operator	form	reference	count	eclipse	document	repository
table	dictionary	developer	compiler	url	layout	row	docs	text	files
match	ienumerable	ios	memory	document	try	inner	servlet	html	master
node	expression	release	struct	json	button	query	web	api	github

7.4 Task Evaluation: Question routing and Expert recommendation

7.4.1 Question Routing: recommending new questions to potential users

(Chang 2009) suggested that topic models should focus on evaluations on real-world task performance rather than on optimizing likelihood-based measures. So, in addition to the perplexity-based evaluation, we used the results of TTEA to perform real-world tasks and we evaluated them. This is described in this subsection and the following ones. In this section we focus on question routing: given a question q and a set of users U, the task is to rank all these users by their interests to answer question q. We score each user u by considering the similarity between his topics of interest and the topics of the question ($Sim(u, q)$). The intuition behind equation 7.12 is that the more a user is interested in the topic of a question, the more likely he is to provide an answer to that question.

$$Sim(u, q) = (1 - JS(\theta_{uk}, \theta_{qk})) \tag{7.12}$$

where θ_{uk} is the user topic interest distribution, θ_{qk} is the question topic distribution, and JS(.) is the Jensen-Shannon divergence distance. We obtain θ_{uk} directly from model results. For θ_{qk}, we apply equation 7.13.

$$\begin{aligned} \theta qk &\propto p(k|w_q, t_q, u) \\ &= p(k|u)p(w_q|k)p(t_q|k) \\ &= \theta uk \sum_{w_i \in w_q} \theta_{kw_i} \sum_{t_i \in t_q} \theta_{kv_i} \end{aligned} \tag{7.13}$$

where w_q and t_q are the sets of all the words and tags in question q and θkw, θkv are the topic-word distribution and topic-tag distribution obtained directly from the model result (e.g. θ_{kw_i} denotes the topic-word distribution of topic k to word i, θ_{kv_j} denotes the topic-tag distribution of topic k to tag j). Then for question q, we compute the Sim score for user set U and rank them in decreasing order.

We used all the posts from July 1^{th} 2011 to October 1^{th} 2011 from users having more than 50 Q&A posts for the training dataset. Rather than using the threshold of 80 post likes in (Yang 2013b), we empirically set it to 50 posts to get enough users for recommendation. The resulting training set contains 297881 posts by 2555 users. For the testing dataset, we used all the questions posted by the same set of users as in the training set but this time from October 1^{th} 2011 to January 1^{th} 2012. Therefore the training and testing datasets have no overlaps. We removed testing questions which have no, or only one, answer. The resulting test dataset contains 6044 questions, 18077 answers and 7888 involved users.

We also chose another period for this experiment. Besides, we varied the number of topics by 15 and 50, we varied the filter limit by 40 and 80. The experimental results are shown in section 7.4.2.

In order to evaluate different models, we considered precision at position N (Precision@N or simply P@N) and recall at position N (Recall@N or simply R@N), which are widely used measures in the Information Retrieval community. Let R_q be the recommendations of users for a question q and U_q be the actual set of users who posted for question q. Then Precision@N is defined in equation 7.14 and Recall@N is defined in equation 7.15.

$$P@N = \frac{1}{|Q|} \sum_{q \in Q} \frac{|R_q \cap U_q|}{|R_q|} \tag{7.14}$$

$$R@N = \frac{1}{|Q|} \sum_{q \in Q} \frac{|R_q \cap U_q|}{|U_q|} \tag{7.15}$$

where Q is the set of testing questions. Like in (Chang 2013), we use the Matching Set Count (MSC) which is defined in equation 7.16. The idea is to count the number of successful recommendations, i.e., for which at least one of the recommended users answered the question.

$$MSC@N = \frac{1}{|Q|} \sum_{q \in Q} 1[R_q \cap U_q \neq \emptyset] \tag{7.16}$$

where $1[condition]$ is equal to 1 if $condition$ is true, otherwise 0.

In addition, our model can capture activity and we believe this information improves

Table 7.5: Question Routing experiments, Random denotes that we randomly recommend users for the test questions.

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.024	0.019	0.015	0.013	0.045	0.072	0.111	0.142	0.112	0.178	0.269	0.339
TTEA-ACT	0.028	**0.022**	**0.017**	**0.014**	0.052	**0.083**	**0.127**	**0.159**	0.134	**0.209**	**0.313**	**0.382**
TEM	0.024	0.019	0.015	0.013	0.045	0.073	0.114	0.146	0.114	0.179	0.275	0.344
TEM-ACT	0.029	**0.023**	**0.018**	**0.015**	0.054	**0.084**	**0.129**	**0.162**	0.137	**0.210**	**0.315**	**0.388**
UQA	**0.030**	0.019	0.012	0.010	**0.062**	0.075	0.095	0.112	**0.149**	0.179	0.224	0.261
GROSTOT	0.027	0.017	0.011	0.009	0.055	0.067	0.085	0.099	0.134	0.164	0.204	0.236
RANDOM	0.001	0.001	0.001	0.001	0.001	0.002	0.005	0.007	0.003	0.007	0.013	0.019

question routing. The intuition is that even if a user has a high Sim score for a question, the less he is active, the less likely he is to provide an answer to that question. Therefore, we defined a score $SimAct$ to combine both topic similarity and activity level as shown in equation 7.17, where $Act(u, q)$ is the computed activity score for user u to question q. A high value of the Act score indicates a high probability of activity on a question. We use TTEA to denote the method using only the similarity information, that is to say, ranking users by Sim score. We use TTEA-ACT to denote the method using both similarity and activity, that is to say, ranking users by $SimAct$ score. We also integrated our activity model to the TEM model and we refer to it as TEM-ACT.

$$\begin{aligned} SimAct(u, q) &= (1 - JS(\theta_{uk}, \theta_{qk})) * Act(u, q) \\ &= (1 - JS(\theta_{uk}, \theta_{qk})) * \sum_{k=1}^{K} \theta_{qk} * \theta_{ku} \end{aligned} \tag{7.17}$$

Table 7.5 shows the results. We ran the experiments five times and listed the average scores. Our observations can be summarized as follows:

- UQA and GROSTOT perform better when the number of recommended users is small, and TTEA and TEM begin to outperform UQA and GROSTOT when the number of recommended users is large;
- TTEA-ACT shows the best performances compared with the baseline competitors;
- both TTEA-ACT and TEM-ACT perform better than the other models. The activity modeling is a generic method that could improve the performance not only of our model, but also of other models although here we only show the result for the activity model with TEM as an example;
- even when TEM or TEM-ACT perform better than our model they still remain time consuming. Experiments show that the training process takes around 3-4 times longer compared to our model.

7.4.2 Experiment Parameter Sensitivity Analysis

The above experiments have shown the effectiveness of our model. However, we used some arbitrary parameters. In this section we show the results obtained when varying the experiment settings and we analyse the sensitivity of the parameters.

- we use posts from another period of time.
- we vary topic number by 15, 50. We used 30 in previous experiments.
- we vary the filter threshold by 40, 80. This threshold equal to 60 means that a user is ignored if she has less than 60 posts. We used 60 in previous experiments.

For the training dataset, we used all the posts in a three month period, from January 1^{th} 2011 to March 31^{th} 2011, from users having at least 50 Q&A posts, rather than 80 posts like (Yang 2013b), in order to get enough users for recommendations. The training set contains 371181 posts by 3123 users. For the testing dataset, we used all the questions posted by the same set of users as in the training set, but this time from April 1^{th} 2011 to June 31^{th} 2011. Therefore the training and testing datasets have no overlaps. We removed questions with no or only one answer. The resulting test dataset contains 9048 questions, 27870 answers and 10147 users. Table 7.6 shows the question routing results. We still find that TTEA-ACT outperforms all the baseline models. Besides, both TTEA-ACT and TEM-ACT outperform all the other models.

Table 7.7 shows the question routing results with the number of topics set to 15. We use the same training and testing datasets as in section 7.4.1.

Table 7.8 shows the question routing results for the number of topics set to 50. We use the same training and testing datasets as in section 7.4.1.

Table 7.9 shows the question routing results with users having more than 40 posts. We use the same period of dataset used in section 7.4.1. Due to the different filter limit, the training set contains 3457 users and 338485 Q&A posts, the testing set contains 8579 questions, 25500 answers and 10135 involved users.

Table 7.6: Question Routing experiments on another Dataset

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.026	0.020	0.015	0.013	0.047	0.073	0.110	0.136	0.123	0.186	0.273	0.332
TTEA-ACT	**0.032**	**0.026**	**0.019**	**0.016**	**0.058**	**0.093**	**0.137**	**0.168**	**0.153**	**0.236**	**0.339**	**0.405**
TEM	0.025	0.021	0.016	0.013	0.047	0.076	0.112	0.139	0.120	0.191	0.274	0.333
TEM-ACT	**0.032**	**0.025**	**0.020**	**0.016**	**0.058**	**0.092**	**0.141**	**0.171**	**0.153**	**0.235**	**0.348**	**0.411**
UQA	0.027	0.016	0.011	0.009	0.052	0.062	0.080	0.096	0.130	0.155	0.196	0.233
GROSTOT	0.023	0.014	0.009	0.007	0.044	0.055	0.069	0.081	0.112	0.137	0.172	0.200
RANDOM	0.001	0.001	0.001	0.001	0.001	0.002	0.004	0.005	0.003	0.005	0.010	0.015

Table 7.7: Question Routing experiments with 15 topics

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.016	0.013	0.012	0.010	0.030	0.050	0.086	0.112	0.076	0.127	0.213	0.269
TTEA-ACT	0.023	**0.018**	**0.015**	**0.012**	0.042	**0.066**	**0.107**	**0.134**	**0.112**	**0.170**	**0.268**	**0.329**
TEM	0.017	0.015	0.012	0.010	0.032	0.054	0.091	0.115	0.083	0.137	0.222	0.276
TEM-ACT	0.024	**0.018**	**0.014**	**0.012**	0.043	**0.068**	**0.103**	**0.131**	**0.114**	**0.172**	**0.254**	**0.319**
UQA	**0.028**	0.016	0.011	0.008	**0.056**	**0.066**	0.083	0.099	0.137	0.159	0.199	0.238
Grostot	0.023	0.015	0.010	0.008	0.045	0.058	0.075	0.089	0.112	0.143	0.183	0.216
Random	0.001	0.001	0.001	0.001	0.002	0.003	0.004	0.006	0.005	0.008	0.012	0.017

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.028	0.023	0.018	0.015	0.054	0.087	0.132	0.168	0.134	0.215	0.319	0.394
TTEA-ACT	**0.033**	**0.025**	**0.019**	**0.016**	0.063	**0.095**	**0.142**	**0.178**	**0.158**	**0.235**	**0.343**	**0.418**
TEM	0.029	0.024	0.018	0.015	0.056	0.088	0.136	0.171	0.141	0.220	0.325	0.400
TEM-ACT	**0.033**	**0.026**	**0.020**	**0.017**	0.062	**0.096**	**0.145**	**0.182**	**0.157**	**0.240**	**0.347**	**0.427**
UQA	0.032	0.019	0.012	0.010	**0.065**	0.077	0.097	0.116	**0.158**	0.185	0.227	0.270
Grostot	0.028	0.017	0.011	0.009	0.056	0.067	0.088	0.102	0.136	0.163	0.210	0.241
Random	0.001	0.001	0.001	0.001	0.002	0.002	0.005	0.007	0.004	0.006	0.013	0.018

Table 7.8: Question Routing experiments with 50 topics

Table 7.9: Question Routing experiments, with users having more than 40 posts

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.021	0.018	0.014	0.012	0.040	0.067	0.104	0.132	0.100	0.167	0.253	0.313
TTEA-ACT	0.026	**0.021**	**0.016**	**0.014**	0.049	**0.076**	**0.118**	**0.149**	0.126	**0.193**	**0.292**	**0.360**
TEM	0.023	0.018	0.014	0.012	0.043	0.069	0.106	0.137	0.109	0.170	0.255	0.323
TEM-ACT	0.027	**0.021**	**0.016**	**0.014**	0.050	**0.078**	**0.121**	**0.152**	0.128	**0.194**	**0.295**	**0.362**
UQA	**0.029**	0.018	0.011	0.009	**0.059**	0.071	0.087	0.101	**0.142**	0.169	0.205	0.235
Grostot	0.025	0.016	0.010	0.008	0.050	0.063	0.077	0.091	0.122	0.152	0.188	0.217
Random	0.000	0.000	0.000	0.000	0.001	0.002	0.003	0.005	0.002	0.004	0.008	0.013

Table 7.10 shows the question routing results with users having more than 80 posts. We use the same period of dataset used in section 7.4.1. Due to the different filter limit, the training set contains 1275 users and 216940 q&a posts, the testing set contains 2589 questions, 8006 answers and 4196 involved users.

From the above experiments on another dataset chosen with another period of time, we can conclude that our model have consistently the best performance . The performance increases when the number of topics increases. This can be explained by the fact that when the number of topics increases, the words in a topic are more concentrated. On the other hand, when the number of topics increases, many generated topics are actually very similar, and the execution time increases. The performance increases means we keep more active users by increasing the filter threshold, which is the minimum number of posts per user. There will be more active users as question routing candidates. In other words, with a high filter threshold, we get a small set of users as recommendation candidates, but these users are very active (contributing to many posts). Conversely, with a low filter threshold, we get a large set of users as recommendation candidates, but some of them may be not very active (contributing to view posts).

7.4.3 Recommendation of expert users: topic based expertise

Given a question q and a set of users U, the task is here to recommend N users until one of the users gets the highest vote. The point is to rank recommended users by their expertise to answer question q. We score each user u by considering the similarity $SimExp(u, q)$ between a user's topic interest and a user's topic expertise to answer question q. The intuition behind equation 7.18 is that if the user is interested in the question, she will probably provide an answer to that question and if the user has expertise about the question, the answer will probably have the highest vote score.

$$SimExp(u, q) = (1 - JS(\theta_{uk}, \theta_{qk})) * Exp(u, q) \tag{7.18}$$

Table 7.10: Question Routing experiments, with users having more than 80 posts

	p@5	p@10	p@20	p@30	r@5	r@10	r@20	r@30	msc@5	msc@10	msc @20	msc@30
TTEA	0.028	0.023	0.019	0.016	0.051	0.083	0.135	0.175	0.132	0.212	0.336	0.424
TTEA-ACT	0.031	**0.026**	**0.020**	**0.018**	0.058	**0.094**	**0.146**	**0.188**	0.150	**0.238**	**0.364**	**0.457**
TEM	0.031	0.026	0.020	0.017	0.056	0.095	0.147	0.188	0.143	0.238	0.356	0.445
TEM-ACT	0.035	**0.027**	**0.021**	**0.018**	0.063	**0.100**	**0.151**	**0.193**	0.165	**0.253**	**0.375**	**0.468**
UQA	**0.040**	0.025	0.016	0.013	**0.077**	0.096	0.124	0.150	**0.194**	0.237	0.299	0.357
Grostot	0.036	0.022	0.015	0.012	0.070	0.086	0.114	0.135	0.177	0.214	0.278	0.325
Random	0.001	0.001	0.001	0.001	0.002	0.003	0.006	0.011	0.005	0.008	0.019	0.030

where θ_{uk}, θ_{qk} is the same than in 7.12 for user topic interest distribution. For our method, we compute $Exp(u, q)$ by equation 7.19

$$Exp(u, q) = \sum_{e=1}^{E} \theta_{kue} * e \tag{7.19}$$

As UQA and GROSTOT do not model expertise, like (Yang 2013b), we set $Exp(u, q)$ to 1 for these two methods. For TEM, we reuse equation 7.20 indicated in (Yang 2013b).

$$Exp(u, q) = \sum_{e=1}^{E} \phi_{z,u,e} * \mu_e \tag{7.20}$$

In order to evaluate different models, we consider the percentage of successful expert recommendation until position N. A successful expert recommendations until position N means that the N-th user, recommended by an algorithm, not only answers the question but also gets the highest votes.

Table 7.11: Expert recommendation experiments

Methods	N=30	N=60	N=100
TEM	0.128	**0.228**	0.392
TTEA	0.079	0.195	**0.443**
UQA	**0.146**	0.206	0.261
Grostt	0.127	0.172	0.220
Random	0.008	0.018	0.028

Table 7.11 shows the results. Random denotes a method where we randomly recommend users for the test questions. We ran the experiments five times and listed the average scores. We summarize our observations as follows: (1) Our TTEA shows the best performances compared with the baseline models when the number of recommended users is large. This means that when we recommend 100 users for each testing questions, in around 44% of cases we have one user not only answering the question, but also winning the highest vote. (2) When the number of recommended users is large, both TEM and TTEA perform better than other models which do not model expertise, so expertise modeling can improve expert recommendation. (3) TEM uses a Gaussian Mixture Model to model ex-

pertise, while we directly model votes which is less precise. Therefore, we perform badly when the number of recommended users is small. (4) After ranking users by topic similarity scores, using expertise scores to re-rank those users actually lowers the probability of the top ranked user to answer the question. The intuition behind is that a user having high expertise on a question does not necessarily have high topic similarity score with the question.

7.4.4 Trends: temporal dynamics at different levels

With the temporal modeling of TTEA, we can explore topic dynamics at many different levels. We present illustrative case studies to show the advantage of temporal modeling.

We first set the time window at the month level. Figure 7.4-a shows the dynamics of $Android$, $Iphone$ and $Flash$ related topics at different months from Jan 2011 to Dec 2011. $Flash$ related topics are more active in the early of 2011, but become less popular in the late of 2011. We then set the time window to the day level. Figure 7.4-b shows the dynamics of $Android$, $Iphone$ and $Flash$ related topics from July 1^{st} 2011 to July 31^{st} 2011. We can see that all topics are active from Monday to Friday, and not active during the weekend. Lastly, we set the time window to the hour level. Figure 7.4-c shows the dynamics of $Android$, $Iphone$ and $Flash$ related topics at different hours during a day. We can verify that both $Android$ and $Iphone$ related topics are more active during daytime, but $Flash$ related topics are more active during the afternoon.

Previous figures show the topic dynamics on a global level. We now illustrate the topic dynamics at the user level. We choose the top active users according to the output of θ_{ku} in the $Android$ related topic and $Iphone$ related topic separately. Figure 7.5-a,b show the activity pattern of the two most active users in the $Iphone$ related topic. We can observe that the user in Figure 7.5-a is only active during work-time. The user seldom answers questions after 7PM. On the contrary, the user in Figure 7.5-b is active until very late but not midnight. Figure 7.5-c,d show the activity pattern of the two most active users in the $Android$ related topic. We can observe that the user in Figure 7.5-c is active in the morning, afternoon and evening. On the contrary, the user in Figure 7.5-d is even active at midnight. For all these users, we can observe that they are not actually active on the topics they are not interested in. We believe this information will benefit many community management related tasks.

7.5 Summary: an effective model to extract expertise and temporal indications

In this chapter, we addressed the problem of topic detection, activity modeling, temporal modeling and expertise detection in Q&A sites. We presented the TTEA (Temporal Topic Expertise Activity) model that simultaneously uncovers the topics, activities, expertise and temporal dynamics. This extracted information enables us to improve tasks such as: question routing, expert recommending and community life-cycle management. We demonstrated that TTEA shows advantages in topic modeling. It also achieves good performances on question routing task and expert detection task compared with the state of the art models. There are still many future directions for this work, for instance, our model is obviously not limited to Q&A datasets and we intend to adapt it to other kinds of social media.

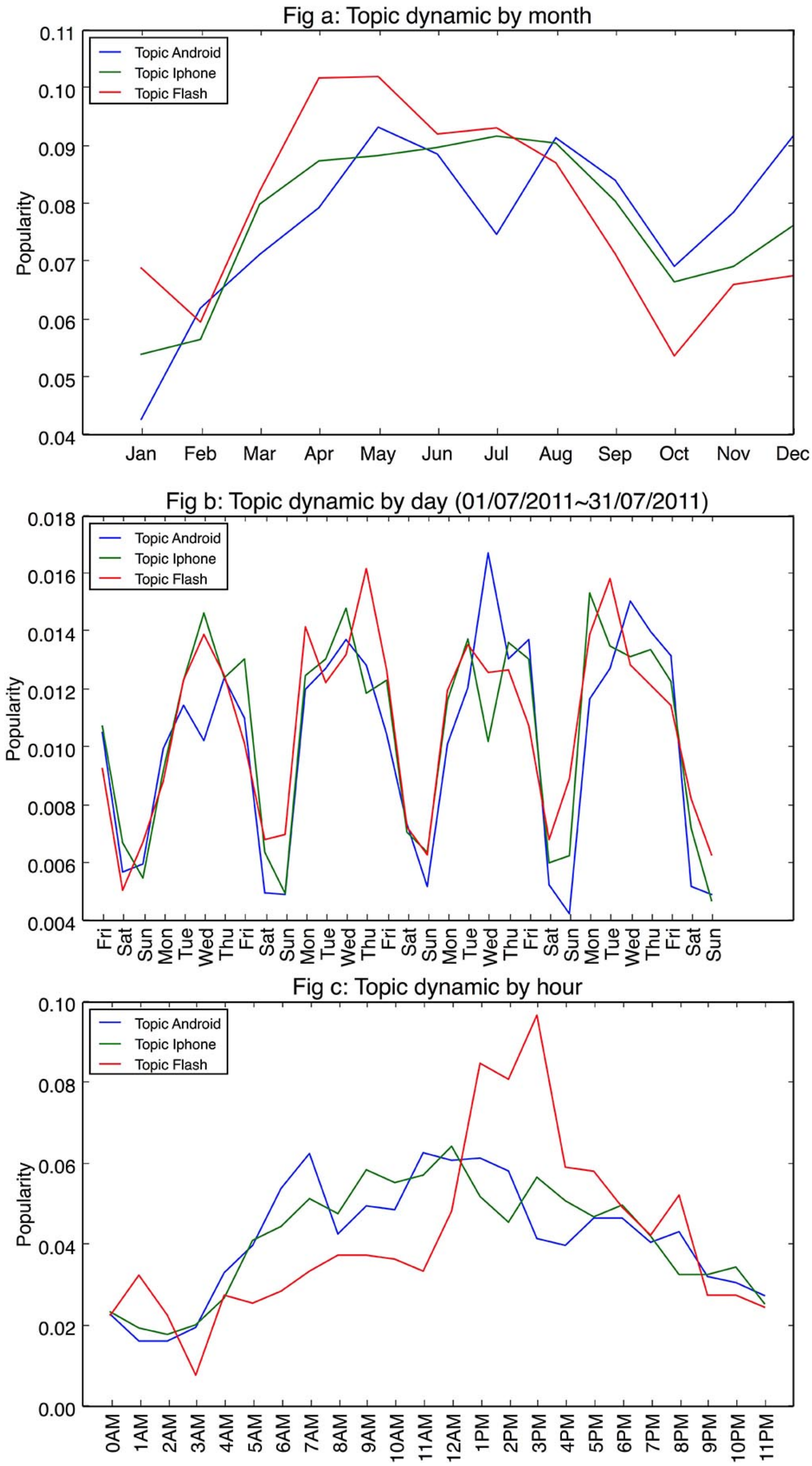


Figure 7.4: Topic dynamics

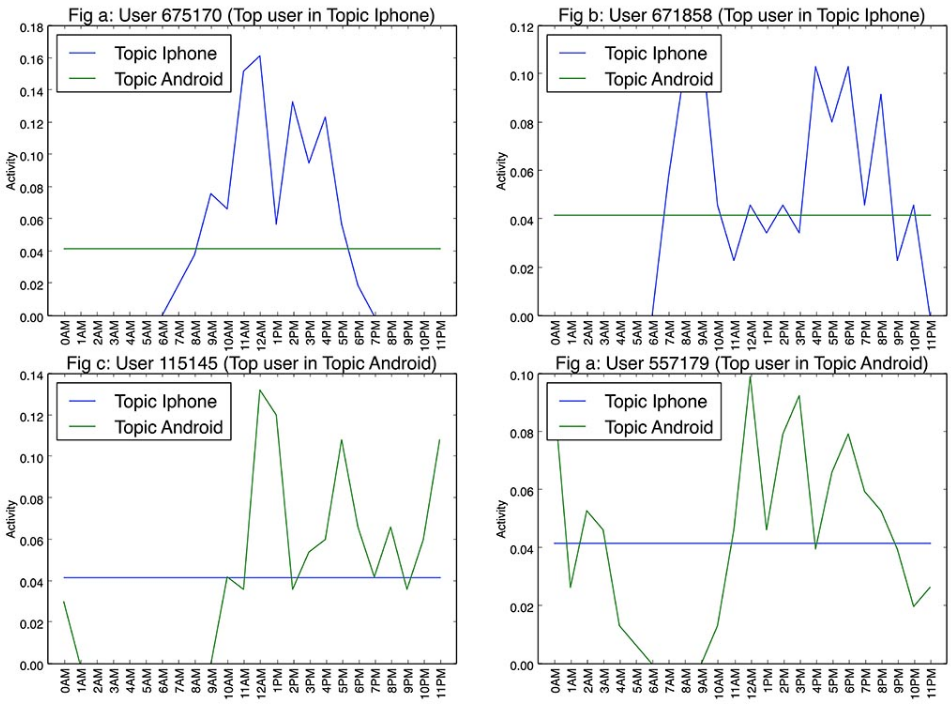


Figure 7.5: User topic activities

CHAPTER 8

Conclusion

Contents

8.1 Summary of contributions

Although the Web always was a social object, the Web 2.0 evolution allowed users to very easily interact and collaborate with each other in a social media platform as creators of user-generated content and members of communities and social networks. When analyzing Social Web activities and productions, it is crucial to jointly consider both aspects: the user-generated contents and the user-generated interactions.

In this thesis, we proposed a framework, which combines social network analysis, social media mining and Semantic Web technologies, to help manage user-generated content websites. The main motivating scenario for our research was the case of question-and-answer sites (Q&A sites), which is a special case of user-generated content (UGC) website and (implicit) communities of interests. Through the archived questions and answers Q&A sites rapidly became huge repositories of potentially reusable knowledge requiring efficient search and access means. They also capture social interactions and structures that can help navigate the knowledge repository by providing interest and expertise indicators.

Therefore we addressed several research questions such as:

- How can we formalize user-generated content? How can we identify the common topics binding users together?
- How can we generate a semantic label for topics? How can we detect topic-based overlapping communities?
- How can we extract topics-based expertise and temporal dynamics?

To answer these research questions, we conducted a study on a data set from the popular question and answer site StackOverflow. First we reused and designed Semantic Web schemas to formalize both the explicit information such as user-generated content and the implicit information such as detected communities, topics and temporal dynamics obtained as a result of our analysis. Then we applied the original LDA model as a first approach to extract this implicit information from the original user-generated content. Based on the results and performances, we extended our work in three directions:

- Firstly, we addressed the efficiency problem of the original LDA model.
- Secondly we automatically generated semantic labels for bag of words which is the output of the original LDA model.
- Thirdly, we proposed a new LDA model supporting the extraction of temporal trends and expertise indicators from user-generated content.

To summarize we consider the major contributions of this thesis are:

- **How can we formalize user-generated content?** We designed a prototype system to formalize both implicit and explicit information in a question and answer site, to extract the implicit information from the original explicit user-generated content, and to provide useful services by using this detected information. In addition, we proposed a vocabulary that can be used to formalize the detected information.
- **How can we identify the common topics binding users together?** We present a topic tree distribution method to extract topics from tags. We also propose a first-tag

enrichment method to enrich questions which only have one or two tags. We show the effectiveness and efficiency of our topic extraction method.

- **How can we generate a semantic label for topics?** We propose and compare metrics and provide a method using DBpedia to generate adequate labels for a bag of words capturing a topic.

- **How can we detect topic-based overlapping communities?** Based on our topic extraction method, we present a method to assign users to different topics in order to detect overlapping communities of interest.

- **How can we extract topics-based expertise and temporal dynamics?** We present a joint model to extract topic-based expertise and temporal dynamics from user-generated content. We also propose a post-processing method to model user activity. Traditionally, this information has been modeled separately.

These results were published in international conferences and journals:

- Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker: Overlapping Community Detection and Temporal Analysis on Q&A Sites. Journal of Web Intelligence and Agent Systems 2016. .

- Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker: Joint model of topics, expertises, activities and trends for question answering Web applications. IEEE/WIC/ACM Web Intelligence 2016.

- Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker, Ge Song: Detecting topics and overlapping communities in question and answer sites. Journal of Social Network Analysis and Mining 5(1): 27:1-27:17 (2015)

- Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker: Simplified detection and labeling of overlapping communities of interest in question-and-answer sites. IEEE/WIC/ACM Web Intelligence 2015

- Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker, Ge Song: Empirical study on overlapping community detection in question and answer sites. IEEE/ACM ASONAM 2014: 344-348

- Zide Meng, Fabien L. Gandon, Catherine Faron-Zucker: QASM: a Q&A Social Media System Based on Social Semantic. International Semantic Web Conference (Posters & Demos) 2014: 333-336

8.2 Perspectives: current limitations and future work

We can group current limitations and perspective according to the research questions we addressed:

- **How can we formalize user-generated content?** We only considered formalizing implicit and explicit information from social media websites, especially question and answer sites. However, people are using different kinds of social media websites as well. We did not conduct research on how to formalize and integrate several social media websites and extract implicit information from the integrated view. For instance a user who showed a interest in economy topics on YouTube may also be interested in the same topic on other platforms. Likewise, a user decreasing his activity on one social media site may indicate a decreasing activity on other social media site (e.g. busy time) or not (e.g. shifting platforms).

- **How can we identify the common topics binding users together?** We designed an efficient method to extract topics from tags on question answer sites. However, some social media sites do not support social tagging on user-generated content. A solution could be to study how to automatically select several keywords or tags for user-generated content and how existing approaches for these questions combine with our analysis.

- **How can we generate a semantic label for topics?** We use DBpedia as external knowledge to help generate labels capturing the meaning of topics. A key step of our method is to link the words of a topic to DBpedia. However, many of these words have no links to the DBpedia knowledge base. One solution could be using more Linked Open Data sources to obtain more links.

- **How can we detect topic-based overlapping communities?** The social network on question answer sites is different to traditional relationship-based social networks. Users are focusing more on the contents rather than links between them. However, for some social media sites, users are interacting and maintaining explicitly social links. In these cases, a perspective would be to combine graph-based overlapping community detection methods with our method.

- **How can we extract topics-based expertise and temporal dynamics?** It is obvious that the proposed models and methods are not limited to the processing of Q&A data sets. We should study how to apply and adapt our model to other kinds of social media websites. In addition, we do not make full use of the extracted user and topic temporal information. A potential direction of work could be combining all the extracted information to optimize question routing and user recommendation tasks and in general provide new functionalities to community managers.

APPENDIX A

Appendix

A.1 Survey Example

A.1.1 Survey Title

Topic Labelling Survey-A

A.1.2 Survey Description

We are studying algorithms to generate a global label for a bag of words representing a topic discussed in a forum. To help us in this study, we invite you to participate to this topic labelling survey. Each question below refers to one bag of words from a real forum topic (e.g. Topic 1 - Bag of words: "css, html, firefox, ie, internet-explorer, browser, xhtml, web-development, div, layout") and several options for a possible label for that topic (e.g. html, firefox,web-development, css, browser") Please choose one option which can represent the best label for that topic. If you find none of the proposed labels is adequate (i.e. if the labels do not well describe the topic in your opinion), please specify your own label using the "Other" label field. Thank you very much for your participation.

A.1.3 Survey Content: An example

Topic 1 - Bag of words: "css, html, firefox, ie, internet-explorer, browser, xhtml, web-development, div, layout" Possible labels:

- html
- firefox

- web-devlopment
- css
- browser
- other

Bibliography

[Adamic 2000] Lada A Adamic and Bernardo A Huberman. *Power-law distribution of the world wide web*. Science, vol. 287, no. 5461, pages 2115–2115, 2000. (Cited on page 73.)

[Ahn 2010] Yong-Yeol Ahn, James P Bagrow and Sune Lehmann. *Link communities reveal multiscale complexity in networks*. Nature, vol. 466, no. 7307, pages 761–764, 2010. (Cited on page 28.)

[Aletras 2014] Nikolaos Aletras and Mark Stevenson. *Labelling Topics using Unsupervised Graph-based Methods.* In ACL (2), pages 631–636, 2014. (Cited on pages 32 and 40.)

[AlSumait 2008] Loulwah AlSumait, Daniel Barbará and Carlotta Domeniconi. *On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking*. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on, pages 3–12. IEEE, 2008. (Cited on page 33.)

[Anderson 2012] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg and Jure Leskovec. *Discovering value from community activity on focused question answering sites: a case study of stack overflow*. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 850–858. ACM, 2012. (Cited on pages 4, 16, 36, 37 and 39.)

[Balasubramaniam 2009] Niroshan Balasubramaniam. *User-generated content*. In Proceedings of business aspects of the internet of things, seminar of advanced topics, pages 28–33. ETH Zurich, 2009. (Cited on page 14.)

[Berners-Lee 2001] Tim Berners-Lee, James Hendler, Ora Lassila*et al. The semantic web*. Scientific american, vol. 284, no. 5, pages 28–37, 2001. (Cited on pages 20 and 21.)

[Berners-Lee 2006] Tim Berners-Lee. *Linked data-design issues*. 2006. (Cited on page 21.)

[Bizer 2011] Christian Bizer. *Evolving the Web into a Global Data Space.* In BNCOD, volume 7051, page 1, 2011. (Cited on page 21.)

[Blei 2003] David M Blei, Andrew Y Ng and Michael I Jordan. *Latent dirichlet allocation*. the Journal of machine Learning research, vol. 3, pages 993–1022, 2003. (Cited on pages 32, 40, 42, 57, 66, 78, 114, 121 and 122.)

[Blei 2006a] David Blei and John Lafferty. *Correlated topic models*. 2006. (Cited on page 32.)

[Blei 2006b] David M Blei and John D Lafferty. *Dynamic topic models*. In Proceedings of the 23rd international conference on Machine learning, pages 113–120. ACM, 2006. (Cited on pages 32 and 33.)

[Blei 2012] David M. Blei. *Probabilistic Topic Models*. Commun. ACM, vol. 55, no. 4, pages 77–84, April 2012. (Cited on page 32.)

[Bouguessa 2008] Mohamed Bouguessa, Benoît Dumoulin and Shengrui Wang. *Identifying Authoritative Actors in Question-answering Forums: The Case of Yahoo! Answers*. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, pages 866–874, New York, NY, USA, 2008. ACM. (Cited on pages 35 and 39.)

[Breslin 2006] John G. Breslin, Stefan Decker, Andreas Harth and Uldis Bojars. *SIOC: an approach to connect web-based communities*. IJWBC, vol. 2, no. 2, pages 133–142, 2006. (Cited on page 2.)

[Breslin 2007] John G. Breslin and Stefan Decker. *The Future of Social Networks on the Internet: The Need for Semantics*. IEEE Internet Computing, vol. 11, no. 6, pages 86–90, 2007. (Cited on page 2.)

[Cano Basave 2014] Amparo Elizabeth Cano Basave, Yulan He and Ruifeng Xu. *Automatic labelling of topic models learned from twitter by summarisation*. Association for Computational Linguistics (ACL), 2014. (Cited on pages 32 and 40.)

[Chang 2009] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish and David M. Blei. *Reading Tea Leaves: How Humans Interpret Topic Models*. In Neural Information Processing Systems, 2009. (Cited on page 126.)

[Chang 2013] Shuo Chang and Aditya Pal. *Routing Questions for Collaborative Answering in Community Question Answering*. In Proceedings of the 2013 IEEE/ACM ASONAM, pages 494–501, New York, NY, USA, 2013. ACM. (Cited on pages 29, 35, 36, 39, 40, 42, 75, 122 and 127.)

[Deerwester 1990] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer and Richard Harshman. *Indexing by latent semantic analysis*. Journal of the American society for information science, vol. 41, no. 6, page 391, 1990. (Cited on page 32.)

[Diao 2012] Qiming Diao, Jing Jiang, Feida Zhu and Ee-Peng Lim. *Finding bursty topics from microblogs*. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pages 536–544. Association for Computational Linguistics, 2012. (Cited on pages 33, 42, 66 and 114.)

[DiNucci 2012] Darcy DiNucci. *âFragmented Futureâ, 1999*. Dostupn é z: http://www. darcyd. com/fragmented future. pdf, 2012. (Cited on page 13.)

[Erétéo 2009] Guillaume Erétéo, Michel Buffa, Fabien Gandon and Olivier Corby. *Analysis of a Real Online Social Network Using Semantic Web Frameworks*. In The Semantic Web - ISWC 2009, 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings, pages 180–195, 2009. (Cited on page 2.)

[Feigenbaum 2007] Lee Feigenbaum, Ivan Herman, Tonya Hongsermeier, Eric Neumann and Susie Stephens. *The semantic web in action*. Scientific American, vol. 297, no. 6, pages 90–97, 2007. (Cited on page 21.)

[Fortunato 2010] Santo Fortunato. *Community detection in graphs*. Physics reports, vol. 486, no. 3, pages 75–174, 2010. (Cited on page 31.)

[Gandon 2013] Fabien Gandon, Michel Buffa, Elena Cabrio, Olivier Corby, Catherine Faron-Zucker, Alain Giboin, Nhan Le Thanh, Isabelle Mirbel, Peter Sander, Andrea G. B. Tettamanzi and Serena Villata. *Challenges in Bridging Social Semantics and Formal Semantics on the Web*. In S. Hammoudi, J. Cordeiro, L.A. Maciaszek and J. Filipe, editeurs, 5h International Conference, ICEIS 2013, volume 190, pages 3–15, Angers, France, July 2013. Springer. (Cited on page 2.)

[Gargi 2011] Ullas Gargi, Wenjun Lu, Vahab S. Mirrokni and Sangho Yoon. *Large-Scale Community Detection on YouTube for Topic Discovery and Exploration*. In ICWSM, 2011. (Cited on pages 29 and 80.)

[Girvan 2002] Michelle Girvan and Mark EJ Newman. *Community structure in social and biological networks*. Proceedings of the national academy of sciences, vol. 99, no. 12, pages 7821–7826, 2002. (Cited on page 41.)

[Griffiths 2004] Thomas L Griffiths and Mark Steyvers. *Finding scientific topics*. Proceedings of the National Academy of Sciences, vol. 101, no. suppl 1, pages 5228–5235, 2004. (Cited on pages 32, 59, 63, 66, 77, 90, 94, 117 and 122.)

[Guo 2008a] Jinwen Guo, Shengliang Xu, Shenghua Bao and Yong Yu. *Tapping on the Potential of Q&a Community by Recommending Answer Providers*. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, pages 921–930, New York, NY, USA, 2008. ACM. (Cited on pages 36 and 39.)

[Guo 2008b] Jinwen Guo, Shengliang Xu, Shenghua Bao and Yong Yu. *Tapping on the potential of q&a community by recommending answer providers*. In Proceedings of the 17th ACM CIKM, pages 921–930. ACM, 2008. (Cited on pages 35, 42 and 121.)

[Harper 2008] F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli and Joseph A. Konstan. *Predictors of Answer Quality in Online Q&A Sites*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, pages 865–874, New York, NY, USA, 2008. ACM. (Cited on page 15.)

[Hofmann 1999a] Thomas Hofmann. *Probabilistic latent semantic analysis*. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, pages 289–296. Morgan Kaufmann Publishers Inc., 1999. (Cited on page 33.)

[Hofmann 1999b] Thomas 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. (Cited on page 32.)

[Hu 2014] Zhiting Hu, Junjie Yao and Bin Cui. *User Group Oriented Temporal Dynamics Exploration*. In Twenty-Eighth AAAI14, 2014. (Cited on pages 34, 40, 41, 42, 66, 114, 117 and 121.)

[Hulpus 2013] Ioana Hulpus, Conor Hayes, Marcel Karnstedt and Derek Greene. *Unsupervised graph-based topic labelling using dbpedia*. In Proceedings of the sixth ACM international conference on Web search and data mining, pages 465–474. ACM, 2013. (Cited on pages 32, 40 and 96.)

[Jeon 2005] Jiwoon Jeon, W Bruce Croft and Joon Ho 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. (Cited on pages 37 and 39.)

[Ji 2013] Zongcheng Ji and Bin Wang. *Learning to rank for question routing in community question answering*. In Proceedings of the 22nd ACM international conference on

Conference on information & knowledge management, pages 2363–2368. ACM, 2013. (Cited on page 35.)

[Jurczyk 2007] Pawel Jurczyk and Eugene Agichtein. *Discovering authorities in question answer communities by using link analysis*. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pages 919–922. ACM, 2007. (Cited on page 34.)

[Lancichinetti 2011] Andrea Lancichinetti, Filippo Radicchi, José J Ramasco and Santo Fortunato. *Finding statistically significant communities in networks*. PloS one, vol. 6, no. 4, page e18961, 2011. (Cited on page 30.)

[Landauer 1997] Thomas K Landauer and Susan T Dumais. *A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.* Psychological review, vol. 104, no. 2, page 211, 1997. (Cited on page 32.)

[Lau 2011] Jey Han Lau, Karl Grieser, David Newman and Timothy Baldwin. *Automatic labelling of topic models*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 1536–1545. Association for Computational Linguistics, 2011. (Cited on page 32.)

[Leskovec 2008] Jure Leskovec, Kevin J Lang, Anirban Dasgupta and Michael W Mahoney. *Statistical properties of community structure in large social and information networks*. In Proceedings of the 17th international conference on World Wide Web, pages 695–704. ACM, 2008. (Cited on page 92.)

[Li 2010a] Baichuan Li and Irwin King. *Routing questions to appropriate answerers in community question answering services*. In Proceedings of the 19th ACM international conference on Information and knowledge management, pages 1585–1588. ACM, 2010. (Cited on pages 5 and 34.)

[Li 2010b] Daifeng Li, Bing He, Ying Ding, Jie Tang, Cassidy Sugimoto, Zheng Qin, Erjia Yan, Juanzi Li and Tianxi Dong. *Community-based Topic Modeling for Social Tagging*. In Proc. of the 19th ACM CIKM, CIKM '10, pages 1565–1568, New York, NY, USA, 2010. ACM. (Cited on page 58.)

[Liu 2009] Ling Liu and M Tamer Zsu. Encyclopedia of database systems. Springer Publishing Company, Incorporated, 2009. (Cited on page 25.)

[Ma 2015] Zongyang Ma, Aixin Sun, Quan Yuan and Gao Cong. *A Tri-Role Topic Model for Domain-Specific Question Answering*. In Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015. (Cited on page 35.)

[McDaid 2010] Aaron McDaid and Neil Hurley. *Detecting highly overlapping communities with model-based overlapping seed expansion*. In Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on, pages 112–119. IEEE, 2010. (Cited on page 30.)

[Mika 2004] Peter Mika. *Social Networks and the Semantic Web*. In 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004), 20-24 September 2004, Beijing, China, pages 285–291, 2004. (Cited on page 2.)

[Mika 2007] Peter Mika. *Ontologies are us: A unified model of social networks and semantics*. Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, no. 1, pages 5–15, 2007. (Cited on page 73.)

[Moens 2014] Marie-Francine Moens, Juanzi Li and Tat-Seng Chua. Mining user generated content. CRC Press, 2014. (Cited on pages 1 and 14.)

[Moro 2014] Andrea Moro, Alessandro Raganato and Roberto Navigli. *Entity Linking meets Word Sense Disambiguation: a Unified Approach*. Transactions of the Association for Computational Linguistics (TACL), vol. 2, pages 231–244, 2014. (Cited on page 99.)

[Ng 2001] Andrew Y. Ng, Michael I. Jordan and Yair Weiss. *On Spectral Clustering: Analysis and an algorithm*. In ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, pages 849–856. MIT Press, 2001. (Cited on page 75.)

[Omitola 2015] Tope Omitola, Sebastián A. Ríos and John G. Breslin. Social semantic web mining. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool Publishers, 2015. (Cited on page 40.)

[O'really 2009] Tim O'really. *Design Patterns and Business Models for the Next Generation of Software*. URL: http://oreilly. com/web2/archive/what-is-web-20. html (01.12. 2013.), 2009. (Cited on pages 13 and 14.)

[Page 1999] Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd. *The PageRank citation ranking: bringing order to the web*. 1999. (Cited on page 103.)

[Pal 2010] Aditya Pal and Joseph A. Konstan. *Expert Identification in Community Question Answering: Exploring Question Selection Bias*. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10, pages 1505–1508, New York, NY, USA, 2010. ACM. (Cited on pages 35 and 39.)

[Pal 2011] Aditya Pal, Rosta Farzan, Joseph A Konstan and Robert E Kraut. *Early detection of potential experts in question answering communities*. In User Modeling, Adaption and Personalization, pages 231–242. Springer, 2011. (Cited on page 36.)

[Papadimitriou 1998] Christos H Papadimitriou, Hisao Tamaki, Prabhakar Raghavan and Santosh Vempala. *Latent semantic indexing: A probabilistic analysis*. In Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pages 159–168. ACM, 1998. (Cited on page 32.)

[Park 2013] Ha-Myung Park and Chin-Wan Chung. *An efficient MapReduce algorithm for counting triangles in a very large graph*. In Proceedings of the 22nd ACM CIKM13, pages 539–548. ACM, 2013. (Cited on page 92.)

[Passant 2009a] Alexandre Passant. *Technologies du Web Sémantique pour lâEntreprise 2.0*. PhD thesis, PhD thesis, Université Paris IV-Sorbonne, 2009. (Cited on page 14.)

[Passant 2009b] Alexandre Passant, Uldis Bojars, John G. Breslin and Stefan Decker. *The SIOC Project: Semantically-Interlinked Online Communities, from Humans to Machines*. In Coordination, Organizations, Institutions and Norms in Agent Systems V, COIN 2009 International Workshops. COIN@AAMAS 2009, Budapest, Hungary, May 2009, COIN@IJCAI 2009, Pasadena, USA, July 2009, COIN@MALLOW 2009, Turin, Italy, September 2009. Revised Selected Papers, pages 179–194, 2009. (Cited on page 40.)

[Plumbaum 2015] Till Plumbaum. *User modeling in the social semantic web*. 2015. (Cited on page 40.)

[Porter 2010] Joshua Porter. Designing for the social web, ebook. Peachpit Press, 2010. (Cited on page 13.)

[Qu 2009] Mingcheng Qu, Guang Qiu, Xiaofei He, Cheng Zhang, Hao Wu, Jiajun Bu and Chun Chen. *Probabilistic Question Recommendation for Question Answering Communities*. In Proceedings of the 18th International Conference on World Wide Web, WWW '09, pages 1229–1230, New York, NY, USA, 2009. ACM. (Cited on pages 38 and 39.)

[Raghavan 2007] Usha Nandini Raghavan, Réka Albert and Soundar Kumara. *Near linear time algorithm to detect community structures in large-scale networks*. Physical Review E, vol. 76, no. 3, page 036106, 2007. (Cited on page 41.)

[Rheingold 2000] Howard Rheingold. The virtual community: Homesteading on the electronic frontier. MIT press, 2000. (Cited on page 12.)

[Sang-Hun 2007] CHOE Sang-Hun. *South Koreans Connect Through Search Engine*. New York Times, 2007. (Cited on page 15.)

[Shah 2010] Chirag Shah and Jefferey Pomerantz. *Evaluating and Predicting Answer Quality in Community QA*. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, pages 411–418, New York, NY, USA, 2010. ACM. (Cited on page 16.)

[Smith 2012] Andrew N Smith, Eileen Fischer and Chen Yongjian. *How does brand-related user-generated content differ across YouTube, Facebook, and Twitter?* Journal of Interactive Marketing, vol. 26, no. 2, pages 102–113, 2012. (Cited on page 14.)

[Sun 2013] Xiaoling Sun and Hongfei Lin. *Topical community detection from mining user tagging behavior and interest.* JASIST, vol. 64, no. 2, pages 321–333, 2013. (Cited on page 30.)

[Sun 2015] Xiangyan Sun, Yanghua Xiao, Haixun Wang and Wei Wang. *On conceptual labeling of a bag of words*. In Proceedings of the 24th International Conference on Artificial Intelligence, pages 1326–1332. AAAI Press, 2015. (Cited on pages 32, 40 and 96.)

[Tang 2008] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang and Zhong Su. *Arnet-Miner: extraction and mining of academic social networks*. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 990–998. ACM, 2008. (Cited on page 30.)

[Thrun 2004] Sebastian Thrun, Lawrence K. Saul and Bernhard Schölkopf, editeurs. Advances in neural information processing systems 16 [neural information processing systems, NIPS 2003, december 8-13, 2003, vancouver and whistler, british columbia, canada]. MIT Press, 2004. (Cited on page 32.)

[Wang 2006] Xuerui Wang and Andrew McCallum. *Topics over time: a non-Markov continuous-time model of topical trends*. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 424–433. ACM, 2006. (Cited on pages 33, 66 and 114.)

[Wang 2007] Xuanhui Wang, ChengXiang Zhai, Xiao Hu and Richard Sproat. *Mining correlated bursty topic patterns from coordinated text streams*. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 784–793. ACM, 2007. (Cited on page 33.)

[Watson 2008] Tom Watson. Causewired: plugging in, getting involved, changing the world. John Wiley & Sons, 2008. (Cited on page 13.)

[Web 2007] Participative Web and User-Created Content. *Web 2.0, Wikis and Social Networking*. SourceOCDE Science et technologies de l'information, vol. 15, 2007. (Cited on page 14.)

[Wei 2006] Xing Wei and W. Bruce Croft. *LDA-based Document Models for Ad-hoc Retrieval*. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM. (Cited on pages 63 and 119.)

[Wu 2008] Hu Wu, Yongji Wang and Xiang Cheng. *Incremental Probabilistic Latent Semantic Analysis for Automatic Question Recommendation*. In Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08, pages 99–106, New York, NY, USA, 2008. ACM. (Cited on pages 38 and 39.)

[Xie 2013] Jierui Xie, Stephen Kelley and Boleslaw K. Szymanski. *Overlapping community detection in networks: The state-of-the-art and comparative study*. ACM Comput. Surv., vol. 45, no. 4, page 43, 2013. (Cited on pages 28, 29, 31, 41 and 82.)

[Xu 2012] Zhiqiang Xu, Yiping Ke, Yi Wang, Hong Cheng and James Cheng. *A model-based approach to attributed graph clustering*. In SIGMOD Conference, pages 505–516, 2012. (Cited on page 29.)

[Yang 2013a] Jaewon Yang, Julian McAuley and Jure Leskovec. *Community detection in networks with node attributes*. In Data Mining (ICDM), 2013 IEEE 13th International Conference on, pages 1151–1156. IEEE, 2013. (Cited on pages 30, 31, 41 and 78.)

[Yang 2013b] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun and Zhong Chen. *CQArank: jointly model topics and expertise in community question answering*. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pages 99–108. ACM, 2013. (Cited on pages 35, 36, 39, 40, 42, 73, 82, 114, 121, 122, 127, 130 and 137.)

[Yao 2010] Junjie Yao, Bin Cui, Yuxin Huang and Yanhong Zhou. *Detecting bursty events in collaborative tagging systems*. In Data Engineering (ICDE), 2010 IEEE 26th International Conference on, pages 780–783. IEEE, 2010. (Cited on page 33.)

[Yao 2012] Junjie Yao, Bin Cui, Yuxin Huang and Yanhong Zhou. *Bursty event detection from collaborative tags*. World Wide Web, vol. 15, no. 2, pages 171–195, 2012. (Cited on page 33.)

[Yin 2013] Hongzhi Yin, Bin Cui, Hua Lu, Yuxin Huang and Junjie Yao. *A unified model for stable and temporal topic detection from social media data*. In Data Engineering (ICDE), 2013 IEEE 29th International Conference on, pages 661–672. IEEE, 2013. (Cited on page 33.)

[Zhang 2007a] Haizheng Zhang, Baojun Qiu, C. Lee Giles, Henry C. Foley and John Yen. *An LDA-based Community Structure Discovery Approach for Large-Scale Social Networks*. In ISI, pages 200–207, 2007. (Cited on page 30.)

[Zhang 2007b] Jun Zhang, Mark S Ackerman and Lada Adamic. *Expertise networks in online communities: structure and algorithms*. In Proceedings of the 16th international conference on World Wide Web, pages 221–230. ACM, 2007. (Cited on pages 34 and 39.)

[Zhao 2011] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan and Xiaoming Li. *Comparing twitter and traditional media using topic models*. In Advances in Information Retrieval, pages 338–349. Springer, 2011. (Cited on pages 66 and 114.)

[Zhou 2012a] Guangyou Zhou, Siwei Lai, Kang Liu and Jun Zhao. *Topic-sensitive Probabilistic Model for Expert Finding in Question Answer Communities*. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, pages 1662–1666, New York, NY, USA, 2012. ACM. (Cited on pages 35 and 39.)

[Zhou 2012b] 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, WWW '12 Companion, pages 783–790, New York, NY, USA, 2012. ACM. (Cited on page 5.)