Harnessing online social media to deal with information overload.

Li, Chenliang.

2013

Li, C. (2013). Harnessing online social media to deal with information overload. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/54827>

<https://doi.org/10.32657/10356/54827>

Harnessing Online Social Media to Deal with Information Overload

A thesis submitted
for a degree of Doctor of Philosophy
by

Li Chenliang

School of Computer Engineering
Nanyang Technological University

2013

Abstract

In online social media, users become information creators and disseminators through the active interplay between information items and other users, instead of just being information consumers of a decade ago. This kind of information production and dissemination in collaborative and active manner further aggravates the problem of information overload on the World Wide Web (WWW). The existing approaches for information retrieval (IR) and natural language processing (NLP) tasks often offer an intolerable response time for Web users. Moreover, given the numerous interactions between users and information items, new kinds of information needs are emerging, such as opinion mining, event detection and summarization, etc. However, the existing IR technologies (based on bag-of-word model), and NLP technologies (based on the linguistical features), often fail to satisfy the web users in these emerging information needs. On the other hand, people participate in online social media to share stories, photos with their friends, vote and leave opinions, or tag web pages, and so on. The digital footprints of these behaviors make online social media semantic resources which we can exploit to better understand and organize the astronomical information.

In this dissertation, we first analyze online social media as multi-dimensional social network by taking Wikipedia as a case study. We find that given the multiple relations exposed from different perspectives in the network, focusing on only one specific relation could lead to biased or even wrong conclusion. Traditional information retrieval approaches are mainly bag-of-word model and keyword based, which ignore the word

ordering in the text and measure the relevance based on the presence of the keywords. We propose a generalized framework for word sense disambiguation based on Wikipedia. The proposed framework can enable effective and efficient disambiguation by relating keyphrases (*i.e.*, n-grams) in the documents to their appropriate concepts in Wikipedia, where a concept is defined as a Wikipedia article. The framework is applicable to the documents of different languages with different settings. By adopting the disambiguation method, we could represent a textual document by the concepts it covers based on Wikipedia. We study the semantic tag recommendation task for web pages based on the concept model by exploring the semantic relations between tags and concepts underlying human annotation activities. Web users participate in the information generation process by commenting news articles, sharing stories and publishing opinions by posting microblogs, etc. However, the information generated by users are often short and written with free style, containing grammatical errors, informal abbreviations (*e.g.*, comments, tweets). These adverse features deteriorate the performance of the existing algorithms for many tasks for online social media, such as named entity recognition, event detection, etc. We propose an unsupervised approach for named entity recognition in targeted Twitter stream. Within this work, we develop an algorithm of tweet segmentation which splits each tweet into non-overlapping phrases, called tweet segments. Inspired by the semantic units produced by tweet segmentation, we further propose an algorithm for event detection for tweets based on tweet segments, which is effective and scalable.

Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisors Dr. Anwitaman Datta and Dr. Aixin Sun, who gave invaluable guidance, encouragement, and wisdom over the past four years. Without their consistent effort and unending patience, this thesis could not have reached its present form.

I'd like to thank Dr. Adam Wierzbicki for hosting me during my visit to PJWSTK, as well as for his advice and suggestion. I would also thank Dr. Jianshu Weng and Dr. Yuxia Yao for being my supervisors during my internship at HP Labs Singapore. During the internship, I also worked closely with Dr. Qi He. The experience of working with them was fascinating and rewarding. I am also grateful to Dr. Lee Bu Sung, Francis, for his support of my internship at HP Labs Singapore.

I also wish to extend my appreciation to the support staffs at the Center for Advanced Information Systems (Loo Kian Hock and Ang Ah Giat) and Parallel Distributed Computing Center (Ng-Goh Siew Lai) for their technical support. Moreover, I also like to thank all my fellow friends at the Center for Advanced Information Systems for their support: Li Bin, Chen Ning, Liu Yong, Hai Zhen, Lin Wenqing, Yuan Quan, Lin Miao, Ma Zongyang, Xu Zhiqiang and Zhu Linhong.

Finally, I must express my love to my beloved family for their encouragement and support. Without them, this work would not have been possible.

Thank you all.

Contents

Abstract	i
Acknowledgments	iii
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Motivation	1
1.2 Research Overview	3
1.2.1 Multi-dimensional Social Network	3
1.2.2 Concept Representation and Its Application	4
1.2.3 Named Entity Recognition from Tweet Streams	7
1.2.4 Event Detection from Tweets	8
1.3 Research Contributions	8
1.4 Dissertation Outline	10
2 Literature Survey	11
2.1 Knowledge Discovery and Data Mining based on Wikipedia	12
2.1.1 Dynamics of Knowledge Building in Wikipedia	12
2.1.2 Exploring Semantic Information in Wikipedia	15
2.2 Word Sense Disambiguation Using Wikipedia	19
2.2.1 Knowledge-based Methods	19
2.2.2 Supervised Machine Learning Methods	21
2.3 Social Tagging System	23
2.4 Named Entity Recognition for Tweets	25
2.5 Event Detection from Tweet Stream	27
2.5.1 Event Detection from Formal Texts	28
2.5.2 Event Detection from Tweets	29

3	Mining Latent Relations: A Case Study with Wikipedia Article Similarity and Controversy	31
3.1	Article Similarity in Wikipedia	34
3.1.1	Expert-based Similarity	34
3.1.2	Article Relevance Aspects	37
3.2	Article Similarity Evaluation	39
3.2.1	Dataset	40
3.2.2	Evaluation with Compactness	40
3.2.3	Evaluation with Partial Ground-truth	45
3.2.4	Evaluation with Linear Correlation	48
3.2.5	Qualitative Comparison	49
3.3	Source of Controversy	50
3.3.1	Methodology	51
3.3.2	Controversial Articles	53
3.3.3	Role of Contributors	55
3.3.4	Concept Perspective	58
3.4	Conclusions	59
4	Word Sense Disambiguation Using Wikipedia	63
4.1	Introduction	63
4.2	<i>TSDW</i> Disambiguation Framework	67
4.2.1	Wikipedia Inventory	68
4.2.2	Keyphrase Recognizer	69
4.2.3	Two-Stage Disambiguator	70
4.3	Experiments	77
4.3.1	<i>TSDW</i> Setup and Performance Metric	78
4.3.2	Comparison with Other Methods	79
4.3.3	Comparison with Illinois Wikifier	84
4.3.4	Evaluation of <i>TSDW</i>	87
4.4	Summary	96
5	Tag Recommendation based on Concept Model	98
5.1	Introduction	98
5.2	Concept model	99
5.3	Experiments	102
5.4	Summary	105

6	Named Entity Recognition in Targeted Twitter Stream	107
6.1	Introduction	108
6.1.1	Tweet Segmentation	110
6.1.2	Segment Ranking	112
6.1.3	Contributions	114
6.2	Tweet Segmentation	114
6.2.1	A Dynamic Programming Algorithm	115
6.2.2	Segment Stickiness Function	116
6.2.3	Enhanced Stickiness by World Wide Web	118
6.2.4	Length Normalization	120
6.3	Segment Ranking	120
6.3.1	Noise Filtering	120
6.3.2	Local Segment Graph	121
6.3.3	Random Walk Model	122
6.4	Evaluation	123
6.4.1	Tweets Data and Performance Metrics	124
6.4.2	Comparison with Other Methods	126
6.4.3	Analysis of TwiNER	128
6.5	Summary	134
7	Segment-based Event Detection from Tweets	136
7.1	Introduction	137
7.1.1	Overview of Twevent	139
7.2	Twevent	141
7.2.1	Tweet Segmentation	141
7.2.2	Event Segment Detection	142
7.2.3	Event Segment Clustering	145
7.2.4	Discussion	150
7.3	Experiments	151
7.3.1	Dataset and Experimental Setting	151
7.3.2	Statistics on Tweet Segmentation	153
7.3.3	Event Detection Results	154
7.3.4	Impact of τ in Twevent	158
7.3.5	Impact of K and k in Twevent	159
7.4	Summary	160
8	Conclusion and Future Work	163
8.1	Summary	163
8.2	Future Work	167

A Appendix	170
A.1 Keyphrase Recognition	170
A.2 List of Publications	173
References	174

List of Figures

3.1	Explicit relations among articles and contributors in Wikipedia	35
3.2	Compactness vs. Number of Trials for expert-based similarity, cosine similarity, P-Rank and SimRank	42
3.3	Two example clusters	44
3.4	Distribution of CAs using expert-based similarity	57
3.5	Distribution of CAs using cosine similarity	62
4.1	A sample article with phrases disambiguated to Wikipedia topics	65
4.2	The two-stage disambiguation process	72
4.3	Accuracy of varying M_1 and c with Dice, Jaccard and WLM for two datasets.	88
4.4	Boxplot for the number of unambiguous keyphrases per article that with <i>keyphraseness</i> value above a threshold	91
4.5	Accuracy (%) at varying θ on the two training sets	92
4.6	Accuracy (%) of varying M_2 on the two sets of articles	94
6.1	System Architecture of <i>Twiner</i>	109
6.2	Example of Tweet Segmentation	111
6.3	Different Segmentations of a Portion of Tweet	119
6.4	Precision of top K named entities	133
7.1	Segment-based Event Detection System Architecture	141
7.2	Statistics on tweets and tweet segments	153
7.3	Comparison of frequency distributions for the two events: Twilight and Park Yong-ha	156
7.4	<i>Twevent</i> and <i>Twevent_u</i> against different τ values	156
A.1	Example of Keyphrase Tree of initial word <i>java</i>	171

List of Tables

3.1	Symbols and semantics	36
3.2	Similarity aspects and metrics	39
3.3	Compactness for similarity measures w.r.t percentages of the noise	43
3.4	K-Medoids clustering results	46
3.5	Agglomerative clustering results	47
3.6	Correlation within five segments of cosine similarity	49
3.7	Dispute Tags Used	55
4.1	Statistics on English and Traditional Chinese datasets	80
4.2	Disambiguation accuracy and execution time on English and Traditional Chinese evaluation sets	81
4.3	Number of ambiguous keyphrases processed by the second stage disambiguation with different settings for <i>TSDW</i> on English and Traditional Chinese evaluation sets	83
4.4	Statistics on the four datasets	85
4.5	Disambiguation accuracy (%) and execution time (second) of <i>TSDW</i> and <i>Illionis Wikifier</i> on four evaluation sets: $M = 10$ for <i>TSDW</i>	86
4.6	Relatedness distribution using <i>Dice</i> , <i>Jaccard</i> and <i>WLM</i>	90
4.7	Disambiguation accuracy and execution time by applying varying <i>keyphraseness</i> thresholds on English training set.	91
4.8	Accuracies (%) and the number of the high-confident disambiguation decisions (#HiCfd) at different stages.	96
4.9	The proportions of errors that are of low, medium and high confidence. .	96
5.1	Statistics on datasets	103
5.2	Statistics on execution time (second).	105
5.3	Performance of methods, where * indicates the difference against the best performance is significant with $p < 0.01$ (paired <i>t</i> -test)	106
6.1	Example Named Entities in Tweet	109
6.2	Different NER systems' performance on tweets	127

6.3	Conventional NER systems' performance on tweets	129
6.4	Impacts of Tweet Segmentation by SCP	131
6.5	Impacts of <i>local context</i> , <i>global context</i> and <i>random walk</i> on segment ranking	133
7.1	Detection results of <i>Twevent</i> , <i>Twevent_u</i> , and <i>EDCoW</i>	154
7.2	Events detected by <i>EDCoW</i> in June 07 – June 12, 2010	155
7.3	List of the 22 events detected by <i>Twevent</i> during the period of June 07 to June 12, 2010.	161
7.4	Events detected by <i>Twevent_u</i> in June 07 – June 12, 2010	162

Chapter 1

Introduction

Without deviation from the norm, progress is not possible.

– Frank Zappa

1.1 Motivation

As Internet in general has played an increasingly important role in our daily life, the rapidly growing amount of information available on the world wide web (WWW) has grown far beyond the scope that people can handle manually. The situation is further aggravated by the success of a number of online social media enabled by Web 2.0 techniques. Compared to traditional websites that restrict the users to be only information consumers, online social media enables the users to produce information via different interactions between users and information items, such as Wikipeida [10] and Open Directory Project (ODP) [7] for collaborative knowledge building; Del.icio.us [3], BibSonomy [1] and CiteULike [2] for collaboratively tagging documents, Digg [4] for evaluating web content, Facebook [5], Twitter [8] and Weibo [9] for information sharing and commenting among friends; Netflix [6] for evaluating movies, YouTube [12] for sharing videos, Yahoo! Answers [11] for knowledge sharing, etc. On one hand, user-generated content with the exponential growth has been produced by online social media. People begin to feel frustrated by the sense that it takes longer to find something relevant to what

they need. Moreover, most of the information generated by web users are short, and written in free styles. These content often contain informal abbreviations, grammatical errors as well as misspellings. This largely deteriorates the performance of the existing state-of-art approaches for related tasks such as document clustering and classification, content understanding and summarization, named entity recognition, etc. Meanwhile, since web users have become information creators and disseminators, new kinds of information needs are also emerging, such as opinion mining, expert finding, event detection and summarization, etc. Existing/traditional solutions for the involved IR and NLP technologies, often fail in such tasks from the perspectives of effectiveness and efficiency [47, 52, 88, 53, 62]. On the other hand, the semantic resource encoded in the user-generated content through the intensive interactions could be exploited to develop useful mechanisms to enhance the performance of existing IR/NLP techniques. Moreover, it provides with a avenue to understand and monitor the users' opinion and identify experts or most influent users. For example, web users unconsciously manifest their expertise and interest by editing Wikipedia articles [84], posting related tweets [152, 135, 141]; Folksonomy is built collaboratively by creating and managing tags to annotate and categorize content [48, 51, 57, 89]; the emerging events are reported shortly by the public via microblogging services, or social tagging [124, 161]. In this thesis, several interesting ideas are investigated:

- As people participate actively in online social media, a multi-dimensional social network over online social media is deduced. There are implicit relations that emerge from the different perspectives. Mining such latent relations, or wisdom of crowds, could help us better understand the dynamics of online social media. Moreover, it could further help us develop better IR/NLP applications that explore the user-generated content in online social media.

- Traditional IR techniques are based on bag-of-word (BOW) models, where the order of the words is ignored and considered interchangeable. The relevance is measured based on only the occurrence of each single word. However, the design of BOW model can not capture the polysemy and synonym that exist naturally in human languages. Also, the high dimension of BOW model makes the BOW based techniques unable to process the information efficiently. The metadatas (*e.g.*, Wikipedia articles, tag annotations) produced collaboratively by web users in the online social media could assemble a semantic space. Furthermore, these metadata could provide with different semantic relations among themselves. Hence, we can integrate it with the existing IR techniques.
- With few restrictions in online social media, the user-generated content are often written with free styles. For example, tweets contains informal abbreviations, grammatical errors, and reliable capitalizations, etc. Similar situation happens in the users' comments, and communications. While the user-generated content is a fruitful resource for understanding the trends, opinions talked by web users, this kind of error-prone nature makes the existing NLP techniques perform poorly. We like to investigate the mechanisms to heal the adverse impact of the error-prone nature of user-generated content, and develop useful tools to help users identify the needles from a huge pile of haystacks efficiently.

1.2 Research Overview

1.2.1 Multi-dimensional Social Network

Online social media enables users more interactions with information items as well as with other users. These interactions induce a multi-dimensional social network over online social media. At first, we conduct a case study based on Wikipedia to illustrate

that knowledge from different perspectives can be obtained by mining various explicit and implicit relations encoded in different dimensions or their combinations. Well, the discovered knowledge by mining one dimension, or declared explicit relation, can offer us only a limited facet of the network rather than enabling a thorough understanding of the social context within the network. Moreover, we show that combining information from different perspectives can offer us more holistic knowledge that may not be discovered by considering the various aspects in isolation. The work demonstrates the possibility of using different aspects of relations to determine new information, which, without the use of a specific relation aspect may simply not be desirable or misleading. It has significant implications for us to design algorithms to harness the wisdom of crowds in online social media for discrete tasks of IR, like semantic tag recommendation (Chapter 5).

1.2.2 Concept Representation and Its Application

Traditional information retrieval technologies mainly focus on BOW models and keyword based search. Documents are naively assumed to be relevant only if they contain the specified keywords. However, words are the smallest unit of language with least semantic information. Moreover, a single word may be ambiguous without adequate specific contextual information. The situation is further aggravated by the short, noise-prone nature of user-generated content. The first step in this dissertation for enhancing information retrieval is to represent information items with metadatas of more semantic information, instead of BOW model. Furthermore, it is desirable that we could derive semantic relations between these dimensions, which is difficult to address without complicated algorithms for BOW model (*e.g.*, co-cluster documents and words for document clustering [32, 131], derive associations among words (*i.e.*, topic model) by using probabilistic latent semantic analysis (PLSA) [59], latent dirichlet allocation (LDA) [20] and their variations [37, 150, 148, 157, 136]). The inception of Wikipedia, the largest online

encyclopedia, could us with such a concept space from which we would derive a concept representation for each information item. In Wikipedia, each article describes a single topic/concept and is titled using the name which is most commonly used and recognized regarding the topic. As a unique feature, wikilinks in Wikipedia links the relevant topics of the article and technical terms mentioned to the corresponding articles which are likely to enhance a reader’s understanding. Thus, the semantic link structure can be exploited to measure the similarity between topics with less computation cost. Moreover, the semantic resources like synonym, abbreviations, associative relations make Wikipedia a promising choice for such task.

The transformation of a document into concept space can be easily conducted by scanning the content of the document and identify the keyphrases¹ that can be referred to some concept in Wikipedia. As being a characteristic of many human languages, a word or term may carry more than one meaning in the language. That is, in our context, a keyphrase may be used to refer to more than one concept. Often, the context of each ambiguous keyphrase is helpful for us to identify its correct meanings. There are many existing works proposed to address the problem of ambiguous keyphrases in the context of Wikipedia. However, neither of them have taken the efficiency as a performance issue in their study. Given the increasingly rapid growth of Web information, reducing the cost of computation becomes more and more important in order to give a quick response to the web users. Based on this point, we propose a generalized framework exploring the use of Wikipedia as the lexical resource for word sense disambiguation. The method identifies the correct meanings of ambiguous keyphrases by approximating the likelihood of each candidate meaning based on the semantic context. Unlike the existing works that take all the unambiguous keyphrases in a document as the context, we construct the context by taking some representative unambiguous keyphrases based on a

¹In this thesis, we define keyphrase as a term of n-grams that can be mapped to some concept of Wikipedia

keyphraseness measure. We believe that the representative unambiguous keyphrases have adequate semantic information for disambiguation. Moreover, a two-stage disambiguator is developed to tackle the data sparsity problem rooted in the existing works by exploiting the semantic information provided in the ambiguous keyphrases. While the existing works were investigated based on some specific similarity measure, the proposed framework can generalize well to different context settings, which is applicable to different application needs. The experimental results show that the proposed method achieves comparable performance with much less computation cost in significance, which is specially desired for real-time applications such as document topic indexing [97].

Afterwards, we investigate the possibility that explores the latent semantic relations encoded in the social tagging process by leveraging the concept space of Wikipedia we discussed above. In social tagging, users assign tags to information items, like web pages, photos, etc., based on their content information as well as the user's personal purpose. Studies [51, 134] show that users often reuse existing tags or contribute new tags themselves. In such a manner, the semantically relevant tags which are considered to be high quality emerge as the prominent tags with top frequency. These are the tags that are often used to represent specific topics, which we call *semantic tags*. Existing work showed us that semantic tags are normally common nouns and proper names, belonging mainly to the category of topic which describes what a tagged resource is about. Thus, it is natural to infer that people subconsciously consider the concepts covered by the document when they are tagging a document. The above discussed concept extraction method based on Wikipedia provides us with a brand new approach to investigate the semantic relations hidden in the tagging process. The semantic tags have been proven to be useful for document clustering and classification [117]. Thus, harnessing the wisdom of crowds encoded in the social tagging systems can not only enhance the users' tagging experience by recommending relevant tags, but also support many other IR tasks that

are still under-investigated. For example, higher recall is obtained when the user’s query is expanded by related tags through the relevance feedback. We adopt a probabilistic framework to model a web document by its contained concepts and the likelihood of the semantic tags representing these concepts for tag recommendation. The study shows that by combining the latent semantic relations extracted from collaborative tagging process and the concept space derived from Wikipedia, we can recommend semantic tags with comparable accuracy to other state-of-the-art approaches but with an order of magnitude reduction in computation cost.

1.2.3 Named Entity Recognition from Tweet Streams

The short and noisy documents dominate the online social media (*e.g.*, comments, tweets). These user-generated content are often studied for event detection, as well as to collect and understand users’ opinions. For example, many private and/or public organizations have been reported to create and monitor the targeted twitter stream to collect and understand users’ opinions about the organizations. Targeted twitter stream is usually constructed by filtering tweets with user-defined selection criteria (*e.g.*, tweets published by users from a selected region, or tweets that match one or more predefined keywords.). There is an emerging need for early crisis detection and response with such targeted stream. Such applications require a good named entity recognition system for Twitter, which is able to automatically discover emerging named entities that are potentially linked to the crisis. However, the adverse features, like short length and informal writing style, deteriorate the performance of the existing named entity recognition (NER) approaches. We investigate the task of named entity recognition for tweets and propose an unsupervised NER approach for targeted twitter stream. The proposed approach first segments each tweet into non-overlapping semantic information units (*i.e.*, phrases), each of which may represent a named entity. Then the local context based on co-occurrence

pattern of the named entities are explored together with the external knowledge base, Wikipedia, to derive the true named entities from a pile of the tweet segments.

1.2.4 Event Detection from Tweets

Related to NER task in Twitter stream, event detection from tweets is an important task to detect and understand the current events/topics and the related users' opinions. We propose a segment-based system for event detection and summarization by utilizing the tweet segmentation employed in our work of NER for targeted Twitter stream. We explore the wisdom of crowds (*i.e.*, user support) to derive the event-related segments during the process. The newsworthy events are then summarized with the most newsworthy event-related segments (*i.e.*, named entities or semantic phrases) based on the semantic knowledge contained in Wikipedia. It is shown that the proposed system can detect more realistic events with better summarization, and require moderate computation cost.

1.3 Research Contributions

This thesis focuses on the development and enhancement of IR and NLP techniques for the challenges amplified by the emergence of online social media. Hence, we make several contributions as follow.

- We investigate the latent relations encoded within the multi-dimensions of online social media, and the hidden knowledge discovered specifically from each perspective. A case study is conducted to illustrate that knowledge from different perspectives can be obtained by investigating the various relations among Wikipedia articles. Then, we discern the origin of some controversies in Wikipedia by combining knowledge obtained from such different perspectives. It is shown that some knowledge may not be deduced by considering any single particular aspect in isolation.

- The semantic knowledge of Wikipedia makes it a fruitful resource for many tasks, such as text clustering, named entity disambiguation, and so on. Among these applications, the task of identifying the word sense based on Wikipedia is a crucial component. It is also an critical component to link words or phrases in text documents to their corresponding Wikipedia topic. Accordingly, we propose a generic two-stage word sense disambiguation based on Wikipedia (called *TSDW*), which is independent of language and context settings. The data sparsity problem potentially rooted in existing works is tackled by exploring the semantic information provided by both ambiguous and unambiguous keyphrases.
- Web users organize documents in social tagging systems by assigning tags which reflect the content of the document or the personal needs. The common tags given by multiple users to a particular document, however, are often semantically relevant to the content of the document and each tag represents a specific aspect (*i.e.*, topic). We propose to address the task of semantic tag recommendation by considering the concepts contained in documents. Specifically, we represent each document using a few most relevant concepts contained in the document, where the concept space is derived from Wikipedia. To the best of our knowledge, this is the first work that explores the semantic relation between the Wikipedia concepts and associated tags annotated to the documents of social tagging systems.
- In order to address the performance degradation of the existing named entity recognition approaches in Twitter streams, we develop an unsupervised approach for named entity recognition in targeted twitter streams (called *TwiNER*). We combine both the semantic knowledge of WWW and local co-occurrence patterns together to identify the named entities in targeted Twitter streams.

- We develop an algorithm to address the task of event detection for twitter streams, called *Twevent*. In order to deal with the adverse features, like short length, noise and dynamic topics, and large data volume, we propose to utilize the tweet segments (*i.e.*, semantic information unit) and the external knowledge base (Wikipedia) to tackle these challenges. The approach is shown to achieve better performance in terms of accuracy and efficiency.

1.4 Dissertation Outline

The thesis is organized as follows: In Chapter 2, a review of related work about the information retrieval and natural language processing based on online social media is given. In Chapter 3, we take Wikipedia as a case study to examine the possibility of combining knowledge from different perspectives to better understand the dynamics of online social media. Chapter 4 proposes a generic framework for word sense disambiguation based on Wikipedia, which aims to associate a phrase in a document to its semantic concept defined in Wikipedia. Then we investigate the task of recommending semantic tags for web pages by exploring the semantic relations between tags and concepts derived from Wikipedia in Chapter 5. In Chapter 6, we propose an unsupervised method for named entity recognition for targeted twitter streams, which explores the global knowledge from WWW and local knowledge derived from the targeted twitter stream. Chapter 7 proposes an event detection algorithm for twitter streams. Finally, Chapter 8 concludes the thesis and identifies potential research directions for future work.

Chapter 2

Literature Survey

The artist is nothing without the gift, but the gift is nothing without work.

– Emile Zola

In this chapter, we present literature survey of the existing works about the online social media analysis and mining, and the related IR/NLP tasks. Because of its massive scale of collaboration as well as usage, and open access to all edit-related history, Wikipedia analysis has become a research subtopic in its own right in recent years. Among various aspects of Wikipedia being studied, the works focusing on the coordination and conflict between users, the quality of Wikipedia articles, and the works about enhancing information retrieval by leveraging the semantic resources offered by Wikipedia are reviewed in Section 2.1. In Section 2.2, we discuss the related works about the task of word sense disambiguation based on Wikipedia and corresponding state-of-the-art methods. In Section 2.3, we survey the related works on social tagging systems and the related works about tag recommendation. Section 2.4 reviews the works regarding the named entity recognition for tweets. Section 2.5 discusses the works about event detection for formal text stream, as well as for tweet streams. In the following literature review, we also compare them with ours.

2.1 Knowledge Discovery and Data Mining based on Wikipedia

2.1.1 Dynamics of Knowledge Building in Wikipedia

In Wikipedia, people collaboratively contribute their knowledge with few restrictions. Consequently, newsworthy events and emerging topics are often added within hours or a few days after their occurrence, and inappropriate statements, spam and vandalism are corrected quickly by the wisdom of crowds of large scale. The quality of its articles are well considered comparable with the traditional Encyclopedia Britannica [46]. Due to its success, many researchers tried to investigate the characteristics that lead to the effective collaboration of this large scale. Since Wikipedia is the largest online collaborative knowledge building system, the characteristics of the collaboration activities, like coordination and conflict, conducted by people with different backgrounds have become interesting topics in the research community. We can gain deep insights about how to guide the coordination in massive scale to achieve goals with less cost in reality. Moreover, the digital footprints recorded in Wikipedia make such a study promising since they offer us every detailed information of this knowledge building process. In the following, we discuss the existing works focusing on the dynamics of online collaboration systems based on Wikipedia.

Coordination and Conflict in Wikipedia Buriol et al. [26] studied the evolution of Wikipedia by focusing on the dynamics of the wikilink structure and content updates. Their work showed that Wikipedia is still growing over time and has become denser in terms of contents and hyperlinks. Kittur et al. [76] investigated the interplay between the team structure and the nature of the collaboration task in the context of Wikipedia. They found out that adding editors may benefit low-coordination tasks but incur more cost for the tasks requiring a high degree of coordination. Their work pointed out that

decomposing the coordination dependencies can enable more efficient work by many editors. By analyzing the dynamics of revision history of a specific article, Nunes et al. [107] found out that the updates to the related article break out significantly when some event is taking place, both due to the flow of new information and to the emerging attention given to the article. For example, they observed that the article “**Michael Jackson**” received more contributions than before when the death of Michael Jackson was reported worldwide. Michael et al. [87] investigated the geographic relations between the contributors of an article in Wikipedia. They found out that by analyzing the edit history of Wikipedia contributors, it is often possible to infer the geographic regions where they presently live or were born. Their work indicated that in Wikipedia there are some implicit relations among the contributors that have edited some common articles. In Chapter 3, we also show that Wikipedia contributors reveal their interests and expertise through their contribution patterns. And we derive the expert-based similarity measure, which measures the semantic relevance between articles based on this hidden relation among contributors.

As knowledge building process is conducted collaboratively by people with different background and expertise, controversy and conflict normally emerge. How to resolve conflict plays an important role in the quality of Wikipedia and impact the passion of Wikipedia contributors. As traditional large-scale collaborations are often intractable, studying the conflict and controversy in Wikipedia is meaningful for us to understand the dynamics of collaboration process in massive scale. Brandes and Lerner [23] developed a visualization tool which reveals the dominant authors who are most involved in a controversy and who plays what role in the article building process. Subsequently Brandes et al. [22] offered an edit network derived from the edit history to illustrate the collaborative work of contributors in Wikipedia. They analyzed the interaction of the contributors in an article to characterize the role each individual contributor plays during

article writing. Potthast et al. discussed the characteristics of vandalism and developed a number of features related to identifying vandalism edits [115]. By training a classifier over these features, their experimental results showed that high precision and recall could be achieved.

Vuong et al. [145] proposed several models to measure the controversy in an article by the amount of disputes occurring in the article and the degree of controversy in each dispute. The models are designed based on the assumption that an article is more controversial if more disputes are from the less controversial contributors while a contributor is more controversial if s/he invites more disputes in less controversial articles. Such a model implicitly assumes that *the source of controversial articles is inherently the nature of the individual contributors*, rather than the subject matter of the article. With similar assumptions, Kittur et al. investigated a set of page metrics regarding the controversy characteristics [77]. These features include revisions, the length of content, number of contributors etc. Then a SVM classifier is trained accordingly. The experiments showed that the learnt classifier was able to rank the controversial articles consistent with their actual degree of controversy. Also, they demonstrated the use of visualization in exposing disputes between users. Similarly Le et al. analyzed the edit history of each individual article to cluster the contributors with concurring opinions, and with antagonistic relation with those with conflicting opinions [82]. Piotr et al. investigated the collaboration and teamwork conducted in Wikipedia to understand the factors that impact the success of open collaborative environments [144, 143].

Quality of Article in Wikipedia Several works exploit edit-related metadata for Wikipedia article quality evaluation [34, 54, 21]. Edit history was used by Ganjisafar et al. [45] to examine the article quality by ranking the search results based on the number of unique contributors of an article. They assumed that more people edited an

article, better the quality of the article is. Dalip et al. explored a significant number of quality indicators and studied their possibility to assess the quality of articles [54]. These indicators were extracted from three main aspects: text features, review features and network features. They showed through experiments, the combination of the structure and text features achieves the best performance in terms of prediction accuracy than the combination of the other features. The correlation between the number of contributors and the quality of articles was studied in [76]. Having more contributors at the formative stage of an article was shown to have positive correlation with the article's quality. They also observed that coordination done directly within the writing of the article improves the quality of the article, while explicit coordination using collaborative channels like talk pages is actually harmful.

Hu et al. [63] proposed three article quality measurement models to rank the quality of articles in Wikipedia based on their edit history. The basic assumption is that good articles are contributed by good contributors and good contributors usually contribute to good articles.

2.1.2 Exploring Semantic Information in Wikipedia

Wikipedia has an extraordinarily broad coverage of human knowledge and each Wikipedia article describes a specific entity or topic. The semantic internal structures of Wikipedia, including wikilinks, categorization and redirect and disambiguation pages, make it a fruitful resource for information retrieval and data mining tasks. In this section, we briefly survey the existing works that explore the semantic information of Wikipedia for different IR tasks.

Document Categorization and Clustering by Exploring Wikipedia Traditional document categorization and clustering approaches are developed based on the bag-of-word (BOW) model that ignores the order and combination of sequent words in the

text. Such restriction results in the limited performance since the loss of semantic value by considering only single words. Many researchers investigated the possibility that enriches the traditional document representation for document categorization and clustering [43, 61, 64, 147, 65]. Gabrilovich and Markovitch [43] proposed to enrich document representation based on the metadata of Wikipedia for document categorization. Each document is augmented by relevant Wikipedia articles and corresponding categories. Their study showed that this knowledge-intensive representation brings document categorization to a qualitatively new level of performance across a diverse collection of datasets. Wang and Domeniconi [147] made a further step by building a semantic kernel for document categorization based on Wikipedia. They embedded background knowledge by integrating the hyperlink structure of Wikipedia into a semantic kernel, which is intended to capture the associative relations of the semantic concepts defined in Wikipedia. The empirical evaluation showed that their approach significantly outperformed the traditional BOW techniques. Hu et al. [61] explored the semantic relations based on Wikipedia to enhance the document clustering. Specifically, the category and hyperlink information are used to enhance traditional content similarity measure. Similar approaches were also conducted by Hu et al. [64] based on the concepts and categories defined in Wikipedia.

Enhancing Web Search based on Wikipedia Many work proposed to improve the web search experience by exploiting the semantic resources encoded in Wikipedia [62, 125, 138, 156]. Hu et al. [62] proposed a random-walk based approach to learn the user’s query intent based on three semantic relations encoded in Wikipedia: article links, category links and redirect links. In their work, the query’s intent is aggregated by the Wikipedia articles mapped from the original search results. Xu et al. [156] proposed a query pseudo-relevance feedback (PRF) approach based on Wikipedia. They firstly derived a method to classify each query into three common types of the web search: 1) queries about a

specific entity (EQ), 2) ambiguous queries (AQ), and 3) broader queries (BQ). And the ambiguous queries are disambiguated to the Wikipedia article with the dominant sense. The terms for expansion are chosen based on *TF-IDF* scheme. The experimental results showed that PRF based on the relevant Wikipedia article brings along more relevant information and hence the performance can be improved significantly than the previous ones that based on the test collection. Moreover, they also investigated the effect of field evidence of the structured Wikipedia articles, and showed that the terms from the fields such as *Links* and *Content* can improve the performance further while the terms based on *Overview* lead to deterioration of the performance. Bin Tan and Fuchun Peng [138] addressed the problem of query segmentation by using generative language models and Wikipedia. They treated Wikipedia as a trusty external lexical resource to measure the quality of a specific segmentation based on the probability of that segmentation as being in titles and links in Wikipedia. Santamaria et al. [125] investigated the search result diversification based on the sense inventory of Wikipedia. They firstly associated each search result with a specific Wikipedia sense (article). Then the top ranked document from each sense are returned to promote diversity. The experimental results indicated that Wikipedia has a much better coverage of search results, compared to WordNet, and the approach by associating Wikipedia articles to the search results is efficient and effective to improve diversity in web search results.

Key Term Extraction based on Wikipedia Key terms (sometimes referred to as keywords or keyphrases) are a set of significant terms in a text document that give high-level description of its content for readers. It is a basic step for various tasks of NLP and IR, such as document classification, topic indexing and summarization. Traditional key term extraction methods focus on applying statistical measures to infer the importance of each key term. These methods often result in a less informative and understandable

results, since the score of each key term is calculated solely in terms of a specific statistical measure. As the largest digital encyclopedia, Wikipedia has been utilized by many researchers to tackle this problem. Grineva et al. [49] proposed a novel method for key term extraction from text documents. In their method, a document is modeled as a graph. They assumed that the terms related to the main topics of the document are strongly related to each other, resulting in densely interconnected subgraphs or communities; and non-important terms (noisy information) are weakly connected to any other terms or become isolated vertices. They leveraged the semantic information encoded in Wikipedia to identify key terms and derive semantic relatedness between these terms. Then they applied graph community detection techniques to partition the graph into thematically cohesive groups of terms. The experimental study showed that their method outperformed existing methods by producing key terms with higher precision and recall. Also, since their method is based on relatedness graph and graph partition, the method is more effective on noisy and multi-theme documents, such as web pages. West et al. [153] derived the missing topics in text documents that are relevant to the context of the document. The key step towards finding missing topics consists in generalizing from the hyperlink structure of Wikipedia using principal component analysis. Carmel et al. [27] proposed a method to label clusters of document by utilizing Wikipedia. They first extract important keywords and phrases from the text. These important terms are treated as candidate labels. Then a list of related Wikipedia pages are identified based on these important terms. The Wikipedia categories and titles of these related pages are considered as candidate labels as well. They aggregated several judgments to select the top candidates as the cluster labels. Their experiments based on ODP (Open Directory Project) and 20NG benchmark showed that the semantic value of Wikipedia is extremely helpful in selecting cluster labels.

2.2 Word Sense Disambiguation Using Wikipedia

Word sense disambiguation is the task of identifying the sense of a word / phrase within a specific context. Two main approaches can be found in the literature that try to address this problem, namely *knowledge-based* methods and *supervised machine learning* methods. The former tries to identify the correct sense by maximizing the agreement between the dictionary definition and the context of the given ambiguous word / phrase. The latter identifies the correct sense by applying a classifier trained on a set of local and global contextual features from a manually sense-tagged dataset. Both approaches suffer from the knowledge acquisition bottleneck problem: either a high-quality sense inventory or a substantial number of training examples are required. Because Wikipedia is the largest online encyclopedia and collaborative knowledge repository, it has become a paradise for the researchers in the field of natural language processing. In between, many studies explore the semantic resource of Wikipedia for word sense disambiguation, because Wikipedia provides both a high-quality sense inventory and a large number of human annotations. In the following, we review the related works in the two directions respectively.

2.2.1 Knowledge-based Methods

Mihalcea and Csomai addressed the problem of word sense disambiguation to Wikipedia in their Wikify! system [100]. Both knowledge-based and supervised machine learning methods were investigated. The knowledge-based method, inspired by the Lesk algorithm [83], utilizes the occurrences of ambiguous keyphrases and the contextual information. A Wikipedia topic that has the maximum overlap with the contextual words of the given ambiguous keyphrase is chosen as the correct sense. However, this method alone performed poorer than the baseline method using the most common sense. Medelyan et al. utilized both *commonness* and *relatedness* measures for disambiguation [98]. For

a candidate topic t , *commonness* for a given keyphrase k is defined as $P(t|k)$, *i.e.*, the prior probability of keyphrase k referring to candidate topic t [100]. For a given document, all keyphrases, each of which uniquely maps to one Wikipedia topic are identified as the context. The context is used then to disambiguate the keyphrases that each can map to more than one Wikipedia topic. In their work, relatedness to the context for each candidate topic of an ambiguous keyphrase is computed by using Wikipedia Link-based Measure [102] (WLM). A score is computed for each candidate topic t of a given keyphrase k to be the multiplication of commonness and relatedness. The topic with the highest score is chosen to be the disambiguated sense. Their approach significantly outperforms the most common sense baseline.

Recently, Ratnov et al. addressed the problem of disambiguation to Wikipedia by combining both local and global approaches with supervised learning [119]. In detail, the local context approach solves the disambiguation by choosing the topic that is the most similar to the input document containing the ambiguous keyphrase. The global approach, then, solves the problem by disambiguating the ambiguous keyphrases as being a coherent set of related topics. They implemented the local approach by using cosine similarity between the candidate topic and the context window of an ambiguous keyphrase. The global approach was implemented by measuring the relatedness between two Wikipedia topics using Wikipedia links (*i.e.* *WLM* and Pointwise Mutual Information (*PMI*)). The set of topics which is to be optimized as a coherent set is augmented by taking all topics of the ambiguous keyphrases that have been disambiguated by the local approach. It is reported that combining both local and global context approaches results in better disambiguation accuracy.

Our work [85] on disambiguation to Wikipedia filters away noisy contextual information, and applies a scaling factor to accommodate the relatedness measures with varying property (*i.e.* the dispersion of relatedness measure). The computation required is

greatly reduced because of the context pruning. Meanwhile, because noisy information is filtered away, the accuracy of disambiguation is improved as well. The experimental results showed that both better effectiveness and efficiency were achieved, compared to the state-of-the-art approaches. Moreover, the scaling factor added to the relatedness measure enables the approach to generalize well to different settings.

2.2.2 Supervised Machine Learning Methods

Mihalcea and Csomai also studied the supervised machine learning method in their Wikify! system [100]. The classifier is learned with a number of contextual features, such as part-of-speech tag and local neighboring words. Milen and Witten further extended this method by considering the cohesiveness of the context [103]. Several machine learning based classifiers are trained based on the relatedness to the context, the cohesiveness of the context, and commonness. While Medelyan measured the relatedness to the context by computing the averaged relatedness to all unambiguous keyphrases, Milen and Witten weighted each unambiguous keyphrase based on their relatedness to each other as well as their *keyphraseness*. *Keyphraseness* is the prior probability that a given keyphrase should be linked in a Wikipedia page. Higher keyphraseness indicates that the keyphrase is a concrete concept of human knowledge. They assumed that if the context is cohesive, the relatedness measure is more important for the disambiguation; otherwise, commonness would be a more significant indicator when the context is diverse. Their empirical study showed that C4.5 classifier achieved better disambiguation accuracy than Medelyan et al.'s method.

While the methods by Medelyan et al., Milne and Witten achieve a promising disambiguation accuracy to date, they measure the context relatedness by taking all unambiguous keyphrases identified in the given document into account. Although Milen and Witten applied a weighting scheme to highlight the more semantically related context keyphrases, it inevitably incurs additional computation. Ratnov et al. explored

the context information by applying Named Entity Recognition (NER) taggers and shallow parser, i.e. to restrict the context information by using named entities and nouns only. While their approach reduces computation in the disambiguation process, the additional computation incurred by applying NER tagger and shallow parser is expensive. As a document often contains some noise, not all unambiguous keyphrases are equally useful for expressing the main topic of the document. Therefore, some unambiguous keyphrases may even hurt the disambiguation accuracy besides incurring computational resources. Our previous work [85] applies a pruning scheme picking the most important keyphrases for use in the disambiguation process. This non-trivial step filters away shallow keyphrases and reduces noise in the context. In *TSDW* (Chapter 4), we extend our previous approach with a two-stage disambiguation process. Specifically, we explore the semantic clue contained in the ambiguous keyphrases in the second stage disambiguation process, by picking some high-confident disambiguation decisions from the first stage. While Ratnov et al. adopted the predictions of ambiguous keyphrases from a local context in their global approach, they augmented the topic set with all predictions of all ambiguous keyphrases. This introduces large computational cost. Similar works were also proposed for general word sense disambiguation [99, 15, 16], where the candidate senses of all ambiguous words are considered together. In *TSDW*, we augment the context in the second stage disambiguation by taking only a limited number of disambiguation decisions of high confidence from the first stage.

Note that *TSDW* can also be configured to have more passes of re-disambiguations, such as third stage disambiguation. The idea of multiple passes of inference to alleviate data sparsity problem is similar to bootstrapping [56, 80, 164] and semi-supervised learning [106, 133]. Nevertheless, in evaluation of *TSDW*, we will show that two-stage disambiguation is adequate for *D2W* task and further stages yield negligible gain.

2.3 Social Tagging System

Social tagging is a kind of online social media that many users add tags in the form of keywords to shared content [48], such as web page, pdf, image and video, and so on. Two organizational taxonomies for social tagging systems based on the system design and attributes and user incentives were proposed by Marlow et al. [94]. They claimed that the characteristics of the system design may affect the output of the system as well as users' incentives. The motivations for users to tag are mainly categorized into two high-level practices: organizational and social. Generally speaking, people assign tags based on the content of information items as well as their personal usage [128, 19, 48]. As users having different background and expertise, different facets of the information item are discovered collaboratively. Many works focusing on the dynamics of social tagging process reveal a lot about how people interact with each other. Halpin et al. [51] showed that in social tagging environments, people often emulate and share others' tagging behavior. This kind of tagging activities can be considered as a voting scheme that high quality tags regarding the semantic content of the item are often enhanced and encouraged. Suchanek et al. [134] investigated the meaning of tags based on the external knowledge base. Their work showed that the tags with high frequency carry more semantic information, while the less frequent tags are often for personal management, like *todo*, *reference*, etc,. Bischoff et al. [19] analyzed the use of tags for different collections and in different environments. They found out that most of tags belong to the category of topic that describe what a tagged resource is about. Moreover, they showed that most of the tags add new information to the tagged resources, especially for images and music. Also, by investigating the query logs of web search, they found out that most of the tags can be helpful for web search enhancement.

Ramage et al. [117] studied the semantic value of social tags in the context of web page clustering. Their work showed that by augmenting web page with tag information,

a better clustering performance can be achieved significantly. The wisdom of crowds emerges from social tagging environments motivates many researcher to leverage the semantic information of tags in many IR tasks. Heymann et al. [57] studied the possibility that social tagging activities can be leveraged for web search. They found that the ongoing tagging activities bring more new information than the search engines. And about 25% of URL shared by the users are unindexed by the search engines. Annotated tags are often used by people when they query information on Web. However, the study also showed that the current tagged resource constitutes only a small part of Web, given the annotated tags are relevant and objective. This analysis implies that given the tags are useful in general, the human effort in the collaboration environments still can't meet the speed of information growth. Efficient and effective tag recommendation methods are urgently desired in the sense. Fortunately, the research of the current social tagging systems is a starting point for such task.

Many work about recommending tags by exploring social tagging environments have been conducted extensively [96, 163, 120, 131, 58]. Heymann et al. [58] investigated the tag prediction problem based on three types of features: page text, anchor text and neighboring pages. Their studies showed that page text can be more useful for tag prediction. And it was found out that more general tags are harder to predict. Rendle et al. [120] modeled the ternary relations between users, items and tags by a tensor and applied tensor factorization to derive tags to the user of concern. Song et al. [131] applied two-stage framework for tag recommendation task for textual documents. The documents, words and tags are modeled by two bipartite graphs. At first stage, Spectral Recursive Embedding algorithm is applied to group the relevant documents to the tags. The documents within each group are further ranked based on their relevance. At second stage, a two-poisson model are applied to model the tag profile. Relevant tags are recommended based on the posterior likelihood of the given document. Yin et al. [163] proposed a

language model for personalized tag prediction. In their method, the effect of others' tagging behaviors are modeled based on the interplay between the users. Medelyan et al. [96] leveraged Wikipedia for topic tag identification. The keyphrases contained in the documents that has concrete meaning in Wikipedia are extracted as candidate tags. A bunch of features based on Wikipedia's semantic information, such as *keyphraseness*, the number of incoming and outgoing links, etc., are used to rank the relevant candidates. Their experimental result showed that Wikipedia is a promising resource that can be used to identify relevant tags. In our work (Chapter 5), we also explore the semantic information contained in Wikipedia for tag recommendation problem. We found out that representing documents in the concept space derived from Wikipedia can achieve better efficient and effective performance than the techniques based on BOW.

2.4 Named Entity Recognition for Tweets

Tweets are infamous for their error-prone and short nature. This leads to failure of many conventional NLP tasks, which heavily depends on local linguistic features, such as capitalization, POS tags of previous words, etc. Also acknowledging the error-prone nature of tweets, Han and Baldwin [52] proposed to normalize ill-formed words in tweets to make the contents more formal. However, this work does not address the problem of NER. NER has attracted renewed interests recently, due to the challenges posed by tweets. Conventionally, NER studies are mainly conducted in a supervised manner. In most of the cases, they depend on the Part-of-Speech (POS) tags, which again need a tagger to be trained with supervised approach based on linguistic features[118, 149, 168].

There are attempts that design linguistic features to capture tweets' unique characteristics and train tweet-specific models. Gimpel et. al trained a POS tagger with the help of a new labeling scheme and a feature set that captures the unique characteristics of tweets [47]. It was reported to outperform the state-of-the-art Stanford POS tagger on

tweets. In [122], Ritter et. al presented an tweet-based NLP framework which contains tweet-specific NLP tools: POS tagger (*T-POS*), shallow parsing (*T-CHUNK*), capitalization classifier (*T-CAP*), and end-to-end named entity recognition (*T-NER*). *T-POS* and *T-CHUNK* are trained by using conditional random field (CRF) model with conventional and tweet-specific features. These tweet-specific features include retweets, @usernames, hashtags, urls and the Brown clusters resulting from 52 million tweets. Both *T-POS* and *T-CHUNK* were reported with better performance compared to the existing state-of-the-art approaches. Since capitalization is much less reliable, *T-CAP* are designed to predict whether a tweet is informatively capitalized by using Support Vector Machine (SVM). *T-NER* is separated into two task: named entity segmenting (*T-SEG*) and named entity classification (*T-CLASS*). *T-SEG* is trained with CRF model. The training features include orthographic, contextual, dictionary features and the output by *T-POS*, *T-CHUNK* and *T-CAP*. They implement *T-CLASS* by applying LabeledLDA [116] with the external knowledge base Freebase¹. Liu et. al [88] applied a KNN-based classifier to conduct word-level classification, leveraging the similar and recently labeled tweets. Those pre-labeled results, together with other conventional features (e.g. orthographic and lexical features), were then fed into a CRF model to conduct finer-grained NER. Nevertheless, due to their supervised nature, those approaches require the availability of labeled data, which is usually expensive to come by. Finin et. al. presented a crowd-sourcing way (using services like Mechanical Turk and CrowdFlower) of preparing labeled data for NER studies in *Twitter* [39]. However, it did not propose a solution for NER.

Similar to *TwI-NER* (Chapter 6), Downey et. al also proposed a *collocation*-based approach, called *LEX* to detect the boundaries of named entities [36]. Nevertheless, it is not designed for tweet-like informal text. It assumes that named entities are either contiguous capitalized words or mixed case phrases beginning and ending with capitalized

¹<http://www.freebase.com/>

words, which is apparently too strong to hold in tweets. Silva et. al. [31] studied five different types of *collocation* measurement and their variations for phrase extraction task. Besides SCP measurement used in both *TwिNER* and *LEX*, there are another four types of *collocation measure*. And SCP performs the best among others.

Wikipedia is exploited as a source of *global context* in the work of Chapter 6. Existing attempts that exploit *Wikipedia* usually assume that named entities should have corresponding *Wikipedia* pages [75, 121]. This assumption makes them unable to identify emerging named entities that are frequently observed in tweets. However, there are many new named entity mentioned in tweets. In *TwिNER* (Chapter 6), information in tweets' *local context* and *global context* are aggregated to calculate the probability that a phrase is named entity. By doing so, *TwिNER* is able to recognize new named entities which may not appear in *Wikipedia* yet. To the best of our knowledge, it is the first to exploit both the *local context* (in *tweets*) and the *global context* (from World Wide Web) together for NER task in Twitter.

2.5 Event Detection from Tweet Stream

Event detection has a long history, which can be traced back to the Topic Detection and Tracking (TDT) project², which is to detect and track events from news stream. Two main approaches have been studied in the literature: *document-pivot* and *feature-pivot* approaches. The former aims to cluster documents related to the same events and then extract event-based features from the document clusters [17, 160, 159, 158, 24]. The latter aims to firstly identify the representative features of the hidden events from the stream, which are assumed to have bursty frequency patterns along time. Then events are detected by clustering these representative features [78]. Because the proposed *Twevent* (Chapter 7) is a feature-pivot method, in our literature survey, we therefore mainly focus on *feature-pivot* methods for event detection from formal texts.

²<http://projects.ldc.upenn.edu/TDT/>

2.5.1 Event Detection from Formal Texts

Kleinberg [78] proposed to detect events by analyzing frequency patterns along time. An infinite-state automation is used to model the changes of word frequency, and the state transitions are considered as events. Fung *et al.* [42] proposed to identify bursty features as representatives for the events hidden in text stream. The frequency of each feature (*i.e.*, unigram word) is modeled with a binomial distribution. The bursty feature extraction is then based on the perspective of statistics. The events are then detected by maximizing the co-occurrences among documents and the consistence of the frequency distributions for all bursty features within an event. The timestamp for an event is calculated based on the bursty periods of the bursty features related to that event. The authors further presented an event-based search framework in [41] to retrieve groups of documents such that the documents in each group are about the same event. In their result, the events are organized in a time-based hierarchy. In this event-based framework, a set of related bursty features with similar frequency distributions are retrieved firstly. Then, documents related to the bursty features are extracted and clustered into a hierarchy of events. Instead of using frequency directly, He *et al.* [55] proposed to use *Discrete Fourier Transformation* (DFT) to extract bursty features. They build a signal for each feature using *document frequency - inverse document frequency* ($df \times idf$) scheme along time domain. Then, DFT transforms the signal in time domain to frequency domain, *i.e.*, a spike in frequency domain indicates a corresponding high frequency signal source. Similar to [42], they group bursty features into events by considering both features' co-occurrence and their distributions in time domain. To estimate the timestamp of the events, they model a $df \times idf$ signal with a Gaussian mixture.

Unlike formal texts that are formally written and published in moderate rate, tweets are short, informally written, and published at an enormous amount. Thus, bursty feature extraction solely based on statistics would result in a huge number of bursty features,

particularity when unigram feature representation is used. Similarly, the application of DFT would be dread and prohibitive. Further, co-occurrence measures used in [42, 55] may not work well in the context of twitter due to sparsity.

2.5.2 Event Detection from Tweets

Recently, event detection on twitter stream becomes a hot research topic. Michael and Nick [95] presented a trend detection system over twitter stream. They firstly identify the bursty terms based on queueing theory. Then bursty terms are grouped into the events based on their co-occurrences. For a detected trend, PCA, SVD and entity extraction techniques are then applied to derive contextual information for the trend description. Petrović *et al.* [111] tracked events on twitter stream by applying locality sensitive hashing (LSH). LSH is applied to each tweet to measure the similarity to existing tweets. The tweets similar to each other are grouped as events. Swit and Tsuyoshi [112] proposed an approach for breaking news detection and tracking by clustering the similar tweets together. The approach only focuses on the tweets with a specific hashtag #breakingnews. The similarity between two tweets of breaking news is measured by using a variant of $tf \cdot idf$ scheme where the named entities detected by a Named Entity Recognizer (NER) are further boosted. Popescu *et al.* [114] proposed a method for entity-based event detection on twitter streams. A set of tweets containing the predefined target entity are processed and machine learning techniques are used to predict whether the tweets constitute an event regarding the entity. Very recently, Li *et al.* [123] proposed to detect crime and disaster related Events (CDE) from tweets. Conventional text mining techniques are applied to extract the meta information (*e.g.*, geo-location names, temporal phrase, and keywords) for event interpretation. To summarize, most existing approaches for detecting events from tweets are applicable to certain types of tweets (*e.g.*, having a specific hashtag, containing a predefined entity, or related to crime and disaster). The

other solutions including [95] and [111] involve complicated processing and lead to heavy computational cost.

The most related work to ours, is the approach proposed by Weng and Lee, named *EDCoW* [151]. There are three steps in their approach. Firstly *wavelet transformation* and *auto correlation* are applied to measure the bursty energy of each word. The words with outstanding high energies are retained as event features. Then they measure the similarity between each pair of event features by using *cross correlation*. At last, modularity-based graph partitioning is used to detect the events, each of which contains a set of words with high *cross correlation*. However, several issues get in the way of the practical application for their approach. *Wavelet transformation* and *auto correlation* for each word of the twitter stream would require a huge amount of computation, making it not scalable. Moreover, utilizing only *cross correlation* for similarity measure would lead to the resulted event consisting of several distinct events that happened at the same period by coincidence (*e.g.*, two football matches hold at the same time during FIFA 2010 World Cup). Thirdly, the detected events with unigram features are difficult for human interpretation. In *Twevent* (Chapter 7), we segment each tweet into possible semantic phrases, making the detected events easy to interpret. During the detection process, we do not employ computational costly Wavelet transformation and auto correlation for tweet segments. Instead, only the tweet frequency and user frequency are needed for bursty tweet segment detection. To distinguish events that happened at the same period, *Twevent* computes content similarity for a pair of tweet segments. Each tweet segment is described by the content of the tweets containing the segment. Although pair-wise similarity computation is computational costly, it is only applied to a relatively small set of bursty tweet segments detected within one time window.

Chapter 3

Mining Latent Relations: A Case Study with Wikipedia Article Similarity and Controversy*

Let go of your attachment to being right, and suddenly your mind is more open. You're able to benefit from the unique viewpoints of others, without being crippled by your own judgment.

– Ralph Marston

From infotainment sites to citizen reporters, blogs, Q&A sites such as Yahoo! Answers to Wiki-based encyclopedic corpus like Wikipedia, in recent years social media has become an integral part of our daily life. Compared to traditional websites, social media sites enable more user interactions with information items as well as with other users, like sharing photos, tagging webpages, submitting and commenting on news stories, as well as making friends online. These interactions interplay with each other, transforming social media sites into complex networks of peer-production environments. Latent relations encoded within these interactions may provide us with a new avenue to discover new knowledge that may complement our understanding about the system of interest based

*This chapter is based on the paper *Mining Latent Relations in Peer-Production Environments: A Case Study with Wikipedia Article Similarity and Controversy* by Chenliang Li, Anwitaman Datta, Aixin Sun, published in *Social Network Analysis and Mining* 2(3), Springer, 2012.

on the declared explicit relations. Also, harnessing information from mining different relations based on the different perspectives would help us discover knowledge that can't be deduced by considering various aspects in isolation.

Wikipedia, as a multilingual, web-based, free-content encyclopedia, has more than 4.0M articles in English, attracting around 470M visitors monthly as of February 2012¹. Moreover, Gartner, the Wall Street Journal, and Business Week have all identified the Wikipedia paradigm of 'peer-production' of knowledge repository as an up-and-coming technology to support collaboration within and between corporations. Enterprise wiki has been increasingly adopted in companies and organizations.²

Wikipedia's open strategy, which allows anyone to create and edit articles, leads to its unsurprising success. The open access property makes knowledge creation in Wikipedia a dynamic process, evolving over time by contributions and collaboration among different people. It is arguably an outcome and ongoing venue for the most massive collaboration online to date. Furthermore, all the actions of every contributor are logged meticulously, and are also openly available for analysis. The *edit related* metadata information can be used to help us understand the collaboration dynamics of Wikipedia. Specifically, these can be analyzed to obtain a deeper and clearer insight on the characteristics of contributors as well as articles.

In this chapter, we investigate similarity measures over Wikipedia articles based on different perspectives of the collaborative knowledge building system enabled by Wikipedia. For example, by analyzing the logs of edit history, we observe that individual contributors only edit a relatively small number of articles. This shows that *people have only focused expertise and/or interest areas with respect to the areas covered by the entire Wikipedia*. Based on this observation, we propose a similarity measure, *expert-based similarity*, to evaluate the relevance of articles among each other. In contrast, existing

¹<http://en.wikipedia.org/wiki/Wikipedia:About>

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

state-of-the-art approaches that have been adopted widely in Information Retrieval (IR) areas to quantify the relevance focus on other perspectives such as content-based [92] and structure-based similarities [69, 167].

We conduct a case study to illustrate that knowledge from different perspectives can be obtained by mining the various relations among Wikipedia articles. We show that by combining knowledge obtained from such different perspectives (essentially based on the different alternate measures), we can better discern the origin of some controversies in Wikipedia, which can not be deduced by considering any single particular aspect in isolation.

To the best of our knowledge, the work of this chapter is the first attempt to explore different relations encoded in Wikipedia by studying the *edit related* metadata.

In Section 3.1 we propose a new measure, *expert-based similarity* to evaluate relevance relationship between articles based on the observation that Wikipedia contributors often edit a small number of articles each. We additionally discuss the relevance relationships between Wikipedia articles based on content, hyperlinks, each of which focuses on one specific perspective.

In Section 3.2 we conduct extensive experiments to evaluate different similarity measures. Specifically we evaluated the similarities based on expertise, and existing metrics such as cosine similarity, P-Rank and SimRank.

In Section 3.3 we cluster articles based on the various similarity measures, and study the distribution of controversial articles in the resulting clusters. We carry out a detailed analysis on the distribution of the controversial articles in the resulting clusters. From our analysis, at least in the data sets considered, we determine that out of several plausible hypothesis for controversies [22, 23, 77, 145], one reason dominates. Namely, the particular controversial topics contained in articles are the principal source of controversy in Wikipedia, instead of the disputes caused by the aggressive contributors or controversial concept in general. Finally, we conclude this work in Section 3.4.

3.1 Article Similarity in Wikipedia

Wikipedia is the largest online encyclopedia and collaborative knowledge building system. Compared with the traditional encyclopedia that are edited by a limited number of experts, it allows everyone in the world to contribute and share their knowledge. Thus, vandalism and incorrect description are spotted and corrected in a very short time by the wisdom of crowds in online communities. Also, traditional encyclopedias are sequential, i.e. ordered along alphabetical, topical or historical lines [153], while Wikipedia articles have rich semantic structures that every article is connected to other articles by the hyperlinks annotated by the contributors on the places where the concepts are mentioned in the original article. Based on the Wikipedia policy, the hyperlinks in an article should be created to relevant topics of the article and technical terms mentioned that are likely to enhance reader’s understanding. Moreover, articles are also organized by Wikipedia categorization system, which an article may be assigned to at least one category based on the concepts it covered. Figure 3.1 depicts the above explicit relations existing in Wikipedia.

Besides the explicit relations mentioned above in Wikipedia, we can also derive latent (implicit) relations hidden from the explicit ones, e.g. expert-based similarity which will be explained. In the following we first propose a implicit relations in relation to the relevance of Wikipedia articles, expert-based similarity, we then discuss relevance relationships based on content and hyperlinks.

3.1.1 Expert-based Similarity

A Wikipedia article is an artifact that evolves from the contributors’ contributions, which induce interactions among these contributors. In the process, contributors also manifest their expertise and interest by making contributions to related articles. By analyzing the distribution of the number of revisions and the articles edited by contributors, we

Figure 3.1: Explicit relations among articles and contributors in Wikipedia

find that most contributors only edit a small number of articles, making limited number of revisions. Furthermore, contributors often have very biased contribution focus. That is, if a contributor has contributed to a large number of revisions, then it is likely a large portion of these revisions went to very few articles [166]. From this observation we infer that individuals have focused interest and familiarity with topics which they frequently contribute.³ Specifically, in our dataset, on an average each contributor edits only 6 articles. This provides the intuition that an article normally contains a topic that attracts a limited scope of readers and contributors. The scope is article dependent.

A two-way implicit selection process can be identified here: *contributors choose articles relevant to their focused expertise and each article has only limited audience who is expert or interested in the related subject matter*. Based on these observations, we hypothesize that two articles are similar to each other (up to a certain degree) if they have been edited by the same contributors. Thus, we use the commonality of contributors

³Note for clarification: From our analysis, we noticed that same contributor may actually contribute to many articles spread across different unrelated categories (and in that sense, the focus is not limited) - e.g., on articles related to American football, Scientology, and Biology, but within each specific category, the contributions are few and rather focused. The rest of the discussion pertains to contributions within one such category, namely “Religious Objects”, on which we carry out our case-study.

Table 3.1: Symbols and semantics

Symbol	Semantic
a, c	the instance of a variable: a for article, c for contributor
V_a	a collection of articles
V_c	a collection of contributors
$r_{c,a}$	the number of revisions of article a made by contributor c

of two articles to determine similarity among these articles based on a metric that we call the *expert-based similarity*. Such a measure complements other similarity measures like content similarity (measured using metrics like cosine similarity) as well as network neighborhood analysis based on graph topology induced by hyper-links. Table 3.1 lists the main symbols that we use in the following.

Formally, the *expert-based similarity* is an implicit relation between Wikipedia articles based on the explicit editorship relation, i.e. $r_{c,a}$ (see Fig 3.1 & Table 3.1). Let $\mathcal{C}_{c_i,a_j}$ represents the contribution score of contributor c_i in article a_j , the *expert-based similarity* for articles a_u and a_v is calculated using the standard metric of cosine similarity defined below, where $1 \leq u, v \leq |V_a|$

$$s_e(a_u, a_v) = \frac{\sum_{c_i \in V_c} \mathcal{C}_{c_i,a_u} \mathcal{C}_{c_i,a_v}}{\sqrt{\sum_{c_i \in V_c} \mathcal{C}_{c_i,a_u}^2} \sqrt{\sum_{c_i \in V_c} \mathcal{C}_{c_i,a_v}^2}} \quad (3.1)$$

The contribution score $\mathcal{C}_{c_i,a_j}$ is chosen based on the context and application. It can be the number of words added by the contributor, the number of revisions made by the contributor, or other quantities calculated by other measures or algorithms. Throughout our study, we define the contribution score to be the number of revisions a contributor has made to the article multiplied by the significance score of the contributor across the collection. The formula to calculate the contribution score, $\mathcal{C}_{c_i,a_j}$, is as follow:

$$\mathcal{C}_{c_i,a_j} = r_{c_i,a_j} \times f_{c_i} \quad (3.2)$$

where f_{c_i} refers to the significance of contributor c_i . Here, we simply define the significance score of contributor c_i as follows:

$$f_{c_i} = \log\left(\frac{|V_a|}{\sum_{a_j \in V_a} I(r_{c_i, a_j} > 0)}\right) \quad (3.3)$$

where the boolean function $I(x)$ takes value 1 when x is true, otherwise 0, i.e., whether contributor c_i has edited article a_j . Equation 3.3 is analogous to the inverse document frequency (*IDF*) in the widely adopted $TF \times IDF$ weighting scheme. A contributor has a lower significance score if s/he has edited more articles. The boolean function is used here since, for the considered data-set, contributors only edit few articles very relevant to their expertise. Although the number of revisions r_{c_i, a_j} is not a very fine grained discriminative feature compared to the number of contributed words, it holds enough discriminative ability to measure the similarity of two articles in that contributors on average make 13 revisions. By using the number of revisions made by a contributor to an article, the computation cost of calculating the expert-based similarity between two articles is substantially decreased since it ignores the details associated with each revision.

3.1.2 Article Relevance Aspects

As being discussed previously, the relevance relationship between Wikipedia articles can be evaluated by similarity measure defined based on the perspective of the co-editorship, (e.g., *expert-based* similarity). Other than from the perspective of contributors, similarity explicitly evaluated based on the textual content of the articles also help us understand the relevance relation from the perspective of textual content. Furthermore, articles are connected by hyperlinks in their content. Recall that, the link structure in Wikipedia is quite different from the hyperlinks in traditional web pages since most of the internal links in Wikipedia point to semantically related content [72, 73].

In this chapter, we evaluate the relevance relationship between Wikipedia articles based on content, hyperlinks, and co-editorship, and the similarity metrics used are summarized in Table 3.2. Among them, content based cosine similarity, which is widely used in IR and related areas, is an explicit relation as discussed previously, i.e., the cosine similarity of bag-of-word models of two articles a_i and a_j . To capture the relevance information from the perspective of the link structure, we use SimRank/P-Rank similarities which are described next.

SimRank [38] is based on the intuition that similar objects are related to similar objects, which in turn are mutually similar. More precisely, objects a and b are likely similar if they are related to objects c and d respectively, and objects c and d are themselves similar. Formally, given two objects a and b in a graph, let $I(a)$ and $I(b)$ be their corresponding sets of in-link neighbors. The SimRank between a and b is computed recursively by Equation 3.4. In this equation, $I_i(a)$ is the i^{th} in-link neighbor of a ; C is a damping factor between 0 and 1. The equation is initialized by setting $s(a, b) = 1$ if $a = b$ and $s(a, b) = 0$ otherwise.

$$s(a, b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \quad (3.4)$$

P-Rank [167] extends SimRank by considering the out-neighbors between two objects, see Equation 3.5. In this equation, $I(a)$ and $O(a)$ denote the set of in-link neighbors and the set of out-link neighbors of object a respectively. Similar to that of SimRank, C is the damping factor and λ is a variable for the weight of similarities derived from in-link neighbors and out-link neighbors respectively.

$$\begin{aligned} s(a, b) &= \frac{\lambda \times C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \\ &+ \frac{(1 - \lambda) \times C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} s(O_i(a), O_j(b)) \end{aligned} \quad (3.5)$$

Table 3.2: Similarity aspects and metrics

Relevance aspect	Similarity metric	Relation type
Content	Cosine similarity	Explicit
Hyperlink	P-Rank, and SimRank similarities	Implicit
Co-editorship	Expert-based similarity	Implicit

To evaluate P-Rank and SimRank, a clustering evaluation measure named *compactness* was used in [167]. The results from their extensive experiments showed that P-Rank consistently outperformed SimRank.

3.2 Article Similarity Evaluation

We conduct three sets of experiments to evaluate expert-based similarity against alternative similarity measures - cosine similarity using article content, and SimRank and P-Rank based measures using link structure among articles. We also validate the generalization of the similarity measures to larger scale.

In the first set of experiments, we adopt the methodology presented in [167] where *compactness* was used to evaluate the clusters produced by K -Medoids clustering algorithm using SimRank and P-Rank, respectively. As the results of K -Medoids may be impacted by the K chosen, we also apply DBScan [38], which does not need the number of expected clusters as input.

In the second set of experiments, we utilize the category labels from Wikipedia as partial ground truth to evaluate the clusters produced by K -Medoids and Agglomerative Hierarchical Clustering (AHC) [139] algorithm using various similarity measures. With these two clustering algorithms, the number of expected output clusters can be set to be the same as that of the ground truth. The experimental results are evaluated using purity and entropy [92].

While the performance of expert-based similarity is evaluated based on a specific Wikipedia category above, we validate the generalization of expert-based similarity to

the entire Wikipedia in the third set of experiments, to test its scalability.

Before we report the experimental setting and results, we describe the dataset used in the experiments.

3.2.1 Dataset

In related existing studies [145], *Religious Objects* was identified to have many controversial articles. Since we attempt to study controversy in Wikipedia as a case study, we used a similar set of articles in our experiments.

We extracted articles under *Religious Objects* category from the English Wikipedia dump generated on 03 January 2008. The dump is consisted of edit history of articles from August 2001 to January 2008. Since our dataset is extracted from a more recently generated dump than [145], the number of articles in our dataset is slightly different. In our dataset, there are a total of 18,973 articles, 69,481 registered contributors and 891,231 revisions after we filter the revisions made by anonymous contributors.

3.2.2 Evaluation with Compactness

Evaluation metric. *Compactness* metric was used in [167] to measure the quality of clustering results by considering *intra-cluster* distances and *inter-cluster* distances of the clusters. Formally, *compactness* is defined as:

$$C_{\mathcal{F}} = \frac{\sum_{i=1}^K \sum_{x \in C_i} d(x, m_i)}{\sum_{1 \leq i < j \leq K} d(m_i, m_j)} \quad (3.6)$$

where $\mathcal{F}$ denotes the specific similarity measure adopted, and $d(x_1, x_2)$ denotes the distance between two data points x_1 and x_2 with regards to the measure $\mathcal{F}$. In this equation, K is the number of clusters generated by a clustering algorithm; C_i is the i -th cluster; m_i and m_j are the centers of the clusters i and j respectively. Note that the numerator and denominator represent the *intra-cluster* and *inter-cluster* distances respectively. A

smaller compactness value means that the clustering results reflect the inherent relationships of the data better.

Similarity Computation. For a fair comparison between the four similarities, we used the articles with more than 5 distinct contributors, leading to 15,018 articles in our experiments. This automatically also ensured that each article had more than five revisions.

- The pair-wise expert-based similarity was then computed for these 15,018 articles according to Section 3.1.1.
- For content based similarity, the last five revisions for each article were aggregated to represent it since only the last revision alone may not fully represent an article due to the fast evolving nature of Wikipedia.
- For P-Rank and SimRank based similarity computation, only hyperlinks that existed in at least three revisions among the latest five were considered. We set the damping factor $C=0.8$ for both SimRank and P-Rank, and the relative weight λ is set to 0.5 for P-Rank. The calculations were run until the scores converged.

Experimental Results by K-Medoids. *K*-Medoids is a center-based partitional clustering algorithm which is similar to the widely used *K*-Means [90] algorithm. Different from *K*-Means, *K*-Medoids chooses the most representative points as centers of clusters [139]. These representative points are called medoids. The medoids are chosen with respect to some measure, e.g., minimize the sum of the distance of a point from the medoid of the cluster.

Similar to that in [167], we ran *K*-Medoids clustering algorithm over the dataset with the 4 similarity measures respectively with *K* set to 10. In total 10 trials were performed to minimize the impact of randomly selected initial medoids. Figure 3.2 plots

Figure 3.2: Compactness vs. Number of Trials for expert-based similarity, cosine similarity, P-Rank and SimRank

the compactness values of the four similarity measures w.r.t the 10 trials. We note that the expert-based similarity is consistently the best in the compactness measure. SimRank achieves the worst compact clustering results, while the cosine similarity outperforms P-Rank in 8 out of 10 trials. It partially confirms that the expert-based similarity measure is robust and achieves better performance in the Wikipedia context. In order to further evaluate the performance of the expert-based similarity, we run the same experiments using DBScan algorithm.

Experimental Results by DBScan. The DBScan algorithm clusters the data points by finding clusters of points that are density-reachable to each other within the cluster [38]. In contrast to the K -Means and K -Medoids algorithms, which partition all data points into k clusters, the DBScan algorithm groups the points within an area with the high density into a cluster and classifies the points of the low density areas as noise. A noise point does not belong to any cluster and can be considered as an outlier. Two parameters are required in DBScan: (1) ϵ is the radius of the neighborhood of a point; (2) $minPts$ specifies the minimum number of points required to form a cluster. The suggested way of setting the parameters is given in [38], which involves plotting a sorted

Table 3.3: Compactness for similarity measures w.r.t percentages of the noise

Similarity	Percentage of Noise			
	45%	60%	75%	90%
Expert Based	5.093	0.603	0.105	0.035
Cosine	56.540	6.613	0.719	0.170
P-Rank	38.448	6.450	0.707	0.492
SimRank	587.104	44.985	7.402	2.090

k -dist graph. All points with a higher k -dist value are considered to be noise in the clustering whereas the remaining points are assigned to some clusters. Hence the percent of noise points can be specified and the parameters can be derived correspondingly. In our experiments, we picked four points that set the percentage of noise to be 45%, 60%, 75%, 90%, by fixing the *minPts* to 4 respectively. We conduct clustering with the four similarity measures and summarize the results in Table 3.3. We see that expert-based similarity again outperforms the rest while SimRank remains the worst in these experiments. The cosine similarity and P-Rank have similar performance for the four parameter settings.

Manual Verification. From the clustering produced by DBScan, we examine the clustering results manually to verify whether the expert-based similarity reflects the reality. Figures 3.3(a) and 3.3(b) show two clusters produced by DBScan with the percent of noise being 60%. As illustrated in Figure 3.3(a), the cluster 1 consists of six Pathis⁴ and a worship place (Nizhal Thangal⁴) of Ayyavazhi, and a famous theertha of the temple (Muthiri kinaru) which is located half a kilometer west of Swamithope pathi⁴. Ayyavazhi is a dharmic belief system that originated in South India in the 19th century⁴. Pathi is the name given to the primary centres of congregational worship for Ayyavazhi, having a relatively large structure like that of a temple. There are 7 Pathis in the Ayyavazhi religious system. Vakaippathi⁴, the one missing from the cluster result is not included in

⁴http://en.wikipedia.org/wiki/{Pathi,Nizhal_Thangal,Muthiri_kinaru,Ayyavazhi,Vakaippathi}

No.	Article
1.	Muthiri Kinaru
2.	Poo Pathi
3.	thamaraikulam Pathi
4.	Nizhal Thangal
5.	Ambala Pathi
6.	Mutta Pathi
7.	Pancha pathi
8.	Swamithope pathi

(a) Cluster 1

No.	Article
1.	Metropolitan Community Church East London
2.	Metropolitan Community Churches in London
3.	Metropolitan Community Church
4.	Metropolitan Community Church in South London
5.	Metropolitan Community Church of New York
6.	Metropolitan Community Church of Toronto
7.	Metropolitan Community Church of Manchester
8.	Metropolitan Community Church of Edinburgh

(b) Cluster 2

Figure 3.3: Two example clusters

the dataset since it had only 5 contributors up till 03 Jan 2008 as explained previously. Figure 3.3(b) shows another cluster, which consists of articles relevant to the metropolitan community church. The clusters are quite intuitive and self-explanatory after human inspection. The conclusion over all experiment results is that expert-based similarity is indeed an effective measure to find semantically relevant articles in Wikipedia. Although the expert-based similarity is proposed and evaluated in the Wikipedia context, it is also applicable to other online social collaborative systems - and is expected to be useful to identify and recommend experts, for example in a forum or a Q&A system.

3.2.3 Evaluation with Partial Ground-truth

The evaluation with compactness reported in the above section reflects the structural cohesiveness of the clusters produced by the clustering algorithm with the chosen similarity measure. However, the semantic cohesiveness of the articles is not captured by the compactness measure. In this section, we report on experiments evaluating the semantic cohesiveness of the clusters utilizing the category labels in Wikipedia, which are manually assigned. We used 3033 articles (subset of the 15018 articles used in Section 3.2.2) from 39 categories in Wikipedia and evaluate the clustering results with the four similarity measures respectively. Since each article has a class label (*i.e.*, category), we evaluate the clustering result by using the standard clustering validity indexes: Purity and Entropy. In this set of experiments, K -Medoids and AHC were used since the number of expected clusters can be specified for both clustering algorithms to be same as that in the ground truth.

Dataset. A category in Wikipedia contains a list of member articles and a list of sub-categories within it, as well as its parent categories. However, as categories in Wikipedia can be created based on topic, location, time, etc., the categories do not form a strict hierarchical tree but rather a graph. In this experiment, we extracted all descendent categories under “Religious objects”, containing a total of 2708 categories. We then manually selected 39 categories, each of which refers to a specific topic. We do not consider the content of these categories during the selection. The only criterion is to make sure that two categories have minimum overlap. For each of the selected 39 categories, the articles belonging to it and its sub-categories are considered to form one cluster. The articles that belong to more than one category are ignored. We finally have 3033 articles from these 39 categories, which we consider to reflect the ground truth.

Experimental Results by K -Medoids. We perform the clustering using the K -Medoids algorithm over the 3033 articles, with K set to 39. The initial medoids are

Table 3.4: K-Medoids clustering results

Similarity	Purity	Entropy
Cosine	0.7592	0.9817
Expert Based	0.7312*	1.2695*
P-Rank	0.7689	1.0383
SimRank	0.6632*	1.6279*

chosen randomly from the 39 categories respectively. 10 trials were performed with different sets of initial medoids and the average was taken as the result to minimize the effect of the initialization. For each trial, the same initial medoids were used for the four similarity measures. Table 3.4 shows the experimental results by using the four similarity measures. The bold values indicate the best performance obtained for purity and entropy measures. The symbol * indicates the change is significant according to the paired t -test at the level of $p < 0.05$, compared to the cosine similarity. From Table 3.4, we can observe that the cosine similarity achieves the best performance according to the entropy, while P-Rank outperforms others in the measure of purity. The changes between the cosine similarity and P-Rank are not significant according to both purity and entropy. The expert-based similarity achieves much better performance than SimRank. And a closer performance is obtain compared to cosine similarity and P-Rank. As consistent with the compactness measure, SimRank achieves the worst performance in the experiments.

Experimental Results by AHC. We perform agglomerative hierarchical clustering using the CLUTO package⁵ with the complete link function. Table 3.5 lists the results of agglomerative clustering. The bold values indicate the best scores obtained for purity and entropy measures. From Table 3.5, it is observed that the cosine similarity achieves the best performance according to both the purity and entropy, while the P-Rank performs worst in the experiment. For the expert-based similarity, it achieves 21.8% increase in

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

Table 3.5: Agglomerative clustering results

Similarity	Purity	Entropy
Expert Based	0.625	0.448
Cosine	0.725	0.199
P-Rank	0.513	0.576
SimRank	0.543	0.510

purity and 22.2% increase in entropy as compared to the P-Rank. While the SimRank only improves the performance by 5.8% and 11.5% as regards to the P-Rank.

From the experimental results from *K*-Medoids and AHC clustering, we can see that P-Rank and SimRank achieve the modest performance. We think the main reason is the deviation of the relevance carried by the hyperlinks as well as the missing links among articles. In detail, some articles are underlinked so that many relevant topics are not linked together, while some articles are overlinked which result in the existence of the links of less value. Some previous works have been conducted to address such problem [13, 153]. Thus, the structure based similarity measures, such as P-Rank and SimRank, suffer from the noisy links in the Wikipedia context. As shown in the experiment of *K*-Medoids, by picking an article from a category as a initial medoid can decrease the impact of the noisy links, which may bring about the topic drift for some cluster to some extent. Comparing to the cosine similarity, the expert-based similarity performs much worse in the measure of entropy than in purity (see Tables 3.4 and 3.5). This indicates that a small number of articles from the different categories also share common authors to some extent, which makes them clustered into the same group. We think the reason is that the ground truth is selected from a dataset of high cohesion. Thus there is no explicitly strict boundary between different categories. This is due to the ground truth we build as well as the nature of the dataset we study on. However, combined the experimental results together, we can conclude that the expert-based similarity is better than hyperlink based similarity in most cases and not too far from the content based similarity.

3.2.4 Evaluation with Linear Correlation

Since the previous evaluation is conducted based on articles from a specific category, *Religious Objects*, it is arguable that expert-based similarity may only work well in some specific categories. In order to validate that expert-based similarity generalizes well, we measure the linear correlation between expert-based similarity and the most widely adopted measure, cosine similarity using a corpus of Wikipedia articles spanning all across the Wikipedia. We sample 100 random articles as seeds from the whole Wikipedia. In detail, given an article, we first extract the articles within its neighborhood of 2-hops by following the out-going hyperlinks of Wikipedia. We call the original article as a seed. The similarity scores between the seed and its neighbors are calculated using cosine similarity and expert-based similarity respectively. Then we measure the Pearson’s linear correlation between the values generated by the two measures. Each seed article has on average 6,458 neighbors within the network of 2-hops. The average correlation coefficient for these 100 seed articles is 0.4012 ± 0.1913 . This indicates that the two measures are correlated with each other positively. Moreover, it also indicates that the two measures expose different information to some extent. We further split the neighbors of a seed into 5 segments based on the cosine similarity (i.e. each segment with a value range of 0.2). And we calculate Pearson linear correlation coefficient for each segment separately. The average correlation coefficient for each segment are reported in Table 3.6. From Table 3.6, we can see that expert-based similarity is more strongly correlated with cosine similarity at the two extreme ends than in the middle-range. It means that when articles are too similar/dissimilar, the two similarity measures concur, while for the other articles, each similarity measure can tell us something different based on different perspectives. We have repeated the same experiments with different samples 10 times, similar results are observed. Thus, we conclude that expert-based similarity generalizes well.

Table 3.6: Correlation within five segments of cosine similarity

Segment	Correlation
[0.0, 0.2)	0.3033
[0.2, 0.4)	0.1518
[0.4, 0.6)	0.0239
[0.6, 0.8)	0.0155
[0.8, 1.0]	0.2562

In summary, we compare the performance of the four similarity measures by using agglomerative hierarchical clustering and K -Medoids clustering on the ground truth we build from the Wikipedia’s category system. We do not intend to demonstrate that a specific similarity measure is better. Instead, we evaluate the appropriateness of the similarity measures with regard to the different aspects and purposes in the context of Wikipedia. Moreover, our experiments validate that expert-based similarity is an effective metric to quantify relevance relationship between articles in Wikipedia.

3.2.5 Qualitative Comparison

Wikipedia keeps a history of all operations from contributors. Thus, the contribution matrix needed to compute expert-based similarity score is obtained without much overhead cost. P-Rank and SimRank are computationally expensive because of the iterative calculation process involved and the huge size of Wikipedia. Moreover, with any change in the structure which may not even concern directly the articles being compared, the measure needs to be recomputed. Wikipedia’s open strategy could be misused by some editors. Vandalism, defined by Wikipedia, as “any addition, removal, or change of content made in a deliberate attempt to compromise the integrity of Wikipedia”. Cosine similarity can’t tolerate the effect of vandalism. On the contrary, the IDF scheme used in the expert-based similarity to attenuate the effect of editors that edit too many articles to be meaningful for relevance determination can make it more defensible against vandals. For example, Wikipedia account “Ananny” tried to promote the work of a non-notable artist

by adding the artist’s name to different Wikipedia articles. This behavior has affected 19 Wikimedia wikis and last for over 5 years⁶. By using IDF scheme, the significance score of this contributor could be negligible small, and hardly affect the calculation of the expert-based similarity. The expert-based similarity measure is thus more efficient and robust in comparison.

3.3 Source of Controversy

In the previous section, we evaluate the performance of relevance measures over Wikipedia articles from different perspectives. We see that focusing on different perspectives results in the varying measure qualities, respectively. Moreover, since a social network consists of multi-dimensions as well as explicit or implicit relations associated with, relying on only one dimension or one relation provides limited, or even misleading, knowledge rather than the underlying truth. Considering the interplay between different relations with different dimensions involved, it is hard to derive the cause and effect by just looking at one specific relation. Taking Wikipedia as an example, controversy during the knowledge building process may originate from the in-fighting of aggressive contributors, or the controversial property of the topic covered in articles, etc,. It is obvious that considering all aspects that would bring about controversy is the only correct way to resolve it. Here, we demonstrate that by examining relations based on different perspectives, we can obtain a clear insight about the origin of controversy in Wikipedia.

The investigation of origin of controversies is important - because some conflicts in collaborative activities in general is inevitable - and because of the rich meta-information available openly in the case of Wikipedia, it provides an unique avenue to understand collaboration dynamics in online social collaborative environments in general. It is also specifically relevant for Wikipedia in that - while presence of articles which deal with

⁶http://en.wikipedia.org/wiki/Wikipedia:Long-term_abuse/Ananny

controversial content is natural, controversies may also arise simply because of discord among collaborating contributors. The later reflects poorly upon the overall collaboration environment as well as quality of content. In this section, we illustrate by combining the relevance information from different perspective, the essential factor regarding the controversy during the collaborative knowledge building process can be identified clearly.

3.3.1 Methodology

In group theory [71], controversy is defined as “the conflict that arises when one person’s ideas, information, conclusions, theories, and opinions are incompatible with those of another person, and the two seek to reach an agreement”. Given that Wikipedia is a collaboration system of knowledge building, controversy in Wikipedia will lead to the high volume of delete operations of someone else’s contributions⁷. This happens because people argue with each other by adding their opinions or revising others’ work. The fundamental question is thus “What is the root cause of such behavior?”. John Stuart Mill [101] once said “Since the general or prevailing opinion on any subjects is rarely or never the whole truth, it is only by the collision of adverse opinion that the remainder of the truth has any chance of being supplied”. Thus controversy could happen in the articles inherently containing some specific concept about which the contributors hold adverse opinions. Besides the issues of topics, contributors expose their social relations, such as prejudice, aggression, intimacy, through the interaction of their edit behaviors. Among multiple types of social relations identified in the study of social psychology [105], prejudice is the most relevant one with regards to controversy in the context of Wikipedia. Gordon Allport [18] defined prejudice in his book *The Nature of Prejudice* as “an antipathy based upon a faulty and inflexible generalization”. Consequently, it is possible that a group of contributors hold negative tendencies towards a class (a category) of topics,

⁷Here, we use the delete operation as a proxy for controversy in the context of Wikipedia. But, not every delete operation should be related to the conflict among the editors.

which inevitably causes argument. Since contributors are the main force driving the development of Wikipedia, contributors with emulative or aggressive personalities could also cause conflicts. Based on the above discussion and speculations in existing Wikipedia specific studies [22, 23, 77, 145], we identify and investigate three plausible hypothesis.

- The article deals with *specific controversial concepts* that are championed by different groups of users. For example, the article on Michael Jackson deals with specific controversial topics like child abuse, drugs, etc. In this example, the controversial concepts are sections of the article while the other sections of the article are not controversial.
- Alternatively, the article may belong to *a category of topics* that is generally controversial in nature. For example, all articles on nuclear technology and related scientific concepts may get controversial. Users have different understanding or points of views, which they may be championing across all associated articles.
- It is also possible that some aggressive contributors fight against each other, more because of *personality and egoistic reasons* rather than to do anything with the content of the articles themselves, and thus inadvertently make the articles look controversial.

We first determine the relevance relationships of controversial articles based on the above assumed causes. If the controversy is from the topical category of an article, then the relevant articles with similar content should attract the attention of the community too and will be controversial as well. Thus, controversial articles should be grouped in regards to the topical categories they deal with. Alternatively, if the aggressive contributors are the source of controversy, then controversial articles must share a lot of such

contributors. By measuring the article relevance from the commonality of its contributors, controversial articles should be much closer to each other. We can then cluster the controversial articles together by considering their contributors.

By clustering the controversial articles over different aspects, we expect to identify the common properties among them and confirm or discount the plausible source(s) of controversy. However, it is difficult to measure the relevance of controversial articles that contain some specific controversial topics (e.g., some specific sections in an article). Articles dealing with specific controversial topics should be less relevant to each other either in terms of content similarity or in terms of common contributors.

3.3.2 Controversial Articles

Using the method used to identify the controversial articles in [145], we build up a ground truth of 68 controversial articles by looking for the dispute tags assigned to the articles in their whole lifespan. We denote this corpus of controversial articles as CA⁸. There are 6 dispute indicative tags. Table 3.7 shows these tags and explains their meanings.

We first analyze the coordination and conflict of the contributors of these 68 articles. 5,203 contributors, excluding bots,⁹ had contributed to these. For identifying the disputes between contributors, we compare two successive revisions by counting the words in the old revision that were deleted in the new revision. It is likely that a contributor makes several successive revisions. In that case we consider only the last revision. Two contributors are said to have disputes in an article if one contributor has deleted some words from another’s contribution, or they both have deleted each other’s words. This simplistic and albeit somewhat flawed, nevertheless useful model to determine disputes was proposed in [14]. Results over the ground truth set are as follows:

⁸We also use CA as a short name for controversial article when the context is clear.

⁹The bots in Wikipedia are automated or semi-automated tools designed by contributors to carry out some edits, for example, adding some content and some links, reverting vandalism or removing some images, to a specific class of articles. Bots must be harmless and useful and be approved by Wikipedia.

- 4,285 contributors have edited exactly one controversial article, which is 82.4% of all the contributors. They can be downright discounted from being disputative.
- 15,444 contributor pairs have edited at least 2 common controversial articles. There are 917 unique contributors in these 15,444 contributor pairs.
- Among these 15,444 pairs only 81 contributor pairs (0.58%) comprising 71 unique contributors (0.46%) have disputes in at least 2 controversial articles.

The above results indicate that most contributors, approximately 99.54%, definitely do not stalk each other even if they had disagreement on some specific article and argued with each other, which would otherwise have led to more disputes in other articles. It also means that users don't intentionally form groups with any specific agenda and pursue such agenda across articles - even if such groups may form automatically in any given article, as has been witnessed previously [82].

So far we observed the disputes by counting the words deleted by contributors. Next, we zoom in on these 15444 contributor pairs by measuring edit wars among them. An edit war occurs when individual contributors or groups of contributors repeatedly override each other's contributions, rather than try to resolve the disagreement by discussion¹⁰. Here, we define an edit war as a pair of contributors who have deleted each other's contributions at least 3 times within an article. The following results are obtained from the analysis of edit wars among contributors:

- 104 contributors are involved in 93 edit wars within 29 articles.
- 2 contributor pairs have edit wars within 2 controversial articles, and 1 contributor pair has edit wars within 3 controversial articles.

¹⁰http://en.wikipedia.org/wiki/Wikipedia:Edit_war

Table 3.7: Dispute Tags Used

Tag	Meaning
<code>{{disputed}}</code>	The factual accuracy of this article is disputed.
<code>{{totallydisputed}}</code>	The neutrality and factual accuracy of this article are disputed.
<code>{{controversial}}</code>	This is a controversial topic, which may be under dispute.
<code>{{disputed-section}}</code>	Some section(s) has content whose accuracy or factual nature is in dispute.
<code>{{totallydisputed-section}}</code>	The neutrality and factual accuracy of some section are disputed.
<code>{{pov}}</code>	The neutrality of the article is disputed.

From the above results we note that while contributors have different opinions on some articles and argue with each other, even starting an edit war to repeatedly override each other’s contributions, still most of these contributors do not carry their fight to other articles. However, one cannot say that no contributors are aggressive or disputative. A total of 81 contributor pairs comprising 71 unique contributors have disputes in more than 1 article. This indicates that an aggressive contributor may choose different contributors to argue with across different articles. Also, the results of this section are restricted to a small corpus of 68 documents. We explore further the role of contributors next.

3.3.3 Role of Contributors

As described in Section 3.1.1, the expert-based similarity measure considers the common contributors shared by two articles. If there are more common contributors with more revisions, there will be higher expert-based similarity scores between such articles. Thus, the expert-based similarity measure opens an avenue for us to identify recurring disputes among arguing contributors.

In [145], a controversy model was built based on the assumption that the controversial contributors are the sources of disputes. If it is true, then controversial articles should have more shared controversial contributors with high contribution scores, leading to the very high expert-based similarity score for two controversial articles.

To determine whether contributors are the source of disputes, we first retrieve the top 30 relevant articles for each of the 68 CA articles by using the expert-based similarity and

aggregate them, including the original 68 CA articles, as a sub-dataset. The sub-dataset has 1696 unique articles. This is because some of these articles are close to at least two CA articles. We then use DBScan to cluster these articles.

If contributors are the cause of controversy, the controversial articles should have high expert-based similarity scores, so the CA articles should be clustered into the same cluster along with a large proportion of controversial articles in general. If this is true, then we can conclude that contributors are the cause of controversy. If not, we can likewise conclude that the disputes originate from the controversial nature of the topic that is specific to the article or the category to which the article belongs, rather than because of the contributors.

Since there is no obvious threshold point in the sorted k -dist graph, we set the percent of noise to 15%, 30%, 45%, 60% and 75% respectively. We then check how many clusters contain CA articles and the percentage of CA articles in the clusters. Figure 3.4(a) shows the number of clusters produced, the number of clusters containing the CA articles and the number of CA articles classified as noise w.r.t the five settings for the percent of noise. As illustrated, most of CA articles are classified as noise and only a small proportion of generated clusters contain CA articles.

When the percent of noise is 30% or 15%, more than half of the CA articles are put into the clusters. It looks like controversial articles are closer to each other. But as the number of generated clusters decreases, each cluster becomes larger and contains a large number of articles. Thus, the percentage of the CA articles, instead absolute numbers in their cluster can help us figure out the real distribution of CA articles. If some cluster contains almost only CA articles and a major proportion of CA articles, then we could say that the controversial articles have higher expert-based similarity scores between themselves. It would then indirectly confirm that the source of controversy is controversial contributors.

Feature	Percentage of Noise				
	75%	60%	45%	30%	15%
#Cluster	24	37	29	16	9
#Cluster with CA	7	14	14	8	5
#CA as Noise	59	50	36	24	15

(a) Statistic about clustering result w.r.t the different parameter settings. #Cluster: the number of clusters produced; #Cluster with CA: the number of clusters containing the CA articles; #CA as Noise: the number of CA articles classified as noise.

(b) Number of controversial articles vs. Percentage of controversial articles in their clusters

Figure 3.4: Distribution of CAs using expert-based similarity

Figure 3.4(b) illustrates the percentages of the CA articles in their clusters w.r.t the absolute number of CA articles each cluster contains. We note that there are two kinds of clusters: 1. clusters with small number of CA articles (less than 10); 2. clusters with a large number of CA articles (more than 10). The relatively low percentage of CA articles in the latter type of clusters indicates that the size of cluster increases along with the number of CA articles it contains. By observing Figures 3.4(a) and 3.4(b), we conclude that the controversial articles are not highly related to each other. In other words, their

mutual expert-based similarity is not higher than that to other articles in the dataset. Based on the property of the expert-based similarity, we can say that these controversial articles don't have a large number of shared contributors, even when they have a large number of revisions. Thus, one can refute that the controversial contributors are the source of disputes.

Thus it must be the article itself that contains some specific controversial subject matter or otherwise belong to a category involving general controversial concepts. For example, the article "Michael Jackson"¹¹ is an example that invites the disputes by its specific controversial matter contained in the article. People argued with each other about his changing appearance, the child sexual abuse, his marriages and even his death. However, we can't find such similar conflicts in other articles about dance musicians. On the other hand, there are some debates in the article "Nuclear power"¹² about its pollution, radioactive waste and safety issues. It is possible that similar disputes occur in articles related to nuclear power, for example, in nuclear reaction, nuclear power stations and nuclear entombment, etc. Next, we investigate disputes from the concept perspective.

3.3.4 Concept Perspective

Let's assume that the article deals with some general controversial concepts - which would recur in other articles dealing with the same concept. Then relevant articles under the same concept should attract the attention of the community too. Thus, these articles could be clustered together as they are semantically relevant in terms of their content. As discussed in Section 3.2, the semantic relevance can also be derived from the link structure of Wikipedia. But considering the mediocre performance of the P-Rank and SimRank similarity measures obtained in Section 3.2, we only use the cosine similarity

¹¹http://en.wikipedia.org/wiki/Michael_Jackson

¹²http://en.wikipedia.org/wiki/Nuclear_power

here and repeat the experiments conducted above in Section 3.3.3. We build the sub-dataset by retrieving the top 30 relevant articles for each CA article by using the cosine similarity. There are 1,589 unique articles, including the original 68 articles. The results are illustrated in Figure 3.5(a) and Figure 3.5(b). Similar to our previous experiments exploring the role of contributors, we note that most of the clusters contain only one or two controversial articles. And the percentage of CA articles in clusters is very low (less than 20%), especially in the clusters containing a large number of CA articles. We can say that the controversial articles don't have high relevance with each other in their semantic content, which means that the controversial concept is not a principal source of controversy either.

Having eliminated the other possibilities, we infer that specific controversial topics contained in articles are the primary source of controversy in Wikipedia. This conclusion is specific to the "Religious objects" category, but our methodology can be emulated for any other data-set.

3.4 Conclusions

Over the last decades, an increasing amount of our daily life and business is being carried out online using digital technologies, with the advent of Web 2.0 and online social networking sites. The interactions among users and information items offer us a new avenue to discover information that complement with those from other approaches. Thus, rich information regarding different kinds of relations may be obtained by considering relations mined from different perspectives, which offer interpretations not always available when investigating individual facets in isolation. In this chapter, we examine different relations from different perspectives in the context of Wikipedia by studying the multiple relations induced among contributors and articles based on edit history, link structure, contributors' expertise, etc.

Besides the link structure and content information associated with Wikipedia articles, we propose a new similarity measure, named *expert-based similarity*. Experiments show that the expert-based similarity is an effective and efficient similarity measure to measure the relevance of articles.

Moreover, as a case study, we studied the source of controversy from different aspects of Wikipedia, including the contributors' edit history, general controversial subject which an article belongs to, as well as topics specific to individual articles. By leveraging different dimensions, i.e., the content, the semantic link structure as well as the editorship of Wikipedia articles, we find that, while isolated edit-wars and group formations have been witnessed in previous studies, most contributors however don't continue such antagonisms across articles (i.e., there is no stalking effect) - and we conclude that disagreements among contributors is nothing personal. We similarly find that general topics and concepts do not cause controversies. Thus, we conclude that controversies arise from specific content typically confined to individual articles themselves.

This conclusion on 'origin of controversies' is valid specifically for the articles in "Religious Objects" category of Wikipedia, but our methodology to infer the conclusion is generic.

In the context of the bigger picture of online social media mining and analysis, the work presented in this chapter is a tiny albeit important step in systematically demonstrating the possibility of using different aspects of relations to determine new information, which, without the use of a specific relation aspect may simply not be derivable, or worse still, may be misleading. Similar conclusion is also made in the study of word sense disambiguation based on Wikipedia (Chapter 4), where we observe that a specific similarity measure could lead to a misleading decision. Following this guideline, we consider multiple relations of different perspectives in the follow-up works. In Chapter 5, we show that a combination of tag and document perspectives results in a better performance in

tag recommendation. In Chapter 7, we propose an event detection system that considers both the segment’s bursty pattern and associated user support together to fight against noisy tweets, such as tweets from the categories of self-promotion, spam and pointless babble.

Feature	Percentage of Noise				
	75%	60%	45%	30%	15%
Cluster	22	26	16	5	1
Cluster with CA	10	24	11	4	1
CA as Noise	52	39	23	16	6

(a) Statistic about clustering result w.r.t the different parameter settings. #Cluster: the number of clusters produced; #Cluster with CA: the number of clusters containing the CA articles; #CA as Noise: the number of CA articles classified as noise.

(b) Number of controversial articles vs. Percentage of controversial articles in their clusters

Figure 3.5: Distribution of CAs using cosine similarity

Chapter 4

Word Sense Disambiguation Using Wikipedia*

Last night I shot an elephant in my pajamas. How he got in my pajamas, I'll never know.

– Groucho Marx

4.1 Introduction

In Chapter 3, the article similarity measures based on different relations in the context of Wikipedia are studied intensively. As being a multi-dimensional knowledge building system, Wikipedia has become a credible resource in many research fields. Particularly, Wikipedia's categorization scheme and hyperlink (wikilink) have been extensively studied in text mining, such as document clustering and classification [43, 147, 64, 91], semantic relatedness [132, 44, 102, 162], topic detection and indexing [98, 49] and named entity disambiguation [25, 30, 53], etc. That is, the traditional IR/NLP techniques based on bag-of-word model are complemented with the semantic resources offered by Wikipedia. Besides doing feature expansion, we can conduct wikification for the text documents to

*This chapter is based on the paper *TSDW: Two-Stage Word Sense Disambiguation using Wikipedia* by Chenliang Li, Aixin Sun, Anwitaman Datta, published in *Journal of the American Society for Information Science and Technology* 64(6), 2013.

associate them to the dimensions of the concept space defined by Wikipedia. Wikification is the task which links words or phrases in text documents to their corresponding Wikipedia topics [100]. In contrast to traditional BOW model, representing text documents with Wikipedia topics could retain many useful information, such as polysemy, synonym, abbreviations, etc.

In this chapter, we are interested in the problem of *Word Sense Disambiguation to Wikipedia*, or simply *Disambiguation to Wikipedia (D2W)*. Formally defined in [119], the task of *D2W* is to disambiguate a set of explicitly identified substrings (*e.g.*, words or phrases) in a given document by mapping each substring to a Wikipedia page¹, if there exists one. For simplicity, we refer the explicitly identified substrings as phrases² and we only focus on the phrases that each has at least one corresponding Wikipedia page. If a phrase maps to exactly one Wikipedia page, the mapping process is straightforward because the phrase is unambiguous. A phrase is ambiguous if it can be mapped to more than one Wikipedia page. The task of *D2W* is to disambiguate the *ambiguous phrases* in a given document to their correct Wikipedia topics based on the content of the document. The *D2W* problem is the fundamental problem of the Wikification task. *D2W* is also similar to the word sense disambiguation task in natural language processing, which can be an important preprocessing step in the aforementioned text mining tasks. Figure 4.1 shows an example of an article in our evaluation dataset, where ambiguous phrases are highlighted in red and boldface, and disambiguated using our proposed approach.

Recently, many works have been proposed to address the problem of *D2W* [98, 103, 119, 85]. The proposed solutions mainly focus on the approximation of the likelihood of a phrase mapping to each candidate Wikipedia page based on the semantic context of the document. The semantic context is defined to be the set of Wikipedia pages that are

¹Because each Wikipedia article introduces a single topic, in our discussion, we use Wikipedia article, Wikipedia page, candidate sense, sense, candidate topic, Wikipedia topic equivalent interchangeably.

²We do not specifically distinguish words and phrases in our discussion.

Figure 4.1: A sample article with phrases disambiguated to Wikipedia topics. Ambiguous phrases are highlighted in red and boldface. The top 10 unambiguous phrases in terms of *keyphraseness* are highlighted in gradual changing green colors. The top 5 unambiguous phrases are highlighted in boldface with their corresponding Wikipedia topics labeled. The most probable candidate topics are listed for the ambiguous phrases along with the corresponding *commonness* values and their correct Wikipedia topics are highlighted in boldface.

uniquely mapped to by the unambiguous phrases in the document [98, 103, 85]. However, the information provided by the unambiguous phrases in a document could be very limited, leading to poor accuracy for disambiguation of some ambiguous phrases. The situation becomes apparent when short documents are processed. To augment the semantic context of such documents, Wikipedia pages of ambiguous phrases disambiguated by some simple methods have been utilized as additional context information. One simple method is to map an ambiguous phrase to the Wikipedia page that has the largest cosine similarity with the local neighboring words of the ambiguous phrase [119]. However, the augmentation may bring in not only additional computational cost but also noisy information. In addition, all existing works are based on English articles only. Considering the popular usage of Wikipedia in other languages as well as its significant contribution in many fields [165, 74, 86, 28, 129, 137], it is interesting to study the performance of the disambiguation solutions to other languages in terms of both effectiveness and efficiency.

In this chapter, we propose a generic two-stage framework for word sense disambiguation to Wikipedia, named *TSDW* (*i.e.*, **T**wo-**S**tage **D**isambiguation to **W**ikipedia). *TSDW* consists of three key components: *Wikipedia inventory*, *keyphrase recognizer* and *two-stage disambiguator*. We build a word sense inventory by extracting the polysemy, synonym and hyperlinks encoded in Wikipedia. Each entry in the inventory is a keyphrase which refers to at least one Wikipedia article. A keyphrase is a phrase which is used either as a Wikipedia article title, or anchor text of a wikilink in Wikipedia. The keyphrases, each of which refers to exactly one Wikipedia article, are unambiguous keyphrases. Some keyphrases are ambiguous, each of which refers to multiple Wikipedia articles (*i.e.*, candidate topics/senses).

Given a document, the unambiguous keyphrases extracted by the keyphrase recognizer from the document serve as context information to help in disambiguating the ambiguous keyphrases. While the ambiguous keyphrases are often ignored in existing works, some of them may provide additional semantic clue, resulting in a better semantic context for disambiguation. The core component of *TSDW*, which distinguishes this work from our previous work [85], is the two-stage disambiguator. In the first stage, it disambiguates the ambiguous keyphrases in a document by exploring semantic context defined by the unambiguous keyphrases. The quality of each disambiguation decision in the first stage is evaluated by a confidence measure. In the second stage, the disambiguated keyphrases with high confidence from the first stage are recruited as additional semantic context for a better disambiguation of the keyphrases with low confidence. One of the main contribution of this research is therefore the confidence measure.

Moreover, the recruitment of high-confident disambiguated keyphrases alleviates the data sparsity problem in the first stage. For example, as demonstrated in Figure 4.1, the ambiguous keyphrase *Germany* cannot be disambiguated with high confidence, because

the context of the 5 unambiguous keyphrases³ cannot provide adequate discriminative information. However, the ambiguous keyphrases *World Cup* and *1998 World Cup* can be easily disambiguated correctly with high confidence at the first stage. Based on the additional semantic information offered by these two disambiguations, *Germany* can be correctly disambiguated to Wikipedia topic *Germany national football team* in the second stage. Note that, not all unambiguous or high-confident disambiguated keyphrases are used as semantic context in our disambiguation process because of both effectiveness and efficiency reasons. We will further illustrate this point by showing the impact of the size of the semantic context in our experiments.

We highlight that the proposed *TSDW* framework is generic because it can be materialized by using any semantic relatedness measure. In our experiments, we evaluate the impact of using three semantic relatedness measures including *Jaccard*, *Dice*, and Wikipedia Link-based Measure (*WLM*). Note that semantic relatedness measures hold varying characteristics for Wikipedia corpora in different languages. In our experiments, we have evaluated our *TSDW* framework using both English and Chinese versions of Wikipedia. More specifically, our empirical evaluation involves several datasets ranging from Wikipedia articles to newswire reports, and in two different languages, English and Traditional Chinese. Our experimental results show that *TSDW* outperforms other state-of-the-art approaches in terms of both effectiveness and efficiency, for both English and Traditional Chinese articles. In next section, we present each component of *TSDW* in detail.

4.2 *TSDW* Disambiguation Framework

In this section, we provide concrete description of the three core components of *TSDW*: *Wikipedia inventory*, *keyphrase recognizer*, and *two-stage disambiguator*, in that sequence

³The 5 unambiguous keyphrases with the highest *keyphraseness* values: *Bulgaria national football team*, *Vasil Levski National Stadium*, *Chelsea F.C.*, *FC Levski Sofia*, *Golden Generation*.

respectively, following the order of their usage in our framework.

4.2.1 Wikipedia Inventory

The Wikipedia inventory is a dictionary that consists of keyphrases and their associated candidate topics based on Wikipedia. Each candidate topic refers to a corresponding Wikipedia article. The sources for the keyphrases and associated candidate topics include: Wikipedia article titles, anchor text of wikilinks, redirect pages and disambiguation pages in Wikipedia. In the following, we describe each of the four sources in detail:

- Given a topic of a Wikipedia article, it is represented by the name that is the most commonly used to refer to that topic⁴. Hence, the titles of Wikipedia articles are included in our Wikipedia inventory as keyphrases, each of which contains the associated Wikipedia article as its candidate topic. Note that Wikipedia pages for administration or maintenance purposes (e.g., discussion, talk, user pages) are excluded.
- Based on Wikipedia policy, wikilinks (or hyperlinks) in Wikipedia are created to help readers better understanding the topic, by linking technical terms and proper names (i.e. abbreviations, synonym, spelling variations) to the related Wikipedia articles⁵. Thus, including the anchor text and the related Wikipedia articles largely enriches the keyphrase inventory, leading to an inventory of broader coverage.
- A redirect page groups the alternative names of a topic together by establishing redirect relations between the alternative names and the Wikipedia article of that topic. For instance, "U.S." is redirected to the Wikipedia article *United States*, because "U.S." is an alternative name for the topic *United States*. Such redirections

⁴<http://en.wikipedia.org/wiki/Wikipedia:TITLE>

⁵<http://en.wikipedia.org/wiki/Wikipedia:Linking>

help us further enrich the keyphrase inventory with abbreviations, synonym, and spelling variations.

- Wikipedia disambiguation pages are designed to help readers finding the topic of interest among possible candidate topics from the polysemy query. The titles of such pages are normally polysemy keyphrases, followed by tag **disambiguation**. The candidate topics are listed in the page, each with a short description about it. We adopt the heuristic by Turdakov and Velikhov [142] to extract the candidate topics from each disambiguation page. When an ambiguous term already exists in the inventory as a keyphrase, we update its list of candidate topics with the ones extracted from corresponding disambiguation page.

In summary, Wikipedia keyphrase inventory is created by taking Wikipedia article titles, processing redirected pages, parsing disambiguation pages and extracting hyperlinks. In the inventory, if a keyphrase is associated with exactly one topic (or article), we call it unambiguous keyphrase. An ambiguous keyphrase is associated with more than one topic.

4.2.2 Keyphrase Recognizer

Given an input document, all keyphrases that appear in the Wikipedia inventory are extracted from the document with the preference of longer phrases. For example, given a sentence segment, “**java programming language**”, we extract keyphrase **java programming language** instead of two keyphrases *java* and *programming language*. The keyphrases extracted are classified into ambiguous keyphrases and unambiguous keyphrases based on the Wikipedia inventory. For the unambiguous keyphrases extracted, their associated Wikipedia topics are obtained directly from the inventory. These Wikipedia topics provide us with the semantic clue to the topics covered in the document, and thus help us disambiguate the ambiguous keyphrases.

Given Wikipedia’s broad coverage of human knowledge, the size of the Wikipedia inventory is extraordinarily large, with millions of keyphrases. Thus, recognizing matched keyphrases efficiently based on the Wikipedia inventory becomes a non-trivial task. We implement the keyphrase recognizer by using prefix tree [79] algorithm over the keyphrases of the same prefix word. This algorithm has a complexity of $O(n)$, where n is the length of the input document in number of words. The detailed description of the algorithm is provided in Appendix A.

4.2.3 Two-Stage Disambiguator

We disambiguate an ambiguous keyphrase based on the semantic context it is associated with. Given an ambiguous keyphrase k and its associated context $\mathcal{C}$, we compute the probability of keyphrase k referring to a candidate topic t , denoted by $P(t|k, \mathcal{C})$. The candidate topic with the highest probability is chosen to be the disambiguated sense of the keyphrase, shown in the following equation, where T_k denotes the set of candidate topics for keyphrase k :

$$t_o = \arg \max_{t \in T_k} P(t|k, \mathcal{C}) \quad (4.1)$$

To compute $P(t|k, \mathcal{C})$, we mimic human being’s disambiguation process by considering two factors: the prior probability of keyphrase k referring to a candidate topic t , also known as *commonness* (Section 2.2.1), and the likelihood of context $\mathcal{C}$ related to the candidate topic. Assuming that the two factors are independent from each other and are independent for a given topic t , we have Equation 4.2 to compute $P(t|k, \mathcal{C})$. In this equation, $P(t|k)$ is the prior probability of topic t given keyphrase k (*i.e.*, *commonness*),

and $P(t|\mathcal{C})$ is the probability of the keyphrase referring to topic t given context $\mathcal{C}$:

$$\begin{aligned}
 P(t|k, \mathcal{C}) &= \frac{P(k, \mathcal{C}|t)P(t)}{P(k, \mathcal{C})} \\
 &= \frac{P(k|t)P(\mathcal{C}|t)P(t)}{P(k, \mathcal{C})} \\
 &= \frac{P(k, t)P(\mathcal{C}, t)}{P(t)P(k, \mathcal{C})} \\
 &= \frac{P(k, t)P(\mathcal{C}, t)}{P(t)P(k)P(\mathcal{C})} \\
 &= \frac{P(t|k)P(t|\mathcal{C})}{P(t)} \tag{4.2}
 \end{aligned}$$

In Equation 4.2, $P(t)$ is the prior probability of topic t . We further assume a flat prior probability for all topics, *i.e.*, $P(t_i) = P(t_j)$. Thus, Equation 4.2 can then be updated as follows:

$$P(t|k, \mathcal{C}) \propto P(t|k)P(t|\mathcal{C}) \tag{4.3}$$

Because $P(t|k)$ is independent of the context and can be estimated from the Wikipedia data directly, the task of disambiguation is then reduced to estimating $P(t|\mathcal{C})$. As discussed in related work, the context is usually defined by the set of unambiguous keyphrases in the given document. However, a document may cover many diverse topics. Therefore, not all unambiguous keyphrases from the document are equally important in defining the context for a keyphrase to be disambiguated. While the more related keyphrases help identify the correct sense of an ambiguous keyphrase, the less related ones may hurt the disambiguation accuracy and incur additional computation. The situation would be further exaggerated by the noise contained in Wikipedia. This calls for an appropriate construction of context $\mathcal{C}$ for an ambiguous keyphrase, aiming for both disambiguation effectiveness and efficiency. On the other hand, the discriminative information carried by the related unambiguous keyphrases may be limited. For example, a short text may contain a very small number of unambiguous keyphrases; or the

Figure 4.2: The two-stage disambiguation process

unambiguous keyphrases only cover a sub-topic of the text and are not useful for the ambiguous keyphrases about other sub-topics. This calls for an approach to enlarging the context for disambiguation by adding additional information on top of the existing unambiguous keyphrases extracted from the document.

To answer the aforementioned two requirements, we propose a *two-stage disambiguator* to exploit the semantic clue provided by both the unambiguous keyphrases and the ambiguous keyphrases. Illustrated in Figure 4.2, in the first stage, the disambiguator disambiguates the ambiguous keyphrases based on the context formed by unambiguous keyphrases after pruning the less related unambiguous keyphrases. The small number of high quality unambiguous keyphrases as context address the first requirement. Based on the confidence measure, each disambiguation is classified as disambiguation with high- or low-confidence. In the second stage, the keyphrases with high confident disambiguation is added into the context for the re-disambiguation of the keyphrases with low confidence in the first stage by re-estimating $P(t|\mathcal{C})$. The use of additional context originated from the ambiguous keyphrases answers the second requirement. We detail the two-stage disambiguator in the following. Note that, the second stage disambiguation process can be easily repeated for more than one pass leading to a third stage (or more) re-disambiguation. However, we observe in our experiments that more iterations do not necessarily lead to improvement in the disambiguation accuracy.

4.2.3.1 First-Stage Disambiguation

In the first stage, we try to approximate the likelihood $P(t|\mathcal{C})$ by restricting context $\mathcal{C}$ to be a subset of all unambiguous keyphrases extracted from the input text. Given all unambiguous keyphrases extracted from the input text, not all of them are equally helpful for word sense disambiguation. For instance, the keyphrase *Mr.* is a keyphrase in the Wikipedia inventory. However, it is unlikely that the keyphrase would contribute positively to the estimation of $P(t|\mathcal{C})$ for any topic t . We therefore construct context $\mathcal{C}$ by applying keyphrase pruning, and approximate $P(t|\mathcal{C})$ by using weighted relatedness to $\mathcal{C}$.

We use *keyphraseness* measure to quantify the importance of a keyphrase. For a given unambiguous keyphrase, *keyphraseness* is the a prior probability that a keyphrase is selected as a link, no matter where it appears in Wikipedia. For example, keyphrase *Conference halls* highlighted in Figure 4.1 has a very low *keyphraseness* value, because it is rarely linked by other Wikipedia articles or used as anchor text for wikilinks. Based on this measure, we select the top M_1 keyphrases with the highest *keyphraseness* values to form context $\mathcal{C}$. The keyphrases in $\mathcal{C}$ are known as *context keyphrases* and $|\mathcal{C}| \leq M_1$.

Recall that a document may cover many diverse topics, which is often reflected by its M_1 context phrases. That is, some context phrases from M_1 may not be strongly related to other context phrases. A context keyphrase is weighted by its relatedness to all other context keyphrases, shown in the following equation:

$$w(k, \mathcal{C}) = \frac{\sum_{k' \in \mathcal{C} \setminus k} r(k, k')}{|\mathcal{C}| - 1} \quad (4.4)$$

where $r(k, k')$ denotes the relatedness between two Wikipedia articles referred to by k and k' respectively. Because each keyphrase (or one of its candidate topics) refers to one Wikipedia article, the relatedness measure is therefore reduced to the problem of computing the relatedness between their associated Wikipedia articles. A few measures

have been reported in the literature to measure the semantic relatedness between two Wikipedia articles, and cosine similarity based on the content of the articles [119]. As a generic framework, *TSDW* can use any such measure. In the following discussion we use $r(k, k')$ to denote the relatedness between two keyphrases k and k' (or candidate topic t) computed by any chosen semantic relatedness measure.

With a chosen relatedness measure, the weighted relatedness $r(t, \mathcal{C})$ between a candidate topic t to context $\mathcal{C}$ is computed using Equation 4.5. The likelihood $P(t|\mathcal{C})$ is approximated by using $r(t, \mathcal{C})$ with an exponential scale-up factor c as in Equation 4.6. Note that the scale-up factor c is specific to the relatedness measure under use, and its value can be learned by applying grid search over the range of $[0, 10]$ [85].

$$r(t, \mathcal{C}) = \frac{\sum_{k \in \mathcal{C}} w(k, \mathcal{C}) \times r(t, k)}{\sum_{k \in \mathcal{C}} w(k, \mathcal{C})} \quad (4.5)$$

$$P(t|\mathcal{C}) \approx r(t, \mathcal{C})^c \quad (4.6)$$

Given a keyphrase k to be disambiguated, Equation 4.3 can be updated as follows:

$$P(t|k, \mathcal{C}) \propto P(t|k) \times r(t, \mathcal{C})^c \quad (4.7)$$

Thus, the topic t_o is chosen as the disambiguated topic to k based on the following Equation:

$$t_o = \arg \max_{t \in T_k} P(t|k) \times r(t, \mathcal{C})^c \quad (4.8)$$

Through the first stage disambiguation, using Equation 4.8, the topic with the highest approximated probability is chosen as the disambiguated sense for an ambiguous keyphrase. We then predict the quality of each disambiguation decision by using a loss function defined in Equation 4.9.

$$\mathcal{L}(t_o, k, \mathcal{C}) = \sum_{t \in T_k \setminus t_o} P(t|k, \mathcal{C}) \propto \sum_{t \in T_k \setminus t_o} P(t|k) \times r(t, \mathcal{C})^c \quad (4.9)$$

We observe that the top 2 candidate topics of the highest probabilities often dominate the probability space. Thus, given topic t_2 being the second highest likelihood, we approximate the loss function by using $P(t_2|k, \mathcal{C})$. While the loss function (Equation 4.9) is defined as a scalar of scale-free, specific to each ambiguous keyphrase, we define the confidence for each disambiguation decision by standardizing the loss function with respect to $P(t_o|k, \mathcal{C})$:

$$cfd(t_o, k, \mathcal{C}) = \frac{P(t_o|k, \mathcal{C}) - P(t_2|k, \mathcal{C})}{P(t_o|k, \mathcal{C})} \quad (4.10)$$

Observe in Equation 4.10, confidence measure $cfd(t_o, k, \mathcal{C})$ is the relative difference between the top 2 candidate topics and has a range of $[0, 1]$, from the smallest confidence to the largest confidence. It is reasonable because a very low confidence is indicative of a random guess among the two candidate topics with very close approximations of $P(t|k, \mathcal{C})$. Note that, measuring decision confidence using the gap between the two best decisions is a strategy that has been applied in speech recognition tasks [60, 154].

To summarize, the first stage disambiguation involves two parameters: M_1 for the size of the context, and c for the probability approximation. A smaller M_1 keeps most useful topics for disambiguation and improves the efficiency, with the risk of filtering away helpful topics as well. A larger M_1 , on the other hand, may bring in more useful topics as well as noise, and certainly increases computation. As for the scaling factor c , it gives the flexibility of adjusting the impact of relatedness measure based on various relatedness definitions (e.g., *Jaccard* and *WLM*).

4.2.3.2 Second-Stage Disambiguation

By applying the confidence measure, we classify the disambiguated keyphrases from the first stage as either high-confident keyphrases or low-confident keyphrases by using a pre-determined threshold θ . We believe that the high-confident keyphrases would provide helpful information to better disambiguate the keyphrases of low confidence. Thus, in

the second stage, we update context $\mathcal{C}$ with the topics referred to by the high confident keyphrases. The ambiguous keyphrases of low confidence are disambiguated again based on the updated context. This is reasonable because the low-confident disambiguations indicates that the context we utilized in the first stage may not provide adequate discriminative information for these difficult ambiguous keyphrases. Specifically, we add M_2 topics from the high-confident keyphrases as additional context keyphrases. The resulting new set of context keyphrases is denoted by $\mathcal{C}_2$.

The new M_2 topics used in the second stage are selected from the high confident keyphrases by leveraging the semantic clue provided by the exiting context keyphrases used in the first stage. More specifically, we assign a score to each disambiguated keyphrase as follows:

$$\begin{aligned} score(t_o, k, \mathcal{C}) &= cfd(t_o, k, \mathcal{C}) \times P(t_o | \mathcal{C}) \\ &\approx cfd(t_o, k, \mathcal{C}) \times r(t_o, \mathcal{C})^c \end{aligned} \quad (4.11)$$

Observe that the disambiguated keyphrase (i) that has higher confidence and, (ii) is highly related to the existing context is assigned a higher score. Since each disambiguated keyphrase refers to a specific Wikipedia topic, we assume that picking topics with high relatedness to the existing context $\mathcal{C}$ would provide more discriminative information. For low confident keyphrases, we re-disambiguate them using the new context $\mathcal{C}_2$:

$$t_o = \arg \max_{t \in T_k} P(t | k) \times r(t, \mathcal{C}_2)^c \quad (4.12)$$

The parameter M_2 is similar to the parameter M_1 in the first stage to balance the efficiency and effectiveness. That is, a small M_2 may not provide enough discriminative information; a large M_2 brings in not only more useful information for disambiguation but also noise and extra computation. Thus, *TSDW* involves four parameters: M_1 for the context size in the first stage, c for the relatedness measure scaling, confidence threshold θ , and M_2 for the size of additional context keyphrases in the second stage.

The existing works [98, 103, 85, 119] can be considered as specific cases of *TSDW* without the second stage disambiguation (*i.e.*, $M_2 = 0$). In the first stage of disambiguation, these works differ in the way to approximate $P(t|k, \mathcal{C})$. Medelyan et al. tried to approximate $P(t|\mathcal{C})$ by considering all unambiguous keyphrases equally. Milen and Witten improved the approximation by weighing all unambiguous keyphrases based on their relatedness to each other. The latter also tried to approximate $P(t|k, \mathcal{C})$ indirectly by using machine learning techniques. The approach proposed by Ratinov et al. is hard to be compared directly with, because they tried to obtain a better approximation of $P(t|\mathcal{C})$ by combining the relatedness measures based on both local and global information in a supervised learning manner. Our previous approach [85] constitutes the first stage disambiguation only. Note that only the low-confident keyphrases are re-disambiguated in the second stage with the updated context. In the next section, we empirically study the performance of *TSDW* and other state-of-the-art approaches in terms of both effectiveness and efficiency.

4.3 Experiments

In this section, we conduct extensive experiments to evaluate the performance of *TSDW*. To demonstrate that *TSDW* is generic in accommodating different relatedness measures and Wikipedia in different languages, we evaluate *TSDW* using three relatedness measures on two versions of Wikipedia (English and Traditional Chinese). After detailing the *TSDW* setup and performance metric, we report the performance comparison between *TSDW* and existing methods. Lastly, we report a performance analysis of *TSDW* in different settings.

4.3.1 TSDW Setup and Performance Metric

Wikipedia Inventory. We build two Wikipedia inventories from the English and Traditional Chinese Wikipedia dumps respectively. Specifically, we used the English Wikipedia dump released on 30 January, 2010⁶. There are 3,246,821 articles and 266,625,017 hyperlinks among them, excluding all redirection pages. The built Wikipedia keyphrase inventory consists of 6,168,269 unambiguous keyphrases and 526,081 ambiguous keyphrases respectively. For the latter, each keyphrase refers to 4.22 candidate topics on average. The Traditional Chinese Wikipedia dump released on 28 June, 2011⁷ are used to build the Chinese version of keyphrase inventory. There are 355,245 articles and 24,720,728 hyperlinks among them, excluding all redirection pages. The built Wikipedia keyphrase inventory consists of 815,303 unambiguous keyphrases and 40,895 ambiguous keyphrases respectively. Each ambiguous keyphrase refers to 3.0 candidate topics on average.

Relatedness measure. Three relatedness measures, namely *Dice* [33], *Jaccard* [67] and *WLM* [102] are investigated in *TSDW* for validating the general applicability of the framework. All the three measures compute the relatedness between two Wikipedia articles by considering their in-coming wikilinks:

$$\begin{aligned} r_{WLM}(a, b) &= \frac{\log(\max(|A|, |B|)) - \log(|A \cap B|)}{\log(W) - \log(\min(|A|, |B|))} \\ r_{Jaccard}(a, b) &= \frac{|A \cap B|}{|A \cup B|} \\ r_{Dice}(a, b) &= \frac{2|A \cap B|}{|A| + |B|} \end{aligned}$$

where a and b are two Wikipedia articles, A and B are the sets of Wikipedia articles that link to a and b respectively, and W is the number of articles in Wikipedia. While *Dice* and *Jaccard* are general similarity measures over sets, *WLM* is a Wikipedia-specific

⁶<http://download.wikimedia.org/enwiki/20100130/>.

⁷<http://dumps.wikimedia.org/zhwiki/20110628/>.

similarity measure based on *Normalized Google Distance* [29]. In contrast to *Dice* and *Jaccard*, *WLM* explicitly considers the issue that the two sets under consideration would have very unbalanced cardinalities. *WLM* has been widely adopted in the related works that exploit the semantic resources of Wikipedia [103, 53, 85, 119, 126, 66].

Performance Metric. In our experiments, for each ambiguous keyphrase k to be disambiguated, exactly one candidate topic t is assigned to k by a disambiguation method to be evaluated. We report the *accuracy* of the assignments, i.e., the ratio of the correct assignments for all ambiguous keyphrases involved in the evaluation. The correct assignments are determined by manual verification in our experiments. Note that as each ambiguous keyphrase cannot have more than one sense in a given context, the accuracy reported here is the same as precision or recall.

The performance of efficiency is also reported in the experiments. All experiments are conducted on the same workstation with a 2.40GHz Xeon quad-core CPU and 24GB of RAM. The execution time by each method is the time taken for keyphrase recognition and keyphrase disambiguation ignoring the time taken for data loading or classifier training.

4.3.2 Comparison with Other Methods

In this set of experiments, we compare our method with four state-of-the-art methods and two baseline methods for both effectiveness and efficiency. In specific, we compare our method with the methods reported in Milne and Witten (M&W) [103], Medelyan et al. [98], Ratnov et al. [119] as well as our previous proposed approach [85]. The first builds machine learning classifiers to disambiguate the keyphrases; the second maximizes the balance between commonness and relatedness using equal weight; the third combines both the local word windows and link structures for disambiguation; and the last is equivalent to the first stage disambiguation (i.e. $M_2 = 0$ in *TSDW*). For the first method, we build two classifiers C4.5 and Bagged C4.5 using Weka library [50]. The two baseline methods

Table 4.1: Statistics on datasets. *en* and *zh* refers to English and Traditional Chinese respectively. #articles: the number of Wikipedia articles; #word: the average number of words per Wikipedia article; #unambiguous: the average number of unambiguous keyphrases per Wikipedia article; #ambiguous: the average number of ambiguous keyphrases per Wikipedia article; #candidates: the average number of candidate topics per ambiguous keyphrase.

Dataset	#articles	#word	#unambiguous	#ambiguous	#candidates
<i>Training_{en}</i>	500	9,759	103.5	30.9	45.7
<i>Validation_{en}</i>	100	11,294	117.0	38.4	46.3
<i>Evaluation_{en}</i>	200	9,647	108.2	37.9	46.8
<i>Training_{zh}</i>	500	4,453	260.6	24.9	10.5
<i>Validation_{zh}</i>	100	3,837	229.1	24.1	12.5
<i>Evaluation_{zh}</i>	200	3,969	237.4	21.0	11.3

are *Random sense* and *Most common sense* which simply assign topics to ambiguous keyphrases randomly and to the most common sense respectively.

Note that Ratnov et al. published the implementation⁸ of their system (called *Illinois Wikifier*) along with the four datasets used in their work [119]. Because their system is implemented based on the Wikipedia dump of 2009, we compare *Illinois Wikifier* and *TSDW* on the four datasets used in their work separately. Given that *Illinois Wikifier* requires NER tagger and shallow parser, we do not compare it with *TSDW* on Traditional Chinese articles.

For the comparison with the other methods, we prepare two datasets of 800 articles each, sampled from English and Traditional Chinese Wikipedia respectively. In each dataset, these 800 articles are randomly split into three non-overlapping subsets of 500, 100 and 200 articles. The subset of 500 articles is used for classifier training. The trained classifiers are validated using the subset of 100 articles. For a fair comparison, all methods are evaluated on the subset of 200 articles of each dataset. The statistics of the two datasets with their subsets are reported in Table 4.1.

We first report the evaluation results on English dataset. We use the following parameter settings for *TSDW*: $c = 1.5$, 1.5 and 6.0 for *Dice*, *Jaccard* and *WLM* respectively,

⁸http://cogcomp.cs.illinois.edu/page/download_view/Wikifier

Table 4.2: Disambiguation accuracy and execution time on English and Traditional Chinese evaluation sets

Method	English		Traditional Chinese	
	Accuracy(%)	Time(second)	Accuracy(%)	Time(second)
Random sense	18.25	13	27.99	< 1
Most common sense	78.21	41	92.56	< 1
Medelyan <i>et al.</i>	85.96	5,568	94.65	757
M&W with C4.5	85.60	5,917	93.79	1,167
M&W with bagged C4.5	85.14	5,948	93.98	1,164
TSDW(Dice, $M_1 = 5, M_2 = 0$)	92.39	78	95.96	64.0
TSDW(Dice, $M_1 = 5, M_2 = 5$)	93.08	104	96.00	64.2
TSDW(Dice, $M_1 = 10, M_2 = 0$)	92.68	155	96.39	73.0
TSDW(Dice, $M_1 = 10, M_2 = 5$)	92.79	183	96.24	73.3
TSDW(Dice, $M_1 = 15, M_2 = 0$)	92.35	226	96.29	79.0
TSDW(Dice, $M_1 = 15, M_2 = 5$)	92.58	258	96.31	82.4
TSDW(Jaccard, $M_1 = 5, M_2 = 0$)	92.25	78	95.91	62.0
TSDW(Jaccard, $M_1 = 5, M_2 = 5$)	92.96	104	95.93	64.7
TSDW(Jaccard, $M_1 = 10, M_2 = 0$)	92.47	156	96.24	72.0
TSDW(Jaccard, $M_1 = 10, M_2 = 5$)	92.67	183	96.24	74.0
TSDW(Jaccard, $M_1 = 15, M_2 = 0$)	92.08	225	96.22	79.0
TSDW(Jaccard, $M_1 = 15, M_2 = 5$)	92.52	259	96.08	82.6
TSDW(WLM, $M_1 = 5, M_2 = 0$)	93.74	78	96.69	64.0
TSDW(WLM, $M_1 = 5, M_2 = 5$)	94.15	102	97.05	64.6
TSDW(WLM, $M_1 = 10, M_2 = 0$)	94.14	167	97.19	72.0
TSDW(WLM, $M_1 = 10, M_2 = 5$)	94.37	180	97.34	72.4
TSDW(WLM, $M_1 = 15, M_2 = 0$)	94.14	226	97.41	79.0
TSDW(WLM, $M_1 = 15, M_2 = 5$)	94.35	256	97.36	81.5

$\theta = 0.9$ and $M_2 = 5$, according to the findings reported in evaluation of *TSDW*. As for the size of the context in the first stage, we vary M_1 using three different settings: 5, 10 and 15. The disambiguation accuracy and execution time of the evaluated methods on English evaluation set are reported in Table 4.2 (see Columns 2 and 3). Note that, for *Random sense*, the result is averaged over 10 runs. The number of ambiguous keyphrases that are processed by the second stage disambiguation is reported in Table 4.3 (see Column 2).

Overall, *TSDW* with WLM and $M_1 = 10, M_2 = 5$ achieves the best accuracy among all methods. Meanwhile, the methods with Dice and Jaccard yield competitive accuracies. The method of Medelyan et al. performs significantly better than M&W with C4.5 and bagging C4.5 classifiers. While classifier bagging improves the accuracy by 0.3%

in [103], it does not contribute positively to the accuracy in our experiments. All these methods, on the other hand, significantly outperform the two baselines. Specifically, most common sense delivers an accuracy of 78.21%, and random guess has an accuracy of 18.25%. Compared to the approach without second stage disambiguation (*i.e.*, $M_2 = 0$), the two-stage disambiguation offers positive improvements on the accuracy for all three relatedness measures and M_1 values. Considering that the number of ambiguous keyphrases undergoing the second stage disambiguation process is relatively small, the contribution is rather significant. As for efficiency, our results are significantly faster than Medelyan et al. and M&W. For instance, with $M_1 = 10$ and $M_2 = 5$, *TSDW* achieves at least 30 and 10 times faster than Medelyan et al. and M&W respectively. Observe that the additional execution time incurred by the second stage disambiguation is relatively small, because of the small number of ambiguous keyphrases that need to go through the second stage disambiguation process. Note that, the time taken by random sense and most common sense are mainly for keyphrase recognition due to the large size of Wikipedia inventory. Based on the results, we highlight that the second stage disambiguation improves both effectiveness and efficiency, compared to the approaches with first stage disambiguation only. More specifically, by setting $M_1 = 5$, $M_2 = 5$ in *TSDW*, a higher accuracy and efficiency is achieved than applying only first stage disambiguation with $M_1 = 10$ (*i.e.* *TSDW* with $M_1 = 10$, $M_2 = 0$) for all three relatedness measures. Similar observation is made for the case of *TSDW* with $M_1 = 10$, $M_2 = 5$.

We next report the experimental results on Traditional Chinese dataset. For *TSDW*, we set $c = 1.4$, 1.3 and 4.0 for *Dice*, *Jaccard* and *WLM* respectively. The confidence threshold θ and M_2 are fixed at 0.9 and 5. Similarly, we vary M_1 to three different values: 5, 10 and 15.

The disambiguation accuracy and execution time of all methods on the Traditional Chinese dataset are reported in Table 4.2 (see Columns 4 and 5). The number of ambiguous keyphrases that are processed by the second stage disambiguation is reported

Table 4.3: Number of ambiguous keyphrases processed by the second stage disambiguation with different settings for *TSDW* on English and Traditional Chinese evaluation sets

Setting	English	Traditional Chinese
TSDW(Dice, $M_1 = 5$)	1,461	427
TSDW(Dice, $M_1 = 10$)	1,485	400
TSDW(Dice, $M_1 = 15$)	1,421	411
TSDW(Jaccard, $M_1 = 5$)	1,459	420
TSDW(Jaccard, $M_1 = 10$)	1,493	415
TSDW(Jaccard, $M_1 = 15$)	1,424	423
TSDW(WLM, $M_1 = 5$)	923	314
TSDW(WLM, $M_1 = 10$)	871	292
TSDW(WLM, $M_1 = 15$)	857	290

in Table 4.3 (see Column 3). Observe that Traditional Chinese dataset is easier for word sense disambiguation compared to the English version. *Random sense* and *Most common sense* offer accuracies of 27.99% and 92.56% respectively, a much better performance than for English articles, probably because each ambiguous keyphrase in the Chinese Wikipedia inventory has only 3.0 candidate topics on average, compared to 4.2 in the English version. Similar difference is also reflected in Table 4.1 (see Column 6). All methods achieve better accuracies than the *Most common sense* baseline. Similar to what we observe on English dataset, the method of Medelyan et al. performs significantly better than M&W with C4.5 and bagging C4.5 classifiers. The classifier with bagging achieves marginally better accuracy than C4.5. Our previous approach achieves much better performance than the three existing approaches in terms of both accuracy and efficiency, for all three relatedness measures and M_1 values. Meanwhile our previous approach offers 90% reduction on average in computation time compared to the three existing approaches. *TSDW* gives both marginally positive and negative effects across all settings. 5 out of the 9 cases benefit slightly from the second stage disambiguation; 3 out of the 9 cases show slight performance degradation. The additional computation time incurred by the second stage disambiguation is almost negligible. This is confirmed by the relatively small number of ambiguous keyphrases handled by the second stage,

mainly because the accuracy of the first stage disambiguation is very high (about 96%). With $M_1 = 5$, we observe that *TSDW* achieves a slightly better performance than the corresponding method with first stage disambiguation only for all three relatedness measures. This indicates that the updated context in the second stage disambiguation indeed brings in more discriminative information. When $M_1 = 10$ or 15, degradations are observed for some settings. One possible reason is that Traditional Chinese Wikipedia is still immature and under development. Compared to English Wikipedia, Traditional Chinese Wikipedia is an order-of-magnitude smaller in the size. Thus, a larger context (i.e. $M_1 = 15, M_2 = 5$) may incur more noise than benefit. However, the experimental results (5 vs 3) demonstrates that *TSDW* is still a promising approach across different languages. Considering Table 4.2 and Table 4.3 together, we can see that the number of ambiguous keyphrases of low confidence is positively correlated with the accuracy for both English and Traditional Chinese articles. It partially justifies the correctness of the confidence measure we proposed in Equation 4.10 (further investigation of the affect by confidence threshold θ is conducted in evaluation of *TSDW*).

4.3.3 Comparison with Illinois Wikifier

In this section, we compare *TSDW* with *Illinois Wikifier* on the four datasets used in [119], ranging from short newswire to Wikipedia paragraphs. The four datasets are briefed as follows:

- AQUAINT, is a subset of the AQUAINT corpus of newswire text where the Wikipedia keyphrases are annotated to Wikipedia topics. This dataset was also used in [103].
- MSNBC, is taken from MSNBC news, where only named entities after running NER are disambiguated to Wikipedia. This dataset was used in [30].

Table 4.4: Statistics on the four datasets. #articles: the number of documents; #word: the average number of words per document; #unambiguous: the average number of unambiguous keyphrases per document; #ambiguous: the average number of ambiguous keyphrases per document; #candidates: the average number of candidate topics per ambiguous keyphrase.

Dataset	#articles	#word	#unambiguous	#ambiguous	#candidates
AQUAINT	50	1,347	18.4	8.8	55.4
MSNBC	20	3,262	40.0	10.3	81.6
ACE	57	2,276	28.4	2.6	94.3
Wikipedia	40	3,628	47.1	12.3	47.4

- ACE, is a subset of ACE co-reference dataset which was built by [119]. The annotations to Wikipedia topics were assigned by Amazon Mechanical Turk⁹, and the inconsistent annotations were manually corrected by the authors.
- Wikipedia, is a sample of paragraphs from Wikipedia articles sampled by [119]. The annotations correspond to the hyperlinks in the Wikipedia text.

Because *Illinois Wikifier* was built using the English Wikipedia dump of 2009, for a fair comparison, we retain all the annotations that are both solvable to *Illinois Wikifier* and the Wikipedia inventory we build in this work. Solvable annotations refer to the annotations that appear in the inventory and the correct disambiguations are among the candidates indexed by the inventory [119]. The unsolvable annotations in the datasets are removed. The details of the four datasets are reported in Table 4.4.

Table 4.5 reports the disambiguation accuracies and execution times on the four datasets. We fix $M_1 = 10$ and $M_2 = 5$ for *TSDW*. Observe that *TSDW* outperforms *Illinois Wikifier* in almost all cases. While the second stage disambiguation of *TSDW* does not contribute any positive improvement on ACE, it provides additional benefit for the other three datasets. The main reason is that there are very few ambiguous keyphrases from the articles in ACE dataset; the dataset has only 2.6 ambiguous keyphrases on average for each article. For this reason, there is almost no additional discriminative

⁹<https://www.mturk.com/>

Table 4.5: Disambiguation accuracy (%) and execution time (second) of *TSDW* and *Illinois Wikifier* on four evaluation sets: $M = 10$ for *TSDW*

Method	ACE		AQUAINT		MSNBC		Wikipedia	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
Illinois Wikifier	85.91	123.4	84.35	99.3	81.76	108.4	87.39	264.1
TSDW(WLM, $M_2 = 0$)	93.92	7.1	91.16	8.8	84.47	9.5	88.84	10.0
TSDW(WLM, $M_2 = 5$)	93.24	8.2	91.84	10.2	85.44	10.4	90.47	10.6
TSDW(Jaccard, $M_2 = 0$)	93.92	7.2	92.97	8.7	85.92	9.0	86.00	9.9
TSDW(Jaccard, $M_2 = 5$)	93.92	8.1	93.20	9.9	85.92	10.2	87.22	11.0
TSDW(Dice, $M_2 = 0$)	93.92	7.0	92.97	8.7	85.44	8.6	86.00	10.0
TSDW(Dice, $M_2 = 5$)	93.92	8.1	93.42	10.0	85.92	10.1	87.22	11.2

information that can be explored in the second stage disambiguation. *Illinois Wikifier* achieves comparable accuracy with our method on Wikipedia dataset for most settings used in *TSDW*. However, using *WLM* as relatedness measure and two-stage disambiguation, our method achieves the best accuracy on the dataset. In terms of efficiency, as can be observed in Table 4.5, *TSDW* is at least 10 times faster than *Illinois Wikifier* for all the four datasets. Note that Ratnov *et al.* only used the top 20 candidate topics of the highest likelihood in their system, while the average numbers of candidate topics per ambiguous keyphrase for the four datasets vary from 47 to 94 in *TSDW* (see Table 4.4). This again confirms that *TSDW* is superior to *Illinois Wikifier* in terms of efficiency. Compared to the performance reported in [119], we observe that the disambiguation accuracy degrades a bit for *Illinois Wikifier* in our experiments. One possible explanation is that the different sets of solvable annotations are used in the experiments. In particular, only a subset of solvable annotations used in [119] is used for our evaluation. Similar performance deterioration is also observed for *TSDW* on the Wikipedia dataset, compared to our previous experiments conducted earlier. Recall that AQUAINT dataset was also used in Milne & Witten’s work [103]. While it is unfair to compare the effectiveness directly because of different experimental settings, an accuracy of 76% was reported in their original work [103].

4.3.4 Evaluation of *TSDW*

To evaluate the disambiguation accuracy of *TSDW* and the impact of the parameter settings, we conduct another set of experiments on the training sets of 500 articles in English and Traditional Chinese respectively. Recall that our proposed approach involves four parameters M_1 , M_2 , c and θ , and a relatedness measure. M_1 determines the number of unambiguous keyphrases involved in the first stage disambiguation; M_2 determines the number of additional context keyphrases in the second stage; θ is the threshold for the high/low-confident keyphrases from the first stage disambiguation, and c is specific to the relatedness measure.

4.3.4.1 First Stage Disambiguation

The first stage disambiguation involves two parameters: M_1 for the size of the context, and c for the probability approximation. We apply grid search to analyze the impact of these two parameters on both the datasets: English and Traditional Chinese datasets. We investigate the impact of M_1 value on the disambiguation accuracy by varying M_1 from 5 to 50 with a step of 5, and *All* that takes all unambiguous keyphrases into account, given c is fixed. Similarly, we learn an c value for each given relatedness measure by varying c from 0 to 10 with a step of 0.1, given M_1 is fixed. Observe that, when $c = 0$, our method degrades to the most common sense method. Three types of relatedness measures, *Dice*, *Jaccard* and *WLM* are evaluated in *TSDW*.

Figures 4.3(a) and 4.3(b) report the disambiguation accuracy of the first stage disambiguation by varying M_1 and c on three relatedness measures for English and Traditional Chinese datasets respectively. We make the following observations on the experimental results:

- Parameter c significantly affects the results for all relatedness measures. For a specific relatedness measure, different optimal c values are observed for the two

(a) English

(b) Traditional Chinese

 Figure 4.3: Accuracy of varying M_1 and c with Dice, Jaccard and WLM for two datasets.

datasets. When *Dice* is used, $c = 1.5/1.4$ gives the best accuracy across different M_1 values for English and Traditional Chinese datasets respectively. Similarly, $c = 1.5/1.3$ gives the best accuracy for *Jaccard*. For *WLM*, the best accuracy is achieved when c is in the range of $[5.0, 7.0]/[3.5, 5.0]$ for English and Traditional Chinese datasets respectively.

- A larger M_1 does not necessarily lead to better accuracy for both datasets. In particular, accuracies dropped for all settings when $M_1 = All$, i.e. taking all unambiguous keyphrases as the context. Specifically, $M_1 = 5, 10$ and 15 offer the best accuracies.

Observe that the two figures showed in Figure 4.3 demonstrate very similar patterns. Nevertheless, the comparison of the two sets of results reveals that parameter c is dependent on not only relatedness measure but also the Wikipedia dataset. Considering the size difference between the two Wikipedia datasets (English and Traditional Chinese), the characteristics of a specific relatedness measure based on the hyperlink structure would be largely affected. To better demonstrate the impact of c value on the three relatedness measures over the two languages, as a case study, we calculate the pair-wise relatedness measure between the Wikipedia topic *Great Wall of China* and all its out-going neighbors for both English and Traditional Chinese respectively, using *Dice*, *Jaccard* and *WLM*. There are respectively 105 and 335 out-going neighbors for English and Traditional Chinese articles about the topic *Great Wall of China*. Table 4.6 reports the mean, standard deviation (std) and coefficient of variation (CV) for each set of similarity values to empirically reflect the varying characteristics of each similarity measure over Wikipedia of the two languages. The relatedness values by *Dice* and *Jaccard* are widely scattered for both English and Traditional Chinese settings. *WLM* produces a narrow dispersion of relatedness values. This is consistent with the observations made in Figure 4.3 that a larger c obtains a better disambiguation accuracy with *WLM*. Moreover, we can observe that each similarity measure holds different value patterns for English and Traditional Chinese, in terms of the three statistics we studied here. Hence, different optimal c values for a specific relatedness measure over the two Wikipedia datasets is reasonable. This also illustrates the robustness of TSDW in adapting to different settings.

In the first stage disambiguation, *TSDW* filters away noisy contextual information by retaining only top M_1 unambiguous keyphrases with the highest *keyphraseness* values. An alternative option is to apply a predefined threshold value for the *keyphraseness*, so that all unambiguous keyphrases with *keyphraseness* value larger than the threshold are considered as the context. Such a threshold can be applied globally to all articles of

Table 4.6: Relatedness distribution using *Dice*, *Jaccard* and *WLM*. *en* and *zh* refer to the case study by using English and Traditional Chinese article about the topic *Great Wall of China* respectively.

Measure	Mean		Std		CV	
	En	Zh	En	Zh	En	Zh
<i>Dice</i>	0.025	0.078	0.028	0.096	1.120	1.231
<i>Jaccard</i>	0.013	0.044	0.015	0.070	1.154	1.591
<i>WLM</i>	0.435	0.527	0.245	0.166	0.563	0.315

interest. However, given such a threshold, an article may have too many or too few unambiguous keyphrases left as context. As observed in Figures 4.3(a) and 4.3(b), more unambiguous keyphrases did not lead to significantly better disambiguation accuracy on either English or Traditional Chinese dataset. Figure 4.4 plots the boxplot¹⁰ of the number of unambiguous keyphrases per article with *keyphraseness* values larger than a specific threshold for the 500 articles of English training set. We make two observations: (i) 50% of the articles contain more than 10 unambiguous keyphrases with *keyphraseness* larger than 0.8; and (ii) 10% of the articles contain just one unambiguous keyphrase whose *keyphraseness* is larger than 0.1. That is, a high threshold on *keyphraseness* leads to a empty context for these articles.

We also conduct experiments for the first stage disambiguation on English training set by applying a specific *keyphraseness* threshold. If an article has no unambiguous keyphrase left as context after applying the threshold, *Most common sense* is used for disambiguation. Table 4.7 lists the accuracies obtained by using different threshold values, along with the corresponding computation time. Given the marginal change of the accuracy, two observations are made: (i) a low threshold leads to more computation and relatively low accuracy; and (ii) a high threshold achieves a shorter computation time with relatively low accuracy. These two observations are consistent with the results shown in Figure 4.4. By applying $M_1 = 15$, an accuracy of 95.32% is achieved with

¹⁰Boxplot is a convenient way of graphically depicting groups of numerical data through their five-number summaries: 10th, 25th, 50th, 75th and 90th percentiles.

Table 4.7: Disambiguation accuracy and execution time by applying varying *keyphraseness* thresholds on English training set.

Threshold	Accuracy	Time	Threshold	Accuracy	Time
0.1	95.27	1183.7	0.6	95.37	740.9
0.2	95.31	1019.6	0.7	95.36	672.0
0.3	95.34	935.4	0.8	95.23	622.8
0.4	95.34	878.4	0.9	94.94	521.2
0.5	95.40	824.5	1.0	94.89	456.6

Figure 4.4: Boxplot for the number of unambiguous keyphrases per article that with *keyphraseness* value above a threshold

execution time of 482.8 seconds. Compared with results listed in Table 4.7, retaining top M_1 unambiguous keyphrases as the context achieves comparable accuracy with shorter computation time.

The first stage disambiguation of *TSDW* predicts a confidence for each disambiguation decision. Here, we investigate the correctness of the confidence measure used in Equation 4.10 by using the three relatedness measures, and three M_1 values: 5, 10, 15. Parameter c for *Dice*, *Jaccard* and *WLM* is fixed at 1.5, 1.5, 6.0 / 1.4, 1.3, 4.0 for English and Traditional Chinese respectively. Let the accuracy at confidence cf_d be the accuracy of disambiguation decisions with confidence less than or equal to cf_d . Figures 4.5(a) and 4.5(b) plot the accuracies at varying cf_d on the English and Traditional Chinese

(a) English

(b) Traditional Chinese

Figure 4.5: Accuracy (%) at varying θ on the two training sets

training datasets respectively. Observe that the accuracy increases monotonically with the confidence. An accuracy of about 50% is achieved when the confidence is 0.5 for all settings. The accuracy of disambiguation decisions with confidence larger or equal to 0.9 is more than 98% for all cases. Accordingly, we set $\theta = 0.9$ to distinguish the low/high confident disambiguation decisions. The experimental results show that the confidence measure is a reasonable indicator to reflect the quality of disambiguation decisions.

4.3.4.2 Second Stage Disambiguation.

The second stage disambiguation of *TSDW* involves two parameters: confidence threshold θ and the number of new context keyphrases M_2 . According to experimental results in previous section, we fix θ to be 0.9. The impact of M_2 is studied by varying its value from 0 to 20 with a step of 5. The experiments are conducted by using $M_1 = 5, 10, 15$ and the best c on the three relatedness measures found earlier. Figures 4.6(a) and 4.6(b) illustrate the effectiveness of the second stage disambiguation on different M_2 values on English and Traditional Chinese training datasets respectively. Note that $M_2 = 0$ is equivalent to the output of using first stage disambiguation only. From Figure 4.6(a), we observe that the second stage disambiguation further improves the disambiguation accuracy with varying M_2 values for English articles. The largest improvement is observed when $M_2 = 5$ for all the three relatedness measures. This indicates that more additional context keyphrases bring in more noise than discriminative information. For articles in Traditional Chinese, this side effect of bringing in more noisy context keyphrases becomes more apparent. The marginal improvement by the second stage disambiguation is obtained when M_1 is 5 for all the three relatedness measures. We believe that this is specific to the nature of the Traditional Chinese Wikipedia we studied (*i.e.*, incomplete and relatively small in size). Overall, based on the experimental results, we see that $M_2 = 5$ is a reasonable value for the size of additional context in the second stage. For both English and Traditional Chinese articles, *WLM* achieves the best performance over *Dice* and *Jaccard*. This has been confirmed by the extensive experiments conducted in earlier sections. Considering the size of ambiguous keyphrases undergoing the second stage disambiguation is relative small, the positive improvements we observed here are significant.

4.3.4.3 Multi-stage Disambiguation

As demonstrated in Figure 4.2, *TSDW* naturally support multiple re-disambiguations, such as a third stage disambiguation. The stop criterion for further disambiguation

(a) English

(b) Traditional Chinese

 Figure 4.6: Accuracy (%) of varying M_2 on the two sets of articles

stage can be defined based on the amount of additional information gained through the disambiguation decisions of the current stage, *i.e.*, the number of high confident disambiguation decisions of the current stage. Given the high accuracy of the first stage disambiguation, it is expected that the additional information is very limited after the second stage disambiguation. We conduct experiments on the English training set with up to fourth stage disambiguation by using *WLM* relatedness measure. We fix θ , c and M_1 to 0.9, 6.0 and 5 respectively. The number of additional context added at the next stage is fixed to be 5, *i.e.*, $M_i = 5$ for $i = 2, 3, 4$. Table 4.8 reports the accuracies along

with the number of the high-confident disambiguation decisions obtained after each stage. Observe that, after the second stage, only 399 high-confident disambiguation decisions are obtained for the 500 articles. This leads to very little improvement at further stages. Similar observations hold for different settings (*i.e.*, relatedness measure, parameter values and language). Thus, we restrict *TSDW* to have two-stage disambiguation for better efficiency and negligible loss in effectiveness.

4.3.4.4 Error Analysis

We analyze the disambiguation errors made by *TSDW* manually. We find that most disambiguation errors happen when the correct topic is ranked as the second best candidate topic. By setting $M_1 = 15, M_2 = 5$, we found that the accuracy of top-2 and top-3 is 98.56%/99.04% respectively on *Evaluation_{en}*. Similar results are also observed on *Evaluation_{zh}*, 99.13%/99.69%. Particularly, when both the top two best candidate topics are closely related to the context of the article. For example, *joey* can not be disambiguated correctly to the topic *Joey (1985 film)*, but to a wrong topic *Joey (TV series)* in Wikipedia article *Don Porter*¹¹. The actor *Don Porter* appeared in the film *Joey (1985 film)*¹². Since both topics belong to the same concept *show business*, they have very similar context relatedness values. Thus, topic *Joey (TV series)* is wrongly selected because it has a much higher prior probability than topic *Joey (1985 film)* (0.462 vs 0.016). We observe that this kind of mistakes are often related to low confident disambiguation decisions. Table 4.9 lists the proportion of the errors that are of low ([0.0, 0.5)), medium ([0.5, 0.8)) and high ([0.8, 1.0)) confidence. It is clear from the table that about half of the mistakes are from the predictions of low confidence. In this sense, we believe that the use of other relatedness measures, such as cosine similarity based on bag-of-word model, or expert-based similarity based on authorship (Chapter 3) might help us out. Another

¹¹http://en.wikipedia.org/wiki/Don_Porter

¹²[http://en.wikipedia.org/wiki/Joey_\(1985_film\)](http://en.wikipedia.org/wiki/Joey_(1985_film))

Table 4.8: Accuracies (%) and the number of the high-confident disambiguation decisions (#HiCfd) at different stages.

#Stage	Accuracy	#HiCfd
1	94.63	13,708
2	95.34	399
3	95.36	55
4	95.37	8

Table 4.9: The proportions of errors that are of low, medium and high confidence.

Range	<i>Evaluation_{en}</i>	<i>Evaluation_{zh}</i>
[0, 0.5)	47.24%	63.38%
[0.5, 0.8)	34.66%	21.12%
[0.8, 1.0)	18.10%	15.50%

source of disambiguation errors is that an ambiguous keyphrase happens to have a false candidate topic that is very related to the main topic of the article, while its true topic is only related to the surrounding semantic context. For example, the ambiguous keyphrase *underground* appears in the Wikipedia article *The Vinyl Underground*¹³, which is about a comic book series. *underground* refers to the Wikipedia topic *London Underground* in the section of "Characters" of the article. However, because *underground* has a candidate topic *Underground comix*, which is about self-published comic books, *TSDW* mistakenly disambiguates the keyphrase to topic *Underground comix*. In this case, it is better to take the local context information into account to help disambiguation.

4.4 Summary

Word sense disambiguation to Wikipedia is a key component in many applications in the areas of Natural Language Processing, Information Retrieval and others. In this chapter, we propose an innovative two-stage framework for disambiguation to Wikipedia. In contrast to existing works, *TSDW* leverages the semantic clue from both unambiguous

¹³http://en.wikipedia.org/wiki/The_Vinyl_Underground

and ambiguous keyphrases in a given document. The context of the first stage is defined by pruning unimportant and noisy unambiguous keyphrase, leading to a highly efficient and effective disambiguation process for all ambiguous keyphrases. With the confidence measure, the high quality knowledge provided by the ambiguous keyphrases is recruited as additional contextual information in the second stage disambiguation. Since the second stage disambiguation focuses on a small size of ambiguous keyphrases of low confidence, better accuracy is obtained from the additional discriminative knowledge with little extra computation. Extensive experiments are conducted to study the performance of *TSDW* using datasets in two languages, English and Traditional Chinese, to validate its generalization ability. Experimental results show that *TSDW* generalizes well to different languages and measures, and achieves better disambiguation accuracy with lower computation than state-of-the-art approaches. *TSDW* is an ideal choice for Wikification related tasks. In the next chapter, we will apply *TSDW* for the task of semantic tag recommendation, where we represent each web page by their associated Wikipedia topics.

Chapter 5

Tag Recommendation based on Concept Model*

You can, for example, never foretell what any one man will do, but you can say with precision what an average number will be up to. Individuals vary, but percentages remain constant. So says the statistician.

– Sherlock Holmes

5.1 Introduction

Recently, digital resources are increasingly being organized, summarized, shared and searched using tagging mechanisms. Social tagging is a system of classification derived from collaboratively creating and managing tags to annotate and categorize content. Consequently, there is immense research interest in efficient and effective tag recommendation techniques to boost manual tagging to cope with the continuously growing amount of digital data being produced.

In social tagging, often, people emulate or share others' tagging behavior. As a result, semantically relevant tags emerge as the prominent tags with top frequency of occurrences, and a small set of common tags are typically reused to represent specific

*This chapter is based on the paper *Semantic Tag Recommendation using Concept Model* by Chenliang Li, Anwitaman Datta, Aixin Sun, published in *ACM SIGIR 2011*.

topics, which is known as *semantic tags* [134, 51]. Semantic tags are normally common nouns and proper names (75.63% and 70.10% respectively), belonging mainly to the category of topic which describes what a tagged resource (*e.g.*, a document) is about [19]. Thus, it is reasonable to infer that a user subconsciously considers the concepts covered by a document when she is tagging that document. However, the traditional bag-of-word model would not provide us with such concrete semantic information.

In Chapter 4, we propose a generalized framework for word sense disambiguation using Wikipedia. With it, we can represent each textual document with the set of concrete information units (*i.e.*, Wikipedia concepts), instead of bag-of-word model. In this chapter, we attempt to emulate human tagging behavior, in that, people annotate semantic tags based on the concepts underlying a document and the semantic tags associated with these concepts. Specifically, we adopt a probabilistic framework to model a web document by its contained concepts and the likelihood of the semantic tags representing these concepts for tag recommendation. In our implementation, the concepts are derived from Wikipedia by taking each Wikipedia article as a concept.

Evaluated on a Delicious dataset containing more than 53K documents, our tag recommendation technique achieved comparable accuracy as state-of-the-art techniques, but yields a significant speed-up. This suggests that a small number of concepts contained in a document are effective in conveying the meaning of the document for meaningful tag recommendation. The concise concept-based representation against the commonly adopted unigram representation contributes to the speed-up.

5.2 Concept model

The key of the concept model is to represent a document using a small set of concepts contained in the document. In our implementation, the concept space is derived from

Wikipedia, which is the largest knowledge repository. Each Wikipedia article (*i.e.*, article title) defines one concept.

Given a document, all keyphrases, each of which refers to at least one Wikipedia article, contained in the document are first identified through string matching (see Section 4.2.2). The keyphrases inventory is built by following the work presented in Section 4.2.1. The extracted keyphrases are ranked by their *keyphraseness*, the priori probability that a keyphrase is used as anchor text of the wikilinks, no matter where it appears [104]. The lower-ranked keyphrases are then pruned for two reasons. First, the lower-ranked keyphrases are often not strongly related to any concepts, and hence are less effective in conveying the meaning of the document. Second, the pruning leads to a more concise representation of the document, enabling more efficient tag recommendation. The pruning is controlled by a parameter (*prune ratio*) set based on a validation set. Then, since a keyphrase may match multiple concepts, we apply word sense disambiguation technique presented in Chapter 4 to derive the correct concepts under the context of the document of interest.

With concept based document representation, we adopt Equation 5.1 for semantic tag recommendation, where $P(t|d)$ is the likelihood that a tag t represents the meaning conveyed by document d , $P(t|c)$ is the likelihood that tag t represents concept c , and $P(c|d)$ denotes representativeness of concept c to document d .

$$P(t|d) = \sum_c P(t|c)P(c|d) \quad (5.1)$$

The tag recommendation problem is then to pick the top- k tags with the highest $P(t|d)$ values. We next discuss the estimation of $P(c|d)$ and $P(t|c)$ respectively.

Let $f(d, c)$ be the number of times concept c appears in document d . $P(c|d)$ is estimated by the concept's relative frequency with Jelinek-Mercer smoothing, shown in Equation 5.2. In this equation, $\lambda \in [0, 1]$ is a smoothing parameter and $P(c|\mathcal{D})$ is the

maximum likelihood estimate of concept c in the entire collection $\mathcal{D}$.

$$\hat{P}(c|d) = (1 - \lambda) \frac{f(d, c)}{\sum_{c'} f(d, c')} + \lambda P(c|\mathcal{D}) \quad (5.2)$$

The probability $P(t|c)$ is estimated through the concept model of tag t (*i.e.*, $P(c|t)$) by Bayes' theorem: $P(t|c) = \frac{P(c|t)}{P(c)}P(t)$. While $P(c)$ and $P(t)$ can be easily estimated based on their corresponding frequencies in the collection, the estimation of $P(c|t)$ is not straightforward. Let D_t be the set of documents annotated by tag t . We build the concept model of tag t by a weighted concatenation of the concepts appearing in D_t , shown in Equation 5.3.

$$\hat{P}(c|t) = \frac{\sum_{d \in D_t} w(d, t) f(d, c)}{\sum_{c'} \sum_{d \in D_t} w(d, t) f(d, c')} \quad (5.3)$$

In this equation, $w(d, t)$ denotes the relevance or representativeness of document d towards tag t . Note that, many documents may carry the same tag, and a particular document may have multiple tags with varying frequencies given by multiple users (*e.g.*, Delicious).

Let T_d denote the set of tags assigned to document d , and $f(d, t)$ be the number of times tag $t \in T_d$ is assigned to d by (multiple) users. We consider four schemes to compute $w(d, t)$, summarized in the following, where $f(t)$ is the frequency of tag t in the entire collection.

- Uniform Weighting (UW): the uniform weight of 1.0 is used:

$$w(d, t) = 1.0 \quad (5.4)$$

- Document Perspective (DP): $w(d, t)$ is defined as the relative frequency of tag t annotated to d with respect to the tag with the maximum frequency for d :

$$w(d, t) = \frac{f(d, t)}{\max_{t' \in T_d} f(d, t')} \quad (5.5)$$

- Tag Perspective (TP): $w(d, t)$ is defined as the proportion of the annotations of tag t to d :

$$w(d, t) = \frac{f(d, t)}{f(t)} \quad (5.6)$$

- Document & Tag Perspectives (DTP): We define $w(d, t)$ by combining the both DP and TP as follows:

$$w(d, t) = \frac{f(d, t)}{f(t)} \cdot \frac{f(d, t)}{\max_{t' \in T_d} f(d, t')} \quad (5.7)$$

TP measures the importance of document d with respect to tag t by counting the percentage of the annotations of t to d , while DP measures the relevance between t and d by calculating the relative frequency to the most frequent one. Besides these three weighting schemes, of course, we can just treat each document uniformly for each tag, as Naive Bayes approach does (see Equation 5.4). The effects of different weighting schemes are evaluated in Section 5.3.

5.3 Experiments

Wikipedia The concept space (*i.e.*, keyphrase inventory) used in Chapter 4 is used here. To avoid the adverse impact of noisy information, we discard the articles with less than 15 incoming and outgoing links. That is, we got 579,249 articles in the concept space. Correspondingly, We extracted 4,342,732 keyphrases for the keyphrase inventory building.

Del.icio.us is a popular online social bookmarking service and the biggest collection of bookmarks on the web. We evaluate the proposed concept based tag recommendation using Delicious dataset comprising 144,574 documents and 67,104 distinct tags [170]. Each bookmark has on average 182.5 users who have tagged it, with a standard variance of 663.5. And the number of distinct tags applied to a bookmarks ranges from 1 to 37,

Table 5.1: Statistics on datasets

Dataset	#bookmarks	Avg@Words	Avg@Users	Avg@Tags
Training	49,550	1,532.0	408.2	22.2
Development	1,000	1,524.3	408.7	22.4
Evaluation	3,000	1,528.9	390.7	22.2

with the mean value of 14. We define the popularity of a bookmark as the number of users tagged it. Unpopular bookmarks often have few users and distinct tags, the relative frequencies of tags may not reflect the correct relation between the content of the bookmarks and their tags. In other words, it could incur much noise into the learning models by including these less popular bookmarks. Meanwhile, it is also harmful to evaluate the effectiveness of the system over these bookmarks, since they have very few tags. Moreover, bookmarks with few contextual content also may affect the concept models learnt since we are unable to extract any meaningful concepts for these bookmarks, such as web pages for images, video, etc. In this work, we filter the dataset by removing all bookmarks with less than 50 users and less than 100 words in their content. After the filtering process, we end up with 53,550 bookmarks from the initial dataset. We randomly split these bookmarks into three dataset: 49,550 for training, 1,000 for development and 3,000 for evaluation. The statistics of the three datasets are reported in Table 5.1.

Methods. Three baseline methods are compared in our experiments. The naïve baseline is to recommend the most popular tags (MPT) to all documents. The state-of-the-art baseline is the one based on the language generative model (TPL) reported recently in [163]. To evaluate the effectiveness of the notion of concept, another set of baseline methods are designed by using uni-gram (*i.e.*, words) in the documents. Then Equation 5.1 is rewritten as $P(t|d) = \sum_w P(t|w)P(w|d)$. Coupled with the four weighting schemes in Equations 5.4-5.7, we use CM+{UW, DP, TP, DTP} and UM+{UW, DP, TP, DTP} to denote the four methods using concept model and uni-gram model respectively.

Performance Metrics. Since our purpose is to automatically recommend tags based on the semantic content of a document, it is more reasonable to recommend a list of tags that are most semantically relevant to the document of concern. By taking the tags annotated to documents in the evaluation set as the ground truth, we prefer the method that ranks higher frequency tags higher. For evaluation, given a ranking of top N tags recommended $r_d^t = \langle t_1, t_2 \dots t_N \rangle$ for document d , we denote with $t_{d,i}^r$ the i^{th} tag in the ranking. It is considered better that more recommended tags are present in the set of tags annotated to d , and the ranking r_d^t is consistent with the ranking of the annotated tags based on the tag frequency $f(d, t)$. In this work, We evaluate the performance of different tag recommendation methods by adopting the Normalized Discounted Cumulative Gain ($NDCG$) measure. $NDCG$ measures the consistency between two rankings, i.e., the similarity of the orderings of the data in the two rankings. Moreover, $NDCG$ gives high positions higher weight than the low ones. Thus, in the context of social tagging systems, it gives more gain when the recommended tags are higher ranked tags in the ground truth. We adopt the variation of $NDCG$ measure proposed by Suchanek et al. [134] in their study for social tagging context. In detail, let T_d denotes the set of tags annotated to document d , $NDCG$ at the top N recommended tags is defined as follow:

$$NDCG@N(r_d^t) = \frac{\sum_{i=1}^N \frac{f(d, t_{d,i}^r)}{\log(i+1)}}{\sum_{i=1}^{|T_d|} \frac{f(d, t_{d,i})}{\log(i+1)}} \quad (5.8)$$

From Equation 5.8, we can see that the higher ranked recommended tags with the higher frequencies result in the higher $NDCG$ score. We also report the performance in terms of Precision at the top N recommended tags ($Prec@N$) as a reference. $Prec@N$ is the ratio of tags indeed used by users among the top- N recommended tags [92]. Note that $Prec@N$ doesn't take the tag's frequency in the ground truth into consideration.

The execution time is also reported and all experiments were conducted on a workstation with a 2.40GHz Xeon quad-core CPU and 24GB RAM.

Table 5.2: Statistics on execution time (second).

Method	min	max	mean	median	std
TPL	18.10	56.88	22.23	20.91	3.81
UM+DTP	17.82	50.06	21.30	20.23	3.18
CM+DTP	1.03	14.80	2.50	2.34	1.57

Recommendation Accuracy. Table 5.3 reports the performance of all methods in terms of $NDCG@{5,10}$ and $Prec@{5,10}$. For each metric, the best performing method is highlighted in boldface and the second best is underlined. Observe that: (i) unsurprisingly, the simple MPT performs the worst by all measures; (ii) all concept model based methods outperform uni-gram based methods significantly by all measures, regardless of weighting scheme used; (iii) among concept based methods, CM+DTP is the winner in terms of $NDCG@{5,10}$ and CM+UW performed the worst in both metrics. (iv) comparing with the state-of-the-art TPL, CM+DTP achieves better $NDCG@5$ significantly and marginal improvement in $NDCG@10$. Although TPL yields better $Prec@{5,10}$, a close investigation reveals that many tags recommended by TPL are of low frequency, which may only benefit a small group of users. This is also reflected by $NDCG@{5,10}$ as $NDCG$ measures consider tag frequency in the evaluation and tags with higher frequencies are given more weights in the measure.

Execution Time. Table 5.2 lists the statistics of the tagging time of the three algorithms. The average tagging time of UM+DTP, TPL and CM+DTP were 21.30, 22.23 & 2.50 seconds respectively. Note that the tagging time for CM+DTP includes concept detection time. That is, concept model based method incurs an order of magnitude computation cost reduction.

5.4 Summary

In this chapter, we propose a semantic tag recommendation mechanism based on concept model. The concept model enables a concise representation of documents by considering

Table 5.3: Performance of methods, where * indicates the difference against the best performance is significant with $p < 0.01$ (paired t -test)

Methods	$NDCG@5$	$NDCG@10$	$Prec@5$	$Prec@10$
UM+UW	0.1778*	0.2323*	0.4009*	0.3485*
UM+TP	0.2383*	0.2946*	0.4462*	0.3758*
UM+DP	0.2357*	0.2914*	0.4461*	0.3733*
UM+DTP	0.2638*	0.3208*	0.4673*	0.3854*
CM+UW	0.3370*	0.4032*	0.5815*	0.4784*
CM+TP	0.4081*	0.4781*	0.6312*	0.5152*
CM+DP	0.4104*	0.4809*	0.6373*	0.5162*
CM+DTP	0.4213	0.4915	0.6355*	0.5151*
MPT	0.0900*	0.1196*	0.2753*	0.2412*
TPL	0.3983*	<u>0.4843</u>	0.7055	0.5983

only a few concepts most relevant to the documents. Comparing with the state-of-the-art, our method achieved comparable tag recommendation accuracy but yield a significant speed-up. And through the experimental studies, we found that combining information from macro-view (TP weighting scheme) and micro-view (DP weighting scheme) can help us derive a more accurate semantic relations between the semantic tags and underlying concepts. It again demonstrates what we have showed in Chapter 3: combining information from different perspectives can offer us unbiased knowledge that can't be discovered by considering the various aspects in isolation.

Chapter 6

Named Entity Recognition in Targeted Twitter Stream*

Beautiful mathematics eventually tends to be useful, and useful mathematics eventually tends to be beautiful.

– Carl D. Meyer

In last few chapters, we focus on tasks related to a formal text corpus. However, a large portion of user-generated content in online social media are informally written with free writing styles, *e.g.*, users' reviews and communications, and microblogs. Such user-generated content is a fresh avenue for us to understand user opinions, and the topics they are interested in. Named entity recognition is a critical component for these tasks. However, traditional supervised methods based on linguistic features perform poorly on the user-generated content. In this chapter, we study the task of named entity recognition in the context of Twitter streams. The semantic knowledge provided by Wikipedia (*i.e.*, Wikipedia inventory built in Chapter 4, which consists of named entities and other topics) and WWW (*i.e.*, word collocation patterns) are exploited together to help identify the possible named entities in each tweet.

*This chapter is based on the paper *TwNER: Named Entity Recognition in Targeted Twitter Stream* by Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun and Bu-Sung Lee, published in *ACM SIGIR 2012*.

6.1 Introduction

Twitter, as a new type of social media, has seen tremendous growth in recent years. It has attracted great interests from both industry and academia. Many private and/or public organizations have been reported to monitor *Twitter* stream to collect and understand users' opinions about the organizations. Nevertheless, due to the extremely large volume of tweets published every day¹, it is practically infeasible and unnecessary to listen and monitor the whole *Twitter* stream. Therefore, targeted *Twitter* streams are usually monitored instead; each such stream contains tweets that potentially satisfy some information needs of the monitoring organization. Targeted *Twitter* stream is usually constructed by filtering tweets with user-defined selection criteria depends on the information needs. For example, the criterion could be a region so that users' opinions from that particular region are collected and monitored; it could also be one or more predefined keywords so that opinions about some particular events/topics/products/services can be monitored.

There is also an emerging need for early crisis detection and response with such target stream. For example, a cosmetic company could be interested in automatically discovering any new named entities (*e.g.*, person names, competitor names, or location names) in a targeted stream it creates for the company and its products, which may link to a potential public relations crisis. By doing this, the company is able to acquire first-hand information about the crisis and make early response. Such applications require a good named entity recognition (NER) system for *Twitter*, which is the focus of this chapter.

Nevertheless, the nature of tweets brings new challenges. Traditional NER methods on well-formatted documents heavily depend on a phrase's local linguistic features [118], such as capitalization, part-of-speech (POS) tags of previous words, etc. However,

¹There are more than 200 million tweets published per day, according to <http://blog.twitter.com/2011/06/200-million-tweets-per-day.html>.

tweets are usually informal in nature and short (up to 140 characters). They often contain grammatical errors, misspellings, and unreliable capitalizations. These unreliable linguistic features cause traditional methods to perform poorly on tweets. We use some real examples below to illustrate the challenges when applying traditional NER on tweets.

Figure 6.1: System Architecture of *TwiNER*

Table 6.1: Example Named Entities in Tweet

1	PAP POSTERS ARE EVERYWHERE! AND FOR SOME LAMP POLES THERE ARE BOTH NSP AND PAP POSTERS! #whathappentosavingtheearth
2	ya la!! some of them gg to potong pasir . I'm gg to yio chu kang

Table 6.1 shows two real tweets collected during a political event. A POS tagger would fail due to the tweets' abnormal capitalization and grammatical errors. For example, in the first tweet, all words (except "PAP" and "NSP")² are mislabeled as *NNP* (singular noun). Similarly, in the second tweet, "potong" and "yio" are mislabeled as *JJ* (adjective) and *VB* (verb) respectively, although both are a part of location names ("potong pasir" and "yio chu kang"). This kind of noisy POS labels would make NER tagger fail to recognize named entities.

To address the above challenges caused by tweets' error-prone and short nature, this chapter presents a novel unsupervised NER system for targeted tweet streams, called *TwiNER*. Based on the *gregarious* property of named entities in targeted tweet stream,

²"PAP" and "NSP" are short forms of political party names. Hashtags such as "#whathappentosavingtheearth" are not considered in this work.

*TwI*NER recognizes named entities collectively from a batch of tweets in unsupervised manner. More formally, let T be the collection of tweets in question. *TwI*NER receives tweets from T in a batch manner. A batch is the set of tweets posted in the targeted *Twitter* stream within one fixed time interval (e.g. a second). So, $T = \{T_1, T_2, \dots, T_n\}$ and T_i is the batch of tweets posted in the i th interval. *TwI*NER then recognizes all possible named entities in T_i regardless of their types.

It is noted that currently *TwI*NER does not categorize the type of named entity (e.g., person, location). As conventional NER methods fail to address the new challenges posed by emerging social media like *Twitter*, it is more pressing to be able to discover the presence of named entities in targeted *Twitter* stream before we could categorize their types. Furthermore, even without categorizing the types of named entities, *TwI*NER already enable us to make early crisis response. Moreover, as a targeted Twitter stream is constructed for a particular information need, we assume that the user who constructs the stream has the background knowledge in interpreting the named entities detected. In the following subsections, we give an overview of *TwI*NER.

Figure 6.1 shows the general system architecture of *TwI*NER, which has two main components, namely *tweet segmentation* and *segment ranking*.

6.1.1 Tweet Segmentation

As shown in Table 6.1, traditional linguistic features (e.g., capitalization) are unreliable in tweets. Is there any other feature in tweets we can utilize for the task of NER?

In the same examples in Table 6.1, people spell “yio chu kang” rather than “chu kang yio” or “kang chu yio”. In other words, the *correct collocation of a named entity* is still preserved in tweets. This observation holds stronger if a larger set of tweets are aggregated together. This motivates us to learn a weak phrase segmenter for tweets first.

Figure 6.2: Example of Tweet Segmentation

The idea is to segment an individual tweet into a sequence of consecutive phrases³, each of which appears “more than chance” [36, 130]. Figure 6.2 gives an example. In this example, after removing the stop words, a tweet “*My shoes are gg to compete in the youth olympic games sailing competition. It just needs a mast and a rudder*” is segmented into seven parts.

More formally, given a tweet of four words $w_1w_2w_3w_4$, we segment it as $w_1w_2||w_3w_4$ rather than $w_1||w_2w_3w_4$, if $\mathcal{C}(w_1w_2) + \mathcal{C}(w_3w_4) > \mathcal{C}(w_1) + \mathcal{C}(w_2w_3w_4)$, where $\mathcal{C}(\cdot)$ basically captures *the probability being a valid phrase* of a segment.

A straightforward idea of computing $Pr(\cdot)$ is to count a segment’s appearance in a very large corpus. The ideal case is that we use the entire collection of tweets published in *Twitter* to compute the $Pr(\cdot)$ for all possible segments. Unfortunately, to the best of our knowledge, such corpus never exists. Instead, we turn to Microsoft Web N-Gram corpus⁴ [146]. This N-Gram corpus is based on all the documents in the World Wide Web indexed by Microsoft Bing in the EN-US market; it provides a good estimate of the statistics of commonly used phrases in English.

Another idea of computing $Pr(\cdot)$ is to look up the segments in a knowledge base where valid segments are more easily recognized. We exploit *Wikipedia* for this purpose, which is by far the largest online encyclopedia in the World Wide Web. We take a snapshot

³A segment basically contains a phrase. In the rest of this work, “segment” and “phrase” are used interchangeably.

⁴<http://web-ngram.research.microsoft.com/info/>

of English *Wikipedia*⁵, and build a dictionary by extracting all the article titles, disambiguation pages, redirect pages (synonyms), and wikilinks [85]. If a segment matches any entry in the dictionary, it has a higher prior probability of being a true named entity.

TwiNER combines both ideas in a dynamic programming algorithm to efficiently test various segmentation combinations. Note that in this step, we do not use any local linguistic features of a segment, such as its capitalization. Instead, we leverage on the World Wide Web to derive the segmentation. For ease of presentation, information captured from the World Wide Web for a given segment is called its *global context*.

6.1.2 Segment Ranking

Each segment extracted in Step (1) is a candidate named entity, e.g. “youth olympic games” and “mast rudder”. We now have a huge pool of candidate named entities. Undoubtedly, this pool has a high recall but a very poor precision in identifying the true named entities. For example, among the seven segments extracted in Figure 1, only “youth olympic games” can be considered as a true named entity.

Can we automatically identify the true entities from non-entities in the pool? To address this problem, we learn a function that assigns a confidence score of being a true named entity to each candidate named entity. Candidate named entities are then ranked according to this score. By setting a threshold, we can easily remove the long tails of non-entities with low scores.

Recall there are two types of global context for a given segment in *TwiNER*: Web N-Gram and *Wikipedia*. The former apparently has no clear clue about such score because many common word combinations with high frequency are not named entities, like “there is” and “such a”. The latter, on the other hand, provides some hints because many named entities either have corresponding *Wikipedia* pages or have been referenced in *Wikipedia*.

⁵<http://dumps.wikimedia.org/enwiki/>

However, *Wikipedia* is not as *real-time* as *Twitter*. It usually takes a while for a new named entity appeared in tweets to be captured by *Wikipedia*. For example, in the tweets we use in the experiments, “Vincent Wijesingha” is the name of a political figure, which appeared in tweets in the early April 2011. Before end of April 2011, there was no mention about this person at all in *Wikipedia*. Furthermore, there is also no guarantee that all named entities in tweets would appear in *Wikipedia* later.

Since the *global context* is insufficient to identify the true named entity, is there any local feature in tweets themselves that we can utilize? It is observed that there exists a *gregarious* property among the named entities in the targeted tweet stream, since the tweets in the targeted tweet stream are normally about similar or related topics/events. Formally, *gregarious* refers to the interaction of named entities with each other and to their collective co-existence in the targeted tweet stream. For example, “Barack Obama” is a named entity. It often co-occurs with other named entities like “United States” and “Michelle Obama” in a targeted stream about *United States*, but seldom co-occurs with “please look”, a valid segment extracted from tweets but a non-entity. It is also uncommon that same set of non-entities appear together often.

This *gregarious* property of named entities in *Twitter* motivates us to design a “recursive” algorithm to compute the score of a segment being a named entity. The idea is: an undirected *segment graph* using all the segments extracted in Step (1) is built first, in which nodes are segments and edges are weighed proportional to the co-occurrence similarity; then, a *random walk model* is applied on this graph to derive the probability of a segment being a named entity. Because a segment’s confidence is affected by its neighbors in the graph, which only depends on the tweets themselves, we call the *segment graph* as the *local context* of a segment in the tweets. Note that, not only has the *local context* been considered in this model, but also the *global context* is integrated to overcome the limitation of random walk model. Finally, the output of the model is used as the score of a segment being a named entity.

One may wonder that building *local context* (i.e. the *segment graph*) defeats the real-time nature of *Twitter*. Indeed, a buffer of tweets is necessary to construct the local *segment graph*, making *TwNER* not completely real time (response in a “tweet by tweet” manner). Nevertheless, there are more than 2,000 tweets generated every second⁶ in *Twitter*, which is already a big enough buffer to build the *local context* in Step (2). Therefore, *TwNER* is able to give “near real-time” response practically (in a “second by second” manner).

6.1.3 Contributions

To sum up, we made the following contributions in this chapter:

1. We proposed an unsupervised NER system without explicit human label efforts. Our system does not rely on any linguistic features, making it suitable for tweets and potentially other social text streams with unreliable linguistic features.
2. To the best of our knowledge, our *TwNER* system is the first to exploit both the *local context* (in *tweets*) and the *global context* (from World Wide Web) together for named entity recognition task in Twitter.
3. The proposed system has been successfully evaluated on two different collections of real-life tweets, simulating two types of targeted twitter streams. A region-based stream for tweets published by users from a particular geographical region; and a topic-based stream for tweets potentially relevant for a political event.

6.2 Tweet Segmentation

In this section, we detail our solution for tweet segmentation. Given an individual tweet $t \in T_i$, the problem of tweet segmentation is to split t into m consecutive segments,

⁶200 million Tweets per day: <http://blog.twitter.com/2011/06/200-million-tweets-per-day.html>.

$t = s_1 s_2 \dots s_m$; each segment contains one or more words. To obtain the optimal segmentation, we use the following objective function, where $\mathcal{C}$ is the function that measures the *stickiness* of a segment or a tweet defined based on word collocation:

$$\arg \max_{s_1, \dots, s_m} \mathcal{C}(t) = \sum_{i=1}^m \mathcal{C}(s_i), \quad (6.1)$$

A high *stickiness* score of segment s indicates that it is not suitable to further split segment s , as it breaks the correct word *collocation*. In other words, a high *stickiness* value indicates that a segment cannot be further split at any internal position.

If the word length of tweet t is l , there exists 2^{l-1} possible segmentations. It is inefficient to iterate all of them and compute their stickiness. We therefore design a dynamic programming algorithm to tackle the problem, which is presented in the following.

6.2.1 A Dynamic Programming Algorithm

Algorithm 1 outlines our dynamic programming algorithm for tweet segmentation. The basic idea is to recursively conduct binary segmentations and then evaluate the *stickiness* of the resultant segments. More formally, given any segment s from t (s can be t itself or a part of t) and suppose $s = w_1 w_2 \dots w_n$, our solution is to conduct a binary segmentation by splitting it into two adjacent segments $s^1 = w_1 \dots w_j$ and $s^2 = w_{j+1} \dots w_n$ by satisfying:

$$\arg \max_{s^1, s^2} \mathcal{C}(s) = \mathcal{C}(s^1) + \mathcal{C}(s^2). \quad (6.2)$$

The complexity of Algorithm 1 is $O(lue \log(ue))$, where l is the average tweet length, u is the upper bound of segment length, and e bounds top sub-segments of a segment. Long segments are rare in tweets because each tweet is limited to 140 characters. We observed that in our data, $u = 5$ is a proper bound as the maximum length of a segment, which largely reduces the number of possible segmentations. We also set $e = 5$ so that the segmentation only focuses on top-quality segments and are not stuck by trivial ones, which leads to a complexity of $O(l)$.

```

input :
    A tweet:  $t = w_1w_2...w_l$ ;
     $u$ : the maximum length of a segment  $s$ ;
     $e$ : top  $e$  segmentations set  $\mathbb{S}$  for each segment  $s$ ;
output:
    An optimal tweet segmentation  $t = s_1s_2...s_m$ ;
for  $i = 1 : l$  do
    initialize a set  $\mathbb{S}_i = \{\}$  to store possible segmentation of segment  $s_i = w_1w_2...w_i$ ;
    if  $i \leq u$  then
        /* do not split  $s_i$  */
        calculate  $\mathcal{C}(s_i)$ ;
        add  $s_i$  to  $\mathbb{S}_i$  as a possible segmentation of  $s_i$ ;
    /* try different possible ways to segment  $s_i$  */
    for  $j = 1 : i - 1$  do
        if  $i - j \leq u$  then
            form two shorter segments of  $s_i$ :  $s_i^1 = w_1...w_j$  and  $s_i^2 = w_{j+1}...w_i$ ;
            calculate  $\mathcal{C}(s_i^2)$ ;
            foreach  $S_j \in \mathbb{S}_j$  do
                /*  $\mathbb{S}_j$  contains the top  $e$  possible segmentations of  $s_i^1$ , and  $S_j$ 
                is one of them; */
                concatenate  $S_j$  and  $s_i^2$  to form a new segmentation  $S$  of  $s_i$ ;
                add  $S$  to  $\mathbb{S}_i$ ;
                 $\mathcal{C}(S) = \mathcal{C}(S_j) + \mathcal{C}(s_i^2)$ 
            Sort  $\mathbb{S}_i$  and keep only the top  $e$  segmentations;
    return  $S \in \mathbb{S}_l$  with the highest score as the optimal segmentation;

```

Algorithm 1: Tweet Segmentation

6.2.2 Segment Stickiness Function

In Algorithm 1, one key factor is the *stickiness* function $\mathcal{C}$. A high *stickiness* score of segment s indicates that further splitting segment s would break the correct word *collocation*. There are a number of *collocation* measurements [93, 110]. However, all these measures were defined for two arguments. That is, they were designed to measure the collocation of the bigram or the n-grams with the particular binary partition. A variety of studies have been conducted to extend these binary collocation measures to the n-grams case (where n is greater than 2) [31, 36, 127]. We define the *stickiness* functions by using the generalization framework proposed in [31]. Specifically, the generalized collocation measures of Point Mutual Information (PMI) and Symmetric Conditional Probability

(SCP) are studied here.

6.2.2.1 PMI based Stickiness

PMI measures the degree that two words occur together more often than by chance. Mathematically, PMI for bigram w_1w_2 is defined as follows:

$$PMI(w_1w_2) = \log \frac{\Pr(w_1|w_2)}{\Pr(w_1)} = \log \frac{\Pr(w_1w_2)}{\Pr(w_1)\Pr(w_2)} \quad (6.3)$$

Given a segment, $s = w_1...w_n$, PMI is then extended by averaging all binary partitions as follows:

$$PMI(s) = \log \frac{\Pr(w_1...w_n)}{\frac{1}{n-1} \sum_{i=1}^{n-1} \Pr(w_1...w_i) \Pr(w_{i+1}...w_n)} \quad (6.4)$$

If segment s only contains one word w , we have $PMI(s) = \log \Pr(w)$.

Note that PMI defined above falls into the range of $(-\infty, +\infty)$. The *stickiness* of segment s is then defined by mapping the value of Equation 6.4 to the range of $[0, 1]$ as follows:

$$\mathcal{C}(s) = \frac{1}{1 + e^{-PMI(s)}} \quad (6.5)$$

6.2.2.2 SCP based Stickiness

Symmetrical Conditional Probability (SCP) was proposed in [31] to measure the “cohesiveness” of bigram w_1w_2 by considering both conditional probabilities for the bigram given each single term:

$$SCP(w_1w_2) = \Pr(w_1w_2|w_1) \Pr(w_1w_2|w_2) = \frac{\Pr(w_1w_2)^2}{\Pr(w_1)\Pr(w_2)} \quad (6.6)$$

Given a segment, $s = w_1...w_n$, SCP of s is defined similarly as follows:

$$SCP(s) = \frac{\Pr(s)^2}{\frac{1}{n-1} \sum_{i=1}^{n-1} \Pr(w_1...w_i) \Pr(w_{i+1}...w_n)} \quad (6.7)$$

Here, we smooth SCP value by taking logarithm calculation, Equation 6.7 is then updated as follows:

$$SCP(s) = \log \frac{\Pr(s)^2}{\frac{1}{n-1} \sum_{i=1}^{n-1} \Pr(w_1 \dots w_i) \Pr(w_{i+1} \dots w_n)}, \quad SCP(s) \in (-\infty, 0) \quad (6.8)$$

Similarly, we define SCP for any segment s of unit length as $SCP(s) = 2 \log \Pr(w)$. We then define the *stickiness* of s by using the sigmoid function as follows:

$$\mathcal{C}(s) = \frac{2}{1 + e^{-SCP(s)}} \quad (6.9)$$

6.2.3 Enhanced Stickiness by World Wide Web

By now, the calculation of the stickiness is reduced to estimating $\Pr(s)$, $\Pr(s^1)$, and $\Pr(s^2)$ for any segment $s \subset t$, which are prior probabilities of segments. To accurately estimate these prior probabilities, we need a large enough corpus as the *global context* of each segment. The ideal global context is the entire collection of tweets published in *Twitter*. But unfortunately, to the best of our knowledge, such corpus is not available.

Instead, we exploit the one provided by Microsoft Web N-Gram Services [146] as approximation. This corpus is based on the web documents indexed by Microsoft Bing search engine in the EN-US market. The spam and other low quality documents are excluded. Each indexed document is parsed, tokenized, and the text is lower-cased with the punctuations removed. No stemming, spelling correction or inflections are performed [146], which provides a large enough English corpus to estimate prior probabilities of segments.

One problem of segmentation based on the lexical statistics derived from such corpus is its preference towards frequent patterns. Figure 6.3 illustrates such an example, with a portion of tweet and three possible segmentations.

If only Web documents are used as a priori knowledge, then “*youth olympic games sailing competition*” would be segmented into “*youth*” and “*olympic games sailing competition*” (*i.e.*, possible segmentation 3 in the figure), because both “*youth*” and “*olympic*

Figure 6.3: Different Segmentations of a Portion of Tweet

games sailing competition” are frequent in Web documents. Nevertheless, this tweet is in fact referring to “Youth Olympic Games” held in Singapore in 2010.

We therefore leverage a knowledge base in the World Wide Web, *Wikipedia*, as another source of *global context* to tackle this problem. There are several reasons for the choice of *Wikipedia*. It provides rich a priori knowledge about entity information and is publicly available. Article titles, references to other *Wikipedia* pages, and the disambiguation pages, have often been used as named entity candidates [75, 121]. In detail, we build a large *Wikipedia* dictionary by extracting from a snapshot of *Wikipedia* on January 30, 2010 all the English article titles, disambiguation pages, redirect pages as well as hyperlinks [85]. Articles isolated from the rest are removed. Finally, we have a *Wikipedia* dictionary of 4,342,732 entries as well as their polysemes, synonyms.

Let $Q(s)$ be the probability that s appears as anchor text in its mentioning *Wikipedia* article, which is the number of *Wikipedia* articles containing s as anchor text divided by the number of *Wikipedia* articles s appears in. A segment appearing as anchor text with a high probability in *Wikipedia* is a strong indication that it is a valid name entity[85]. The *stickiness* function is now defined as follows:

$$C'(s) = C(s) \cdot e^{Q(s)} \quad (6.10)$$

$\mathcal{C}(s)$ is measured by exploiting the *global context* captured in Microsoft Web N-Gram. The second component of Equation 6.10 is introduced so that segments appearing in Wikipedia as anchor texts are attributed with higher *stickiness*.

6.2.4 Length Normalization

So far, we treat segments of various lengths equally. However, in Twitter NER task, there are only a few long named entities. And it is observed that longer valid segments have higher chances of being named entities than shorter ones. Note that the Web N-gram data has already had a strong preference for short segments. Given this, we introduce length normalization $\mathcal{L}(s)$ to Equation 6.10 to favor moderately long segments in both Web N-gram and *Wikipedia*. Finally, we have the following *stickiness* function:

$$\mathcal{C}''(s) = \mathcal{L}(s) \cdot e^{Q(s)} \cdot \mathcal{C}(s) \quad (6.11)$$

The length normalization factor $\mathcal{L}(s)$ is empirically defined as:

$$\mathcal{L}(s) = \begin{cases} \frac{|s| - 1}{|s|}, & \text{for } |s| > 1 \\ 1, & \text{for } |s| = 1 \end{cases} \quad (6.12)$$

6.3 Segment Ranking

Section 6.2 extracts a large number of segments which are valid in the sense of word collocation by leveraging segments' *global context* in the World Wide Web. However, not all segments extracted are named entities. For example, "please look" is a valid segment but not a named entity. In this section, a set of strategies have been developed to rank segments according to their likelihood of being named entities.

6.3.1 Noise Filtering

We first remove three types of segments that are obviously *not* named entities.

1. Segments containing well-known slang words, e.g. “lol” and “tmr”. We use a compilation of Internet slangs provided by <http://www.noslang.com/dictionary/full>.
2. Segments containing words with consecutive repeating characters, e.g. “hahahaha”, “nooooo”, and “goooooood”. These words are frequently used in tweets to represent exaggerative emotions. Regular expressions are used to recognize them.
3. Segments containing words with “#” as prefix. Those words are usually *hashtags* in tweets, which are created by users to highlight keywords/topics. They are not considered named entities in this work, and can be easily removed by locating the “#” prefix.

6.3.2 Local Segment Graph

The *global context* alone is insufficient to recognize a named entity. We therefore utilize the *local context* of a segment in tweets to tackle this problem. One intuitive solution is to weigh a segment using its frequency in tweets. However, this method would wrongly favor phrases like “please look”, which is not only frequent in World Wide Web, but also frequent in Twitter.

As we discussed in Section 6.1, there exists a *gregarious* property among named entities in targeted tweet streams. A good example is “Barack Obama”. It is a true named entity, and it often co-occurs with other named entities like “United States” or “Michelle Obama” in tweets, but seldom co-occurs with “please look”, a valid segment but non-entity. It is also uncommon that same set of non-entities would appear together often.

Based on this property, we propose a “segment graph”. At the i th interval (recall that *TwNER* recognizes named entities in a batch mode), we build an undirected *segment graph* $G(V, E)$ using all segments V extracted from the tweet set T_i on the fly. In this

graph, each node is a valid segment after noise filtering, and the edge $e_{ab} \in E$ between two nodes (segments) s_a and s_b is weighed by the *Jaccard Index*:

$$w_{ab} = w(e_{ab}) = \frac{|M(s_a) \cap M(s_b)|}{|M(s_a) \cup M(s_b)|}, \quad (6.13)$$

where $M(s)$ is the set of tweets in T_i containing segment s .

The *segment graph* $G(V, E)$ provides a good *local context* for each segment in T_i . It does not use the unreliable local linguistics features of tweets but relies on the relations among segments. Because all segments have been parsed once by their *global context* and then filtered with heuristic rules, these relations are relatively more reliable than local linguistics features.

6.3.3 Random Walk Model

A *random walk model* is then applied on graph $G(V, E)$ to compute the stationary probability of each segment being a true named entity, by considering the graph bidirectional. While random walking, the probability of transiting from node s_a to node s_b (denoted as P_{ab}) is given by

$$P_{ab} = \frac{w_{ab}}{\sum_{c \in V} w_{ac}}. \quad (6.14)$$

All transition probabilities are then aggregated to form a nonnegative transition matrix P for the whole graph.

To overcome the “dangling links” while conducting a random walk on graph $G(V, E)$, a teleportation vector $\mathbf{e}$ is also introduced to make the random walker jump from a node to any other node in the *segment graph* with a small probability [108]. We observe that $Q(s)$, the probability that s appears as anchor text in *Wikipedia*, is a good teleportation a priori. In other words, we favor those segments that are valid hyperlinks in *Wikipedia*, i.e. those segments are more likely to be named entities. Accordingly, we define the

following teleportation probability for node (segment) s :

$$\mathbf{e}_s = \frac{Q'(s)}{\sum_{s_j \in V} Q'(s_j)}, \text{ where } Q'(s) = e^{Q(s)}. \quad (6.15)$$

The exponential function is used here to avoid the situation that the segment is *new* to *Wikipedia* so that its $Q(s) = 0$ and will never be teleported to.

The *Wikipedia*-based teleportation can be considered as an injection of global context into the random walk model of the local context. With the transition matrix P and the *Wikipedia*-based teleportation vector $\mathbf{e}$, the stationary eigenvector π^T of P is calculated iteratively using power method as below:

$$\pi = (\gamma P^T + (1 - \gamma)\mathbf{e}\mathbf{1}^T)\pi, \quad (6.16)$$

where γ controls the probability of teleportation. The lower γ is, the higher probability the random walk will teleport according to $\mathbf{e}_s$. Then, π^T is used as probabilities of segments being named entities in the local context.

Finally, for balancing the advantages of *global context* and *local context*, given an input segment s , *TwiNER* multiplies its stationary probability $\pi(s)$ in the *segment graph* mainly learned from the local context with its teleportation probability learned from *Wikipedia*:

$$y(s) = \mathbf{e}_s \cdot \pi(s). \quad (6.17)$$

Equation 6.17 is used to rank all segments, and only the top K segments are retained as named entities.

6.4 Evaluation

In this section, we conduct extensive experiments to evaluate *TwiNER*. In Section 6.4.1, datasets simulating two targeted twitter streams are described, and performance metrics are introduced. Section 6.4.2 compares *TwiNER* with existing methods. We then present a performance analysis of *TwiNER* in different settings in Section 6.4.3.

6.4.1 Tweets Data and Performance Metrics

Tweets collections. Two collections of tweets are used in the experiments to simulate targeted twitter streams.

The first collection (*SIN*) was collected to simulate targeted twitter stream of one particular geolocation by monitoring a number of Singapore-based⁷ users' tweets published in June 2010. The set of users to be monitored was populated by first getting the top-1000 Singapore-based *Twitter* users with the most number of followers from <http://twitaholic.com>, and then expanding the list by including the top users' followers and friends in *Twitter* within two hops. There are a number of real-life events in the data collection period, such as the flash flood in Orchard Road (a premium shopping belt in Singapore), FIFA World Cup 2010 and WWDC 2010, etc. Collection *SIN* contains 4,331,937 tweets.

The second collection (*SGE*) was collected to simulate targeted twitter stream of one particular event by monitoring a set of pre-defined keywords related to Singapore General Election 2011. Similar to collection *SIN*, only tweets published by Singapore-based users were collected. The data collection started on April 13, 2011 and ended on May 13, 2011, which covered the duration of Singapore General Election 2011 (nomination day on April 27, 2011, and polling day on May 7, 2011). Collection *SGE* contains 226,744 tweets.

It is observed that, by collecting tweets based on users, topics covered in collection *SIN* are diverse in nature. Topics covered in collection *SGE*, on the other hand, are more concentrated since most of the discussions are about the general election. Another observation is that, twitter users are more formal in political discussions than casual discussions. In other words, tweets in *SGE* are more formal than those in *SIN*.

Manual Annotation. For both collections, 5,000 tweets are randomly sampled from the tweets published on one random day. Each tweet is then labeled by two human anno-

⁷A user is considered Singapore-based if she specifies Singapore in the *location* field of her profile.

tators, who have strong background knowledge about Singapore-related named entities. The BILOU schema is used [88, 118]. After discarding retweets and tweets with inconsistent annotations, we get 4,422 tweets for *SIN* and 3,328 tweets for *SGE*. We denote these two randomly sampled tweets collections with groundtruth labeling as *SIN_g* and *SGE_g* respectively. Observe that the annotating agreement is relatively low for collection *SGE*. This is mainly due to the disagreement between the human annotators about the way how to handle the concept *GRC*⁸ and *SMC*⁹, which refer to different types of electoral divisions in Singapore. Annotators did not have an agreement on whether a GRC/SMC should be labeled as part of a location name. For example, some may label “*aljunied grc*” as “*<U>aljunied</U> grc*”, while some may label as “*aljunied <L>grc</L>*”.

Performance Metric. Performance metrics used throughout the experiments include: Precision(*Prec*), Recall(*Recall*), and *F1*. *Prec* quantifies the percentage of the extracted phrases that are true named entities. *Recall* quantifies the percentage of the true named entities that are correctly recognized. *F1* is the harmonic mean of *Prec* and *Recall*, *i.e.*,
$$F1 = 2 \cdot \frac{Prec \cdot Recall}{Prec + Recall}.$$

Note that different values of *K* (the parameter in the *segment ranking* step) would result in different performance of *TwI_{NER}*: larger *K* will increase *Recall* but decrease *Prec*, and vice versa. *TwI_{NER}*’s performance reported in the following sections are the ones with the highest *F1* score *TwI_{NER}* can achieve using SCP-based *stickiness* function, and various values of *K*. For a fair comparison, *K* is set to be larger than 50 (*i.e.*, $K > 50$). The maximum iteration for the random walk is fixed at 500.

⁸http://en.wikipedia.org/wiki/Group_Representation_Constituency

⁹http://en.wikipedia.org/wiki/Single_Member_Constituency

6.4.2 Comparison with Other Methods

In this section, we compare *Twiner* with two conventional NER systems trained on tweets. Specifically, we train Stanford-NER and LBJ-NER with the labeled tweet data and evaluate their performance¹⁰. Moreover, we also compare with a tweet-specific NER system (T-NER)¹¹ proposed in [122] on the two tweet collections.

- LBJ-NER: A NER system based on the regularized averaged perceptron approach, which uses gazetteers extracted from *Wikipedia*, word class models derived from unlabeled text, and expressive non-local features [118]. It is reported to have achieved the best result (F1-Measure of 0.908) on the CoNLL 2003 test set.
- Stanford-NER: A NER system based on CRF model which incorporates long-distance information [40]. It achieves good performance consistently across different domains.
- T-NER: a supervised NER system uses CRF model for learning and inference. A set of widely-used effective features are used in T-NER, including orthographic, contextual, and dictionary features [122].

Note that, other than the proposed *Twiner*, the three methods listed above (*i.e.*, LBJ-NER, Stanford-NER and T-NER) are supervised methods and require labeled examples. For performance comparison, we randomly split both *SIN_g* and *SGE_g* in the ratio of 6 : 4 as training and evaluation sets. Stanford-NER and LBJ-NER are trained with default feature settings¹². While LBJ-NER requires development set for the parameter tuning, we further split the training set in the ratio of 5 : 1 for training and development. All the methods are evaluated on the same evaluation set.

¹⁰Note that as the entity type classification is not the focus of this work, we do not differentiate the entity types when both LBJ-NER and Stanford-NER are trained.

¹¹https://github.com/aritter/twitter_nlp

¹²We use the same settings for training as in the CoNLL-2003 shared task. <http://www.cnts.ua.ac.be/conll2003/ner/>

Table 6.2: Different NER systems' performance on tweets

System	Dataset	Prec	Recall	F1
LBJ-NER	<i>SGE_g</i>	0.933	0.595	0.727
Stanford-NER	<i>SGE_g</i>	0.950	0.880	0.913
T-NER	<i>SGE_g</i>	0.713	0.359	0.478
TwiNER	<i>SGE_g</i>	0.929	0.660	0.772
LBJ-NER	<i>SIN_g</i>	0.764	0.265	0.393
Stanford-NER	<i>SIN_g</i>	0.762	0.293	0.423
T-NER	<i>SIN_g</i>	0.429	0.509	0.466
TwiNER	<i>SIN_g</i>	0.419	0.329	0.419

Table 6.2 shows the evaluation results of different NER systems. It can be observed from Table 6.2 that:

1. As observed in Table 6.2, the overall performance on *SGE_g* is much better than on *SIN_g* by all methods. For the case of LBJ-NER and Stanford-NER, the main reason is the adverse impact of error-prone local context of tweets, since *SIN_g* has more tweets of informal style. This results in a low quality of training set based on the linguistic features. *TwiNER* extracts named entities by exploiting the *local context*. Since *SIN_g* is collected by monitoring twitter users of Singapore, diverse topics are collected. This makes *gregarious* property relatively weaker in this collection, which results in the relatively lower performance of *TwiNER* on *SIN_g*.
2. T-NER performs consistently across the two evaluation sets. Since it was developed by taking into consideration tweets' error-prone context, it outperforms Stanford-NER and LBJ-NER and achieves the best performance on *SIN_g* in terms of *F1*. However, the decrease of *Prec* on *SIN_g* compared to that of *SGE_g* ($0.713 \rightarrow 0.429$) indicates that T-NER still suffers a lot from tweets' error-prone nature.
3. No system outperforms the others on both collections. Stanford-NER performs much better than the other systems on *SGE_g*, with *F1* performance more than

0.91. However, it only achieves a slightly better $F1$ performance than LBJ-NER and *TwiNER* on *SIN_g*. LBJ-NER performs the worst on *SIN_g* and the third best on *SGE_g* in terms of $F1$. Moreover, the supervised system evaluated here, Stanford-NER, LBJ-NER, and T-NER, achieve relatively lower *Recall* across both collections. This indicates that supervised methods relying on local linguistic features may not generalize well for tweet’s error-prone and dynamic nature.

4. *TwiNER*, an unsupervised approach, achieves comparable $F1$ performance with the other supervised systems. *TwiNER* performs much better than LBJ-NER and T-NER on *SGE_g*. It also outperforms LBJ-NER and achieves a comparable performance with Stanford-NER on *SIN_g* in terms of $F1$, where the *gregarious* property is relatively weak.

As this set of experiments show, performance of LBJ-NER and Stanford-NER deteriorate significantly when they are applied on tweets, due to tweets’ error-prone context. To further illustrate the adverse impacts of tweets’ error-prone context, we trained LBJ-NER and Stanford-NER with formal text corpus using the CoNLL-2003 shared task data, and then directly apply the trained models on the evaluation sets of *SIN_g* and *SGE_g*. To distinguish the methods trained using tweet data, we denote the two methods trained on formal text LBJ-NER^F and Stanford-NER^F respectively.

The results of LBJ-NER^F, Stanford-NER^F and *TwiNER* are summarized in Table 6.3. It is observed that the performance of LBJ-NER^F and Stanford-NER^F deteriorate significantly with $F1$ -measure of lower than 0.5 on both *SIN_g* and *SGE_g*.

6.4.3 Analysis of *TwiNER*

In this section, we investigate the impact of different *TwiNER* components on its performance.

Table 6.3: Conventional NER systems’ performance on tweets

System	Dataset	Prec	Recall	F1
LBJ-NER ^F	<i>SGE_g</i>	0.674	0.400	0.502
Stanford-NER ^F	<i>SGE_g</i>	0.691	0.642	0.666
TwिNER	<i>SGE_g</i>	0.929	0.660	0.772
LBJ-NER ^F	<i>SIN_g</i>	0.314	0.314	0.314
Stanford-NER ^F	<i>SIN_g</i>	0.256	0.461	0.329
TwिNER	<i>SIN_g</i>	0.419	0.329	0.419

6.4.3.1 Performance of Tweet Segmentation

Tweet segmentation is used to extract the named entity candidates from tweets, or in other words, to identify the correct boundary of potential named entities in tweets. It is a critical component because the performance of *TwिNER* is heavily affected by the effectiveness of tweet segmentation.

Two *stickiness* functions are defined by using two collocation measures, PMI and SCP, for tweet segmentation. The tweet segmentation algorithm described in Section 6.2 also incorporates an external knowledge base *Wikipedia*. Further, we normalize the segment length to favor long named entities. In this section, we study the impact of the collocation measures (*PMI* or *SCP*), the *Wikipedia* dictionary (*Wiki*), and the length normalization (*Norm*), based on the ground truth in *SIN_g* and *SGE_g*. We use tweet segmentation with only PMI or SCP measures as the baseline (Equation 6.5 and 6.9). We measure the percentage of named entities that are correctly extracted (i.e. split as a segment) as the performance metric, which is denoted as *Prec* as well. The experimental results are listed in Table 6.4. From Table 6.4, we observe that:

1. *SCP* significantly outperforms *PMI* for tweet segmentation. We believe this is because PMI returns disproportionately high values for frequent items. This property makes PMI prefer longer segments, which is confirmed by manual investigation of the segmentation result. For example, we observe that “*sdp*” cannot be extracted

from “*vote sdp*” by PMI, since PMI-based *stickiness* returns 0.397 for “*vote sdp*”, and only 0.017 and 0.004 for “*vote*” and “*sdp*”. While SCP-based *stickiness* returns only $5.76E - 5$ for “*vote sdp*”, and $6.51E - 4$ for “*vote*+“*sdp*”.

2. Length normalization (*Norm*) is effective and improves the accuracy of tweet segmentation in the two collections for SCP-based *stickiness*. For example, given a long segment like “*java programming language*”, *SCP+Norm* is able to fairly treat it as a named entity, while *SCP* split it as “*java*” and “*programming language*”. Since PMI prefers longer segments, further preference introduced by *Norm* does aggravate the problem.
3. *Wikipedia*’s broad coverage and high quality knowledge help reclaim incorrect decisions made by the lexical statistics or reinforce the correct decisions. For instance, *SCP* cannot extract “*pap*” from “*pap team*” given the latter is a frequent phrase. With the priori knowledge from *Wikipedia*, where “*pap*” has a $Q(s)$ value of 0.181, “*pap*” is successfully extracted by *SCP+Wiki*.
4. The combination of *Wikipedia* dictionary and length normalization further boosts up the performance of tweet segmentation of SCP-based *stickiness*. This indicates that *Wiki* and *Norm* are complementary to each other. For example, we observe a tweet part “*pap give free iphones*” in a tweet. All of *SCP*, *SCP+Wiki*, and *SCP+Norm* fail to split “*free*” and “*iphones*” apart due to the frequent usage of “*free iphones*” in web documents. However, by combining the both *Wiki* and *Norm*, “*iphones*” is successfully extracted. Also, positive improvement from *PMI+Norm+Wiki* compared to *PMI+Norm* shows the ability of reclaiming incorrect decisions by exploiting *Wikipedia* dictionary.

Table 6.4: Impacts of Tweet Segmentation by SCP

Scheme	Prec@ <i>SIN</i> _{<i>g</i>}	Prec@ <i>SGE</i> _{<i>g</i>}
SCP	0.721	0.830
SCP+Norm	0.737	0.861
SCP+Wiki	0.739	0.841
SCP+Norm+Wiki	0.758	0.874
PMI	0.317	0.288
PMI+Norm	0.297	0.288
PMI+Wiki	0.344	0.319
PMI+Norm+Wiki	0.332	0.308

6.4.3.2 Impact of Random Walk on Segment Ranking

A *random walk* model is applied to exploit the *gregarious* property of named entities in tweets. The final segment ranking output is an aggregation from the stationary probability of the *random walk* model (*local context*) and the segment’s *Wikipedia*-based teleportation *Wiki* probability (*global context*). We analyze their impact on the performance of segment ranking in this section. Specifically, we investigate the following schemes for segment ranking:

- *MFS*: A naive method that ranks the segments based on their frequency in the collection. That is, the most frequent segments are ranked higher.
- *Wiki*: A naive method that ranks the segments based on their *Wikipedia*-based teleportation probability.
- *RW*: A simple random walk with uniform teleportation. The segments are then ranked based on the stationary probability $\pi(s)$.
- *RWW*: A random walk with *Wikipedia*-based teleportation. The segments are then ranked based on the stationary probability $\pi(s)$.
- *RWW + Wiki*: A random walk with *Wikipedia*-based teleportation, while the segments are ranked based on Equation 6.17.

Table 6.5 lists the experimental results based on the ground truth in SIN_g and SGE_g . From Table 6.5, it can be seen that:

1. While MFS works considerably well on SGE_g , its performance degrades significantly on SIN_g . The tremendous difference in performance is mainly due to the difference between the nature of the two collections SGE_g and SIN_g . As discussed earlier, SGE_g are tweets published in one day during Singapore General Election 2011. Thus, most of the tweets are about the election and related named entities frequently appear in these tweets. Thus, MFS achieves decent performance. However, MFS hardly recognizes any named entities due to the diverse topics covered on SIN_g . This also reflects the degrees of *gregarious* property of the two collections.
2. RW outperforms MFS significantly on SGE_g in terms of $F1$. This validates our assumption that named entities are more likely to co-occur with one another than with non-entities. Since SIN_g is very diverse, *gregarious* property is too weak to improve the segment ranking. RWW significantly outperforms MFS and RW in terms of $Prec$ and $F1$ in SIN_g and SGE_g . The *Wikipedia*-based teleportation priori positively influences the random walk process.
3. An impressive performance is achieved by *Wiki*. It obtains the second best performance in both of the two collections, in terms of $F1$ measure. $RWW + Wiki$ achieves the best performance in both SIN_g and SGE_g in terms of $F1$. We can see that it obtains a large improvement for $Prec$. We believe that the usage of *Wikipedia*-based priori in Equation 6.17 is the main contributing factor for the improvement of $Prec$ compared to *Wiki*. Furthermore, the random walk with *Wikipedia*-based teleportation boost up many named entities that may not be covered by *Wikipedia*, by leveraging the co-occurrence of local context. This results in further improvement of $Prec$.

Table 6.5: Impacts of *local context*, *global context* and *random walk* on segment ranking

Scheme	Prec@ <i>SIN_g</i>	Recall@ <i>SIN_g</i>	F1@ <i>SIN_g</i>
MFS	0.039	0.753	0.074
Wiki	0.433	0.398	0.415
RW	0.039	0.752	0.074
RWW	0.048	0.336	0.084
RWW+Wiki	0.576	0.335	0.423
	Prec@ <i>SGE_g</i>	Recall@ <i>SGE_g</i>	F1@ <i>SGE_g</i>
MFS	0.736	0.565	0.639
Wiki	0.738	0.688	0.712
RW	0.856	0.536	0.659
RWW	0.985	0.536	0.694
RWW+Wiki	0.929	0.646	0.762

Figure 6.4: Precision of top K named entities

6.4.3.3 Impact of K Value

Twiner's performance is related to the choice of K . Larger K will increase *Recall* and decrease *Prec*, and vice versa. In reality, what matters is rather how many real named entities are in the top list, so that the user can gain direct understanding on what the targeted tweets/Twitter users are concerning about. Thus, we calculate *Prec@K* for each K value from 1 to 200¹³. Here, *Prec@K* is the percentage of top K segments returned by *Twiner* that are real named entities. Figure 6.4 shows *Prec@K* curves of *Twiner*

¹³We believe $K \leq 200$ is a good range for the tweet collections we studied here.

on *SGE_g* and *SIN_g*. Major proportion of segments returned when $K < 50$ are true named entities. *TwiNER* achieves a stable *Prec* performance when K is in the range of $[50, 100]$. After 100, $Prec@K$ starts to degrade slowly on *SGE_g*, while $Prec@K$ is still stable at around 0.5 on *SIN_g*. A good strategy of choosing the suitable K value in various scenarios is planned for future work. Nevertheless, it is observed that the choice of K value depends on the nature of targeted tweet streams, such as topic cohesiveness, *gregarious* property, and size of the tweet collections.

Based on the extensive experiments conducted above, we see that by incorporating the encoded intelligence of World Wide Web and local context of tweets, *TwiNER* shows a promising performance. It provides an unsupervised approach for named entity recognition for Twitter, especially for targeted tweet streams with high *gregarious* property.

6.5 Summary

Twitter, as a new type of social media, has attracted great interests from both industry and academia. Many private and/or public organizations have been reported to monitor *Twitter* stream to collect and understand users' opinions about the organizations. Nevertheless, it is practically infeasible and unnecessary to listen and monitor the whole *Twitter* stream, due to its extremely large volume. Therefore, targeted *Twitter* streams are usually monitored instead. Targeted *Twitter* stream is usually constructed by filtering tweets with user-defined selection criteria. There is also an emerging need for early crisis detection and response with such target stream.

Nevertheless, the error-prone and short nature of *Twitter* has brought new challenges to named entity recognition. In this chapter, we present a NER system for targeted *Twitter* stream, called *TwiNER*, to address this challenge. Unlike traditional methods, *TwiNER* is unsupervised. It does not depend on the unreliable local linguistics features. Instead, it aggregates information garnered from the World Wide Web to build robust

local context and *global context* for tweets. Experimental results show promising results of *Twiner*. It is also shown to achieve comparable performance with the state-of-the-art NER systems in real-life targeted tweet streams.

Despite its promising results, there is still space for improvement. First of all, we plan to study *Twiner*'s performance in a larger scale. Second, we plan to study the strategy to identify suitable K value. Last but not least, this chapter does not address the problem of entity type classification. As discussed earlier, this is because we feel this problem is not as pressing as the problem to correctly locate and recognize presence of named entities in tweets, which existing methods largely fail. Extension of *Twiner* for entity type classification is also planned for future work.

Chapter 7

Segment-based Event Detection from Tweets*

The art of discovering the causes of phenomena, or true hypothesis, is like the art of decyphering, in which an ingenious conjecture greatly shortens the road.

– Gottfried Wilhelm Leibniz

In Chapter 6, we propose an unsupervised named entity recognition approach for targeted twitter streams. Besides the business value for enterprises and institutions for market decision making, ordinary users often use Twitter to detect and track the hot events happening around the world. As a result, Twitter has become one of the top-10 most visited website on the Internet¹. As of March 2012, there are more than 140 million active users with over 340 million tweets published a day². However, given the extraordinary large volume of data and users involved in Twitter, timely access to the latest events would be a challenging and tedious activity. Hence, an automatic event detection system would be desirable for normal users and enterprises around the world. However, considering the noisy content and diverse topics dynamically evolving, the traditional algorithms (based on bag-of-word model) for event detection on the formal text stream

*This chapter is based on the paper *Twevent: Segment-based Event Detection from Tweets* by Chenliang Li, Aixin Sun, Anwitaman Datta, published in *ACM CIKM 2012*.

¹<http://www.alexa.com/siteinfo/twitter.com>

²<http://blog.twitter.com/2012/03/twitter-turns-six.html>

(*e.g.*, news stream) obtain the low performance in accuracy as well as the uninterpretable event summarization. We observe that the tweet segmentation algorithm proposed in Section 6.2 help discover more concrete information unit (*e.g.*, named entities) instead of bag-of-word model containing ambiguous words and misspellings. Inspired by the tweet segmentation applied in *TwNER*, we propose a segment-based event detection system for twitter streams in this chapter. The utilization of tweet segments help largely reduce the noisy information contained in tweets. Furthermore, we leverage the semantic resource encoded within Wikipedia to help alleviate the adverse impact of diverse and dynamic topics of tweets.

The rest of the chapter is organized as follows. After the introduction in Section 7.1, we describes *Twevent* and its components in detail in Section 7.2. Section 7.3 presents the experimental results. We conclude this chapter in Section 7.4.

7.1 Introduction

In Twitter, each user becomes an individual news media that not only absorbs/assembles information (such as breaking news), but also publishes/propagates opinions, sentiments, and stories of themselves [81, 140]. The message unit, called a *tweet*, is limited to maximum 140 characters in length. Such a concise unit enables information updates at extremely low-cost and in realtime, making Twitter a timely fresh information resource [35]. Consequently, the intensive interaction between users in realtime enables timely event detection by monitoring tweet updates and many prominent events are timely spotlighted by Twitter users. For example, a record number of tweet updates per second was set within a 30-sec period after the **2010 FIFA World Cup** match between Japan and Cameroon on June 14, 2010. Three days later, the record was broken right after the Lakers' victory in the **2010 NBA Finals** on June 17, 2010³. Other than sports related

³<http://bits.blogs.nytimes.com/2010/06/18/sports-fans-break-records-on-twitter/>

events, it has been reported that earthquake detection based on Twitter is faster than the detection based on traditional media [124]. Moreover, event detection from tweets would help us gain timely understanding of users' opinion/sentiment with respect to the detected events, making it possible for company/organization to take a fast response to any emerging crisis. Event detection from Twitter stream would also contribute to the study of mass communication by analyzing the types of events general users are mostly interested in [155] as well as the reactions by users at different geographical regions [113].

Event detection from Twitter stream is challenging for at least three reasons: short and noisy content, diverse and fast changing topics, and large data volume. The task of event detection has been intensively studied in the past mostly on formal texts, *e.g.*, news articles, blog posts, or academic papers [78, 42, 41, 55]. However, tweets are significantly different from well written texts because of its brevity and informal writing style. According to the principle of least effort [169], people are used to communicate information with the least context, especially in the situation where a short message with free style is allowed. This makes tweets contain a lot of misspellings and informal abbreviations [122]. Because of the noise and shortness, direct adoption of most existing approaches developed for formal texts (*e.g.*, clustering bursty features with co-occurrence measure [42, 55]) is doomed to fail on Twitter streams.

Tweets cover very diverse topics and about half of the tweets are not event-related according to a study by PearAnalytics [109]. They manually categorized 2,000 tweets into six categories: news (3.6%), spam (3.75%), self-promotion (5.85%), pointless babble (40.55%), conversational (37.55%) and pass-along value (8.7%). The numbers indicate the percentage of the tweets in each category. Based on their analysis, about 50% (*i.e.*, spam, self-promotion, pointless babble) of tweets are not related to events. Similar observations are also made in our pilot study of the tweets data used in our experiments. However, a large number of features would be expected being bursty from tweets of

pointless babble category. Obviously, none of these bursty features would help in detecting any event, but would mislead the event detection algorithm and also incur unnecessary computational cost. The situation would be further exaggerated with the fast changing topics in tweets. For example, many users would discuss about a football match during the match or within a few hours right after the match but not for a few days.

7.1.1 Overview of Twevent

To address the above challenges, we present *Twevent*, a novel segment-based event detection system for tweets. One novel feature of *Twevent* is to use the notion of *tweet segment* instead of unigram to detect and describe events. A tweet segment is one or more consecutive words (or phrase) in a tweet message. We observe that tweet segments contained in a large number of tweets are likely to be named entities (*e.g.*, **Steve Jobs**) or some semantically meaningful unit (*e.g.*, **Argentina vs Nigeria**). Therefore, a tweet segment often contains much more specific information than any of the unigrams contained in the segment. The use of tweet segment instead of unigrams therefore greatly reduces the noise in the event detection process and also makes the event detected much easier to be interpreted. For example, *Twevent* detected an event with the following five segments [**south korea, greece, korea vs greece, korea won, korea**] on 12 June 2010; the event is self-explanative. Another novel feature of *Twevent* is the utilization of external knowledge base in guiding the event detection process. In the following, we brief the main steps in *Twevent* for event detection from tweets.

Given tweets published in a Twitter stream, *Twevent* firstly segments each individual tweet into a sequence of consecutive phrases (*i.e.*, segments). Then bursty segments are identified by modeling the frequency of a segment as a Gaussian distribution based on predefined fixed time-window (*e.g.*, a day or an hour). To detect events attracting a larger number of users, we also utilize user frequency (or user support) of the tweet

segments to identify the event-related bursty segments, called event segments. After that, we apply an efficient clustering algorithm to group event-related segments as candidate events, which requires only a single pass through each pair of event segments. To compute the similarity between a pair of event segments, we consider the frequency distribution and the content of the tweets containing each of the tweet segments published within the time-window. The result of event segment clustering is a set of candidate events detected in that time window. The knowledge encoded in Wikipedia is then harnessed to help us figure out the realistic events detected from the trivial ones and to derive the most representative segments for describing the realistic events. As the result, each event detected by *Twevent* is represented by a ranking list of segments including many named entities for easy interpretation.

Twevent holds several features to address the challenges of event detection from tweets. Tweet segmentation employed in *Twevent* identifies informative phrases that reduces noise in further processing. The use of user frequency in bursty event segment extraction makes *Twevent* robust to the negative impact of the tweets of *Spam* and *Self-Promotion*. The external knowledge base offers *Twevent* the ability to resist the adverse impact of diverse and dynamic topics of tweets, such as tweets of *Pointless Babble*, and derive interpretable event descriptions. Lastly, *Twevent* is efficient and scalable by utilizing only the frequency of segments for bursty segment extraction and non-iterative clustering algorithm.

We evaluated *Twevent* with more than 4.3 million tweets published by Singapore-based users over a one month period. In our experiments, *Twevent* achieves much better performance compared to the state-of-the-art method [151] in terms of both precision and recall. More specifically, *Twevent* achieves a precision of 86.1%, and a recall of 75 distinct events detected from the one-month data. Our experimental results also demonstrate the effectiveness of using tweet segments compared to the same detection

Figure 7.1: Segment-based Event Detection System Architecture

process using unigrams. To illustrate that the events detected by *Twevent* often contain named entities or convey concise information, we list the most newsworthy segments detected by *Twevent* in Table 7.3 as part of the experimental results.

7.2 Twevent

In this section, we present a feature-pivot event detection framework. Illustrated in Figure 7.1, our framework consists of three main components: *tweet segmentation*, *event segment detection*, and *event segment clustering*. After receiving a tweet from a Tweet stream, tweet segmentation component splits the tweet into non-overlapping segments. A tweet segment can be either a unigram or multi-gram (*e.g.*, [mtv movie awards], [steve jobs]), and each segment may or may not represent a semantic unit. The resultant tweet segments obtained from a tweet, together with the content and timestamp of the tweet, are indexed in the segment index. The event segment detection component detects abnormal bursty segments by considering tweets frequency distribution and user frequency of the segments. The event segments about the same event are then grouped together to form the event by the event clustering component. In the rest of this section, we describe each component in detail following the order of their usage in our framework.

7.2.1 Tweet Segmentation

The notion of *tweet segment* is described in Section 6.2 for named entity recognition. In this chapter, we apply the tweet segmentation with *SCP*-based stickiness function

defined in Equation 6.9. Given a tweet $d \in \mathcal{T}$, tweet segmentation is to split d into m non-overlapping and consecutive segments, $d = \langle s_1 s_2 \dots s_m \rangle$, where a segment s_i is either a word (or unigram) or a phrase (or multi-gram). The tweet segmentation defined in Algorithm 1 can be finished efficiently in linear time with dynamic programming.

7.2.2 Event Segment Detection

One salient characteristic of emerging events in text streams is that there is a significant coverage of topics related to an event within a certain time period. Accordingly, given a collection of segments of the tweets published within a fixed time window, bursty segments in terms of frequency would be potentially related to some hot events talked and shared by Twitter users. However, considering the dynamic nature and the large volume of tweets published everyday, efficiently detecting bursty segments is non-trivial.

Let N_t denote the number of tweets published within time-window t from Twitter stream, $f_{s,t}$ be the number of tweets containing s published within t , *i.e.*, the tweet frequency of segment s in time-window t . The probability of observing frequency $f_{s,t}$ of segment s in t can be modeled by a binomial distribution [42].

$$P(f_{s,t}) = \binom{N_t}{f_{s,t}} p_s^{f_{s,t}} (1 - p_s)^{N_t - f_{s,t}} \quad (7.1)$$

where p_s is the expected probability of tweets that contain segment s in a random time window. Given that N_t is very large in the case of Twitter stream, it is reasonable to approximate $P(f_{s,t})$ with Gaussian distribution:

$$P(f_{s,t}) \sim \mathcal{N}(N_t p_s, N_t p_s (1 - p_s)). \quad (7.2)$$

Thus, given segment s , the expected number of tweets containing s would be $E[s|t] = N_t p_s$. The more the additional tweets containing s with respect to $E[s|t]$, the more bursty the segment is. On the other hand, segment s with frequency $f_{s,t} \leq E[s|t]$ is considered as a non-bursty segment and will not be considered for further processing. Hence, we define bursty segment as follows.

Definition 7.1 [Bursty Segment] *A segment s is a bursty segment in time window t if its tweet frequency $f_{s,t} > E[s|t]$.*

Next we transfer the frequency of a bursty segment into range of $(0, 1]$ indicating its *bursty probability*.

We consider a bursty segment s to be extremely bursty and assign $P_b(s, t) = 1$ if its tweet frequency $f_{s,t} \geq E[s|t] + 2\sigma[s|t]$, where $\sigma[s|t] = \sqrt{N_t p_s(1 - p_s)}$ is the standard deviation based on Equation 7.2. For a bursty segment whose tweet frequency $f_{s,t}$ falls within the range $(E[s|t], E[s|t] + 2\sigma[s|t])$, we use the following equation to compute its bursty probability.

$$P_b(s, t) = \mathcal{S} \left(10 \times \frac{f_{s,t} - (E[s|t] + \sigma[s|t])}{\sigma[s|t]} \right) \quad (7.3)$$

where $\mathcal{S}(\cdot)$ is the sigmoid function, and a constant 10 is introduced in the equation because the sigmoid function $\mathcal{S}(x)$ smooths reasonable well for x in the range of $[-10, 10]$.

With the above statistical method, we are able to detect the bursty segments and assign each a bursty probability. However, Twitter is significantly different from most text streams (*e.g.*, news stream and blog stream) that have been extensively studied in the literature for bursty feature/event detection, because of its informal writing style and topic diversity. Therefore, a large number of tweet segments would be detected to be bursty segments. A simple statistics in our study shows that the number of distinct bursty segments is about 75% of the number of distinct tweets in a randomly chosen time window. Among the bursty segments detected, many contain misspelling words and informal abbreviations. These noisy bursty segments would not only incur unnecessary computational cost but also hurt the event detection accuracy in the further processing. We therefore source for the wisdom of the crowds to filter the bursty segments.

Instead of solely relying on the tweet frequency of a segment, we believe that a bursty segment has a higher chance to be related to an event if there are more users post tweets

containing the segment. Hence, we define **user frequency** $u_{s,t}$ of a segment s , which is the number of users who post tweets containing s during the time period t .

With the two factors, bursty probability and user frequency, the most simple approach to detect the event-relatedness of a bursty segment is to take the product of the two factors. However, this simple approach would make the top-ranked segments dominated by the ones used by most users, such as "i'm", "i'll", and "guys". To some extent, bursty segments with higher user frequencies are correlated with some events. However, considering the limited length of tweets, the bursty segments with higher user frequencies may not be semantically meaningful and are often ambiguous. For instance, "nigeria", "argentina" and "argentina vs nigeria" are all related to a single event: *a 2010 world cup match between nigeria and argentinean*. However, the bursty segment "argentina vs nigeria" has a relatively much lower user frequency due to the principle of the least effort [169]. In contrast, comparing "argentina vs nigeria" with either "nigeria" or "argentina", the segment "argentina vs nigeria" would convey much more information about the event. Based on this observation, we assign each bursty segment s a weight $w_b(s, t)$ by using a logarithm function.

$$w_b(s, t) = P_b(s, t) \log(u_{s,t}) \quad (7.4)$$

The above weight scheme would keep the more bursty segments of the higher user frequency being ranked higher and the more bursty segments of the moderate user frequency being ranked relatively higher than the others.

By ranking the bursty segments by their weights $w_b(s, t)$, we then retain the top- K bursty segments as potential event-related segments (or simply event segments) for further processing. The value of K is non-trivial because a small K would result in a very low recall of events detected, and a large K may bring in more noise, leading to much higher computational cost as well as lower precision on the detected events. In practical, the optimal K value depends on the size of the time window, and requires some expertise

knowledge (*e.g.*, users from different regions may be interested in different topics [113]). In this work, we apply a heuristic strategy to filter out the bursty segments by setting K to $\sqrt{N_t}$.

Definition 7.2 [Event Segment] *A bursty segment s is a potential event-related segment (or simply event segment) in time window t if it is ranked among top- K bursty segments by $w_b(s, t)$ in descending order, where $K = \sqrt{N_t}$.*

7.2.3 Event Segment Clustering

Given a set of event segments detected from the previous step, we now cluster them into groups, each of which corresponds to a *possible* realistic event. Some event segments that cannot be clustered into groups are considered noise or non-event-related. These non-event-related segments are dropped from further processing.

7.2.3.1 Event Segment Similarity

Accordingly, we need to derive a similarity measure for each pair of event segments. Various similarity measures have been used in the past to cluster bursty features detected in formal texts, mainly based on co-occurrences of bursty features [42, 55]. However, similarity measure based on co-occurrence would not work well on tweets because they are much shorter in number of words compared to formal documents. Moreover, the topics in tweets are extremely dynamic and fast changing. Considering these two factors, we propose to measure similarity between two event segments by the content of their associated tweets and their temporal frequency patterns.

For each time window t , we further divide the time period evenly into M sub-time-window: $t = \langle t_1 \dots t_M \rangle$. The tweet frequency of an event segment s in sub-window t_m is denoted by $f_t(s, m)$. Let $T_t(s, m)$ be the set of tweets that each contains segment s and

is published within sub-window t_m . We define the similarity between a pair of segments s_a and s_b within time window t as follows:

$$sim_t(s_a, s_b) = \sum_{m=1}^M w_t(s_a, m) w_t(s_b, m) sim(T_t(s_a, m), T_t(s_b, m)) \quad (7.5)$$

where $sim(T_1, T_2)$ measures the similarity between two sets of tweets T_1 and T_2 , and $w_t(s, m)$ weighs the importance of sub-window t_m to segment s . To compute $sim(T_1, T_2)$, we concatenate all tweets in T_1 (resp. T_2) to form a pseudo document, and use cosine similarity with $tf \cdot idf$ scheme. The importance of sub-window t_m to segment s is the normalized frequency distribution over M sub-windows:

$$w_t(s, m) = \frac{f_t(s, m)}{\sum_{m'=1}^M f_t(s, m')} \quad (7.6)$$

Equation 7.5 illustrates that two event segments are similar if they have both similar tweet content and consistent frequency patterns along the time sub-windows. Either dissimilar tweet content or inconsistent frequency patterns leads to low similarity. More specifically, dissimilar tweet content suggests that two event segments refer to two distinct events. Inconsistent frequency pattern may suggest that the two event segments refer to two similar events but happened at different time points (*e.g.*, two football matches at the same day).

To be shown in our experiments (Section 7.3.3) content similarity is necessary to distinguish segments of different events having very similar tweet frequency distributions. Note that, because of the specific information conveyed by tweet segment, we believe that the content similarity of using all tweets containing the tweet segment is more meaningful than that using unigram. For example, the tweets containing segment **steve jobs** will be very different from the tweets containing either **steve** or the tweets containing **jobs** only.

7.2.3.2 Clustering by k -Nearest Neighbor Graph

Given the similarity measure in Equation 7.5, clustering event segments into possible events become straightforward and many existing clustering algorithms can be directly applied. We apply an variant of Jarvis-Patrick algorithm [68] for event segment clustering.

Given a graph of objects with edges indicating the similarity between any two objects, Jarvis-Patrick clustering algorithm partitions the graph by measuring the number of common neighbors among the k -nearest neighbors of the two objects. The partitioning involves two parameters: k and ℓ . Two objects are put into the same cluster if: 1) they are in each others' k -nearest neighbors, and 2) they share at least ℓ common nearest neighbors among the k -nearest neighbors. Note that, Jarvis-Patrick requires a single scan of all pairs of objects for clustering, which offers great scalability for Twitter stream-based event detection.

Considering the unique properties of short length and informal writing style of tweets, two event segments referring to the same event may not share a large number of common k -nearest neighbors to each other. Nevertheless, an event segment referring to a realistic event would likely appear in another event segment's k -nearest neighbors, and vice versa, given that the two event segments referring to the same event. We therefore relax the clustering criterion by considering only the first requirement: *two event segments appearing in each others' k -nearest neighbors are put into the same cluster*. With this relaxation, given a complete graph of event segments, the clustering will retain any edge between two event segments s_a and s_b if and only if they appear in each other's k -nearest neighbors. The resultant connected components are considered as ***candidate events***. If an event segment is in isolation and not grouped into any cluster, it is considered not event related and dropped from further processing. The clustering of event segments therefore requires only one parameter k and we set $k = 3$ in our experiments.

7.2.3.3 Candidate Event Filtering

Cambridge Dictionaries Online defines an *event* as “anything that happens, especially something important and unusual”⁴. However, we observe that many candidate events detected through clustering event segments are not realistic events. For instance, the segments "friday night", "friday", "weekends", "trip" and "enjoy" are returned as a possible event by the above procedures at some Friday (*e.g.*, Jun 18, 2010 covered in our dataset). More detailed human investigation shows that the tweets of this candidate event are from people who were talking about the plan or schedule for the coming weekend. Apparently, this kind of events can not be considered as realistic events. This calls for a mechanism to evaluate the “important and unusual” aspect of a candidate event obtained from event segment clustering.

We observe that many events involve well-known entities (*e.g.*, person names, locations, festivals) and many of these entities are documented in Wikipedia. Recall that each segment is produced in Section 6.2.3 with the preference towards Wikipedia entities (see Equation 6.10). We therefore again utilize Wikipedia to approximately evaluate “important and unusual” aspect of a candidate event. More specifically, we define *newsworthiness* measures for event segment and candidate event respectively⁵.

Definition 7.3 [*Segment Newsworthiness*] The *newsworthiness* $\mu(s)$ of a segment s is

$$\mu(s) = \max_{\ell \in s} e^{Q(\ell)} - 1$$

where ℓ is any sub-phrase of s , and $Q(\ell)$ is the prior probability that ℓ appears as anchor text in Wikipedia articles that contain ℓ .

⁴<http://dictionary.cambridge.org/dictionary/british/event?q=event>

⁵*newsworthiness* defined here may be not related to people’s judgment of newsworthiness, but rather a measure of segment novelty.

The exponential function is used in the equation since it is an increasing function with an increasing first derivative in the range of $[0, 1]$. That is, a segment with a larger $Q(\ell)$ would gain a relatively higher *newsworthiness* value. Next, we define the *newsworthiness* of a candidate event as follows.

Definition 7.4 [Event Newsworthiness] *The newsworthiness $\mu(e)$ of an event e containing a set of event segments $e_s = \{s\}$ is*

$$\mu(e) = \frac{\sum_{s \in e_s} \mu(s)}{|e_s|} \cdot \frac{\sum_{g \in E_e} \text{sim}(g)}{|e_s|}$$

where E_e is a set of edges that are retained during applying Jarvis-Patrick clustering, and $\text{sim}(g)$ is the similarity of edge g which is calculated by using Equation 7.5.

Observe that *newsworthiness* of a candidate event considers both the newsworthiness of its member event segments (*i.e.*, the first component) and the topology of the connected component formed by its member event segments (*i.e.*, the second component). The latter is equivalent to measure the density of the connected component in the clustering result. Therefore, a candidate event receives a high newsworthiness score if some phrases in its member segments are commonly used as anchor text in Wikipedia (indicating well known entities) and the member segments are well connected with strong cohesive topology.

We observe in our experiments, most top-ranked candidate events by newsworthiness are likely related to realistic events. On the other hand, noisy events likely have much lower *newsworthiness* scores. That is, the distribution of newsworthiness scores has a positive skewness. Let μ_x be the highest newsworthiness score among all candidate events detected in a given time-window. Based on the above observation, we consider a candidate event e to be a realistic event if the ratio between μ_x and $\mu(e)$ is smaller than a threshold τ , *i.e.*, $\frac{\mu_x}{\mu(e)} < \tau$, otherwise noise. Naturally, a lower threshold results in

better precision of the detected events but poorer recall, and vice versa. We investigate the impact of τ empirically in Section 7.3.

After filtering away noisy events, we represent each detected event with its member event segments sorted by *newsworthiness* scores. The top-ranked segments are used to describe the event. In this work, the top-5 segments are used to describe the event.

7.2.4 Discussion

The efficiency of *Twevent* is a non-trivial factor from a practical perspective. Recall that *Twevent* contains three main components: tweet segmentation, event segment detection, and event segment clustering, shown in Figure 7.1. We next discuss the computational cost for each component.

The running time of tweet segmentation is linear to the length of a tweet (in number of the words). As segmentation of one tweet is independent of segmentation of other tweets, parallel computing techniques can be easily utilized in this component. More importantly, tweet segmentation can be considered as a part of preprocess because all segments are stored in an index for further processing.

Event segment detection only requires one scan of segments' tweet frequency and user frequency. The time complexity is linear to the number of segments in each time window.

Most computation time is consumed by calculating the similarity between two event segments, and event segment clustering. However, the pair-wised event segment similarity is computed for a relatively small set (*e.g.*, $K = \sqrt{N_i}$) of event segments where N_i is the number of tweets published within a time window. That is, the time complexity is $O(K^2)$. The Jarvis-Patrick clustering algorithm used for event segment clustering requires one scan of all pairs of event segments within a time window. Given the relative small number of detected events in each time window, the running time for the candidate event filtering is negligible.

We conducted our experiments on a workstation with a 2.40GHz Xeon quad-core CPU and 24GB of RAM. Without considering the time taken for tweet segmentation⁶, *Twevent* takes about 18 seconds to detect events from average 143K tweets published in one day (*i.e.*, the time window).

7.3 Experiments

In this section, we report our extensive experiments on evaluating *Twevent*. We show that *Twevent* outperforms the state-of-the-art approach with both better precision and recall. We show that newsworthy segments make the detected events much easier to be interpreted by users. Further, we evaluate the usefulness of the notion of tweet segment against unigram, the effect of the parameters in *Twevent*.

7.3.1 Dataset and Experimental Setting

Wikipedia Data. The Wikipedia data used in tweet segmentation (Section 7.2.1) and *newsworthiness* measure (Section 7.2.3.3) are based on the Wikipedia dump released on 30 Jan, 2010. It contains 3,246,821 articles and 266,625,017 hyperlinks. In total, there are 4,342,732 distinct entities appeared as anchor texts in the Wikipedia dump.

Twitter Stream. A collection of tweets published by Singapore-based users (based on the location specified in user profile) in June 2010 is used to simulate a Twitter stream. This dataset was built by Weng and Lee for evaluating the *EDCoW* event detection method in [151]. There are a total of 4,331,937 tweets published by 19,256 unique users in the dataset. A number of realistic events happened in the data collection period, such as FIFA World Cup 2010, WWDC 2010, and MTV Movie Awards 2010.

Figure 7.2(a) shows the average number of tweets published within each hour of a day. Most tweets are published within 6AM to 6PM, with relatively more tweets published in the afternoon.

⁶In our experiments, the speed of tweet segmentation relies on Microsoft Web N-Gram Web service.

Parameter Setting. There are several parameters that could affect the performance of *Twevent*. The size of the time-window t and the number of sub-time-windows M are the two basic parameters for *Twevent*. In our evaluation, we fix t to be a day and set $M = 12$. That is, each sub-time-window is 2 hours.

Recall that we retain only top $K = \sqrt{N_t}$ bursty segments as event segments, where N_t is the number of tweets published in the time window t ; For Jarvis-Patrick clustering, we set $k = 3$ for the number of nearest neighbors in the graph; To distinguish realistic events from noise, we set threshold on the ratio of newsworthiness τ . In our experiments, we observe that parameter τ affects the event detection accuracy of *Twevent* more significantly than the other two parameters K and k . We therefore intensively investigate the impact of varying τ and fix $K = \sqrt{N_t}$ and $k = 3$ throughout our evaluation. We also investigate the impact of varying K and k in this section.

Evaluation Metric. The dataset does not come with ground truth labels on all realistic events within the data collection period. Because it is infeasible to manually label the over 4 million tweets in the dataset, we choose to manually evaluate the detected events returned by *Twevent*. We use *precision* and *recall* to evaluate the accuracy of the events detected.

We follow the definition of **Precision** used in [151], which is defined as the fraction of the detected events that are related to a realistic event. However, recall was not defined in [151] because of the lack of the ground truth labels in the dataset. In our work, we choose to report **Recall** as *the number of distinct realistic events detected* from the dataset on daily basis⁷. Note that, if two detected events are both related to the same realistic event within the same time window (*i.e.*, one day), then both are considered correct in terms of precision, but only *one* realistic event is considered in counting recall.

⁷Note the recall defined here is different from the definition in information retrieval area. We use it here to compare the performance of different algorithms under investigation.

Figure 7.2: Statistics on tweets and tweet segments

Due to this reason, we also define *Duplicate Event Rate* (or simply **DERate**) to denote the percentage of events that have been dublicately detected among all realistic events detected.

7.3.2 Statistics on Tweet Segmentation

We first report statistics on the tweet segments returned by the tweet segmentation component. After removal of stop-words and words with non-English characters, there are 662,088 distinct words or unigrams retained in the data. Tweet segmentation is then applied to each tweet. A total of 1,275,809 distinct segments are obtained after segmentation.

Observe from Figure 7.2(b), the tweet frequency of a segment follows a power-law distribution. On average, each segment is contained in 19 tweets. In Figure 7.2(c), we report the distribution of unigram and multi-gram segments. The figure shows that about 50% of the segments are 2-grams; segments with more than 3 grams are very rare. We observe that 2-gram segments cover a large proportional of named entities, such as "lady gaga" and "justin bieber", or location name like "orchard road". Moreover, many 3-grams segments are very informative, like "mtv movie awards" and "penalty shoot out". More examples are given in Table 7.3.

Table 7.1: Detection results of *Twevent*, *Twevent_u*, and *EDCoW*.

Method	No. events detected	Precision	Recall	DERate
<i>EDCoW</i>	21	76.2%	13	23.1%
<i>Twevent</i>	101	86.1%	75	16.0%
<i>Twevent_u</i>	146	75.3%	78	41.0%

7.3.3 Event Detection Results

We compare *Twevent* with two methods: *EDCoW* and *Twevent_u*. The latter is a variant of *Twevent* without using tweet segment but using unigram in the event detection process, with the same parameter settings as *Twevent* except the setting for threshold τ . In this set of experiments, we set $\tau = 4$ for *Twevent* and $\tau = 3$ for *Twevent_u*. The impact of varying τ will be reported in Section 7.3.4.

Recall that we measure the segment’s newsworthiness using Definition 7.3. Based on this definition, a unigram is likely to have zero newsworthiness score. Thus, we change the newsworthiness definition for unigram to be $\mu(w) = e^{Q(w)}$ for a unigram w . The modified definition strongly favors informative unigrams, leading to better representations for the detected events. Next, we report the event detection accuracy of the three methods.

Event detection accuracy. Table 7.1 reports the number of events detected, the precision and recall, of the three methods respectively. The results of *EDCoW* are reproduced from [151]⁸. Shown in the table, our proposed method *Twevent* yields the best precision of 86.1% which is significantly larger than the precisions achieved by *EDCoW* and *Twevent_u*. Observe that our method *Twevent* detects 101 events with a recall of 75 realistic events. On the same dataset, *EDCoW* detects 21 events in total with 13 realistic events. *Twevent_u* yields a slighter worse precision than *EDCoW* (75.3% vs 76.2%) but detects the largest number of realistic events. In terms of DERate, *Twevent* achieves the lowest rate despite that our method detects much more events than *EDCoW* (101

⁸All events detected by *EDCoW* are reported in Table 1, Page 407 in [151].

Table 7.2: Events detected by *EDCoW* in June 07 – June 12, 2010 (reproduced from [151], following their event id.)

Day	e_{ID}	Event keywords
7	e_7 .	kobe, kristen
	e_8 .	#iphone4, ios4, iphone
8	e_9 .	reformat, hamilton
	e_{10} .	avocado, commence, ongoing
9	e_{11} .	#failwhale, twitter
10	e_{12} .	vuvuzela, soccer
11	e_{13} .	#svk, #svn
12	e_{14} .	#kor, greec, #gre

vs 21). On the other hand, we observe that $Twevent_u$ delivers the worst DERate, more than double of $Twevent$ (41% vs 16.0%). That is, the unigrams about the same event are clustered into two or more events. Because a tweet segment usually conveys very specific information, the tweets containing the tweet segment are all about the same topic (*e.g.*, the event). Two tweet segments about the same event are therefore have higher chance to be clustered together.

From the list of events detected, we observe that an event is re-detected mainly because users discuss the event from different perspectives, or one event is a sub-event of another. We use the two events e_{22} and e_{20} detected on 12 Jun 2012 as an example for illustration. Listed in Table 7.3, e_{22} , detected with segments [usa, england, eng, vs], refers to the football match between England and USA in 2010 World Cup; e_{20} with [steven gerrard, captain, score, scored, gerrard] refers to the caption of England, Steve Gerrard, scored a goal in this match. Because both events refer to the same match, we count them as one realistic event in our recall.

Event interpretation. We argue that the notion of *tweet segment* not only benefits better precision in event detection, but also makes the detected events much easier to be interpreted. In above discussion, we show that person name [steven gerrard] is detected as an event segment in the result. We now give more examples by listing all the events

Figure 7.3: Comparison of frequency distributions for the two events: Twilight and Park Yong-ha

Figure 7.4: *Twevent* and *Twevent_u* against different τ values

detected by *Twevent* between June 07 to June 12, 2010⁹. This 6-day period is chosen because it covers a wide range of events that happened in June, 2010, including Apple WWDC 2010, MTV Movie Awards 2010, and FIFA World Cup 2010, among others.

From Table 7.3, we make two observations. First, many event segments are multi-gram segments such as various types of named entities [steve jobs], [mtv movie awards], [katy perry], and [karate kid], and segments conveying concrete information like [argentina vs nigeria] and [season finale]. These segments make the events much easier to be interpreted

⁹Due to page list, we do not list all the 101 events detected by *Twevent* for the whole month of June 2010. The full list is available at <http://www.cais.ntu.edu.sg/~lichenliang/twevent/eventlist.txt>.

than some unigram keywords. For comparison, we also list the keywords of all the 8 events detected by *EDCoW*, reproduced from [151], and the keywords of all the 40 events detected by *Twevent_u* during the 6-day period in Tables 7.2 and 7.4, respectively. We can see that the keywords detected by *EDCoW* is relatively hard to interpret than the two methods *Twevent* and *Twevent_u*. Compare the latter two methods, we argue that *Twevent* detects more semantically meaningful keywords/phrases than *Twevent_u*. For instance, we use the first two events detected by both methods as example. Although both methods detect similar keywords for the WWDC 2010 event, [steve jobs, imovie, wwdc, iphone, wifi] is easier to interpret than [wwdc, keynote, live, jobs, steve]. Similarly, *Twevent* detects [mtv movie awards, mtv, new moon, twilight, awards] while *Twevent_u* outputs [mtv, moon, twilight, movie, awards]; the phrase [mtv movie awards] makes the event much easier to interpret. Second, the detected events by *Twevent* cover a wide range of events, such as Korea music bands, Apple WWDC 2010, MTV Movie Awards 2010, release of music videos and movies, and football matches of World Cup 2010. That is, *Twevent* does not favor certain types of events than others. Among the 22 events listed in Table 7.3, only 1 event has no corresponding real-life events leading to a precision of 95.5% in this 6-day period.

Case study. We use a case study to illustrate the importance of using content in event detection from tweets. Figure 7.3 plots the relative frequency of tweets published on June 30, 2010 related to two events. The first event with segments [harry potter, twilight, hours, trailer] is about the movie *The Twilight Saga: Eclipse*, which was released on June 30, 2010. The trailer of the Movie *Harry Potter and the Deathly Hallows Part 1* was shown before *The Twilight Saga: Eclipse*. The second event, with segments [park yong ha, rip, peace, hope, park] refers to the suicide case of Korean actor Park Yong-ha on June 30, 2010. Observe from Figure 7.3, the two events have very similar tweet frequency distribution over the 24 hours of the day. The tweet frequency of an event is defined as the relative frequency of the tweets published in an hour that contain any event segment

of the event. We argue that it is hard to distinguish the two events without using content similarity between the event segments.

7.3.4 Impact of τ in Twevent

While event segments are clustered into candidate events, the ratio threshold τ defines the boundary between the realistic events and the noisy events. We next analyze the effect of τ value on the performance of *Twevent* and *Twevent_u*.

Figure 7.4(a) plots the precision and DERate of the two methods when changing τ from 2 to 10. We make the two main observations. First, for both methods, precision degrades along the increase of τ as more events are considered as realistic events. Nevertheless, the rate of degradation for *Twevent* is much smaller than that of *Twevent_u*. Observe that *Twevent* maintains very good precision above 80% even when $\tau = 10$, which is better than the best precision achieved by *Twevent_u* with all τ values. Second, increasing of τ leading to increase in DERate. Shown in Figure 7.4(a), the DERate of *Twevent* is about half of *Twevent_u* for most τ values and stop increasing when $\tau > 6$. In summary, the use of tweet segmentation in event detection contributes to much higher precision and much lower duplicate event detection rate.

Figure 7.4(b) reports the number of events detected and the recall values along the change of τ values for the two methods. For both methods, increase τ leads to better recall. Observe that, although *Twevent_u* always achieves better recall than *Twevent*, the differences between the recall values do not change much along the increase of τ . However, increment of τ leads to sharp increase of the number of events detected for *Twevent_u*. But due to the poorer precision along the increase of τ , the number of realistic events detected does not increase at the same pace as the number of detected events.

7.3.5 Impact of K and k in *Twevent*

We cluster event segments by considering k -nearest neighbors. We conduct the experiments by setting $k = 2, 4, 5$ and $\tau = 4$, and we compare the results with the setting of $k = 3$. We observe that the number of event segments on average for a detected event is smaller when $k = 2$. For example, event e_{10} in Table 7.3 becomes “*watching glee, glee, channel*”, which misses a meaningful segment “*season finale*”. Also, some events produced by $k = 3$ are separated into two events. For example, event e_7 are separated into two events “*lady gaga, music video, gaga, alejandro*” and “*youtube, youtube video, liked*”. The latter one becomes meaningless without the context like “*lady gaga, music video*”. When k is larger, some events are augmented with additional segments that are meaningful. For example, event e_6 in Table 7.3 is augmented with meaningful segments like “*contract*”, “*facetime*” and “*price*”. However, we also observe that most events are augmented with noisy segments. Some events have more than 20 event segments. Given a noisy segment could have a low similarity relation to other segments, a larger k could help cluster these noisy segments as a member segment of a realistic events. In our empirical experiments, we observe that the best performance is obtained when $k = 3$. This is reasonable since the a tweet is limited to 140 characters, which makes the number of the event segments of a realistic event is relatively small.

With a larger K , more bursty segments are considered as event segments. We conduct the experiment by setting $K = 2\sqrt{N_t}$. We observe that more events are detected. However, most of these events are relative less popular than that of $K = \sqrt{N_t}$, or duplicated. We believe that the choice of K is dependent on the nature of the twitter stream. In this work, we empirically study the performance of *Twevent* by setting $K = \sqrt{N_t}$. The experiment conducted here shows that *Twevent* is a robust system against the different settings.

7.4 Summary

Twitter, as a new type of social media, has experienced an explosive growth in terms of both users and information volume in recent years. The characteristics of tweets propose severe challenges to many tasks including event detection. In this chapter, we present a novel event detection system for Twitter stream, called *Twevent*, to tackle the adverse impacts of tweets: short and noisy content, diverse and dynamic topics, and large data volume. One of the key concept in *Twevent* is to use tweet segment instead of unigram for identifying the bursty features and then distinguishing the realistic events from the noisy ones. *Twevent* demonstrates outstanding performance in our experiments: effectiveness, informativeness, and efficiency.

Table 7.3: List of the 22 events detected by *Twevent* during the period of June 07 to June 12, 2010.

Day	e_{ID}	[Event Segments]: Event Description
7	e_1 .	[steve jobs, imovie, wwdc, iphone, wifi] : iPhone4 was released during WWDC 2010.
	e_2 .	[mtv movie awards, mtv, new moon, twilight, robe] : The movie <i>The Twilight Saga: New Moon</i> was the biggest winner in MTV Movie Awards 2010; it took 4 out of 10 "Best" Awards.
	e_3 .	[yesung, yesung oppa, kyuhyun, oppa, kyu] : Korean popular band Super Junior's showcase was held on June 6, 2010 at Singapore. Yesung Oppa and Kyuhyun Oppa are members of Super Junior.
8	e_4 .	[lady gaga, music video, gaga, mv, alejandro] : The music video <i>Alejandro</i> by Lady GaGa was premiered officially on June 8, 2010.
	e_5 .	[ss501, indonesia, ariel, sama, trend] : No clear corresponding real-life event.
	e_6 .	[singapore, iphone 4g, iphone 3gs, iphone, coming out] : Related to event e_1 . People started to talk about the release date of iPhone 4 in Singapore.
9	e_7 .	[lady gaga, youtube, youtube video, music video, gaga] : Related to event e_4 .
	e_8 .	[twitter, whale, stupid, capacity, over again] : A number of users complained they could not use twitter due to over-capacity. A logo with whale is usually used to denote over-capacity.
	e_9 .	[ipad, iphone, apple, new] : Related to event e_1 .
	e_{10} .	[watching glee, glee, season finale, season, channel] : The season finale of the American TV series <i>Glee</i> was broadcasted on June 8, 2010.
10	e_{11} .	[lady gaga, youtube, youtube video, music video, amber] : Related to event e_7 .
	e_{12} .	[justin bieber, try, pa, took, each] : Related to event e_{15} . The song <i>Never Say Never</i> by Justin Bieber serves as the theme song for the movie <i>The Karate Kid</i> , which was released on June 10, 2010 in Singapore.
	e_{13} .	[yesung, tweeted] : Super Junior's Yesung posted a photo about his pet turtles.
	e_{14} .	[twitter, whale, stupid, capacity, over] : Related to event e_8 .
	e_{15} .	[karate kid, watch movie, movie] : The movie <i>The Karate Kid</i> was released on June 10, 2010 in Singapore.
11	e_{16} .	[uruguay vs france, uruguay, france, vs] : A match between Uruguay and France in World Cup 2010.
	e_{17} .	[south africa, vs mexico, mexico, goal, first goal] : A match between South Africa and Mexico in World Cup 2010. And the first goal of the 2010 World Cup was scored in the match.
12	e_{18} .	[arg, argentina, argentina vs nigeria, nigeria, messi] : A match between Argentina and Nigeria in World Cup 2010.
	e_{19} .	[south korea, greece, korea vs greece, korea won, korea] : A match between South Korea and Greece in World Cup 2010.
	e_{20} .	[steven gerrard, captain, gerrard, scores] : Related to event e_{22} . The captain of England, Steve Gerrard scored a goal in the match.
	e_{21} .	[ji sung, park, scored, jisung] : Related to event e_{19} . Park Ji-Sung, the caption of South Korea, scored a goal against Greece.
	e_{22} .	[usa, england, eng, vs] : A match between England and USA in World Cup 2010.

Table 7.4: Events detected by $Twevent_u$ in June 07 – June 12, 2010

Day	e_{ID}	Event keywords
7	e_1 .	wwdc, keynote, live, jobs, steve
	e_2 .	mtv, moon, twilight, movie, awards
	e_3 .	yesung, ryeowook, oppa, hope, welcome
	e_4 .	indonesia, blues, rain, weather, star
	e_5 .	iphone, apple, video, available, os
	e_6 .	singapore, asia, showcase, junior, tickets
	e_7 .	check, website, gain, member, site
	e_8 .	justin, both, pa, took, each
	e_9 .	perry, katy
	e_{10} .	singtel, samsung, galaxy, cute, anniversary
	e_{11} .	tumblr, white, black, hair, model
8	e_{12} .	gaga, lady, mv, alejandro
	e_{13} .	singapore, iphone, apple, july, features
	e_{14} .	wwdc, keynote, jobs, steve
	e_{15} .	right, think, one, need, know
9	e_{16} .	twitter, whale, capacity, fail, over
	e_{17} .	fan, wish, pretty, follow
	e_{18} .	youtube, glee, season, video, finale
	e_{19} .	gaga, lady, mv, alejandro
	e_{20} .	rice, chicken, eat, food, lunch
	e_{21} .	mtv, asia, wednesday, awards
	e_{22} .	try, pa, took
	e_{23} .	home, out, going, i'm
10	e_{24} .	day, work, one, need, know
	e_{25} .	youtube, gaga, video, lady, liked
	e_{26} .	twitter, whale, stupid, capacity, fail
	e_{27} .	radio, station, heard, addicted
	e_{28} .	justin, try, pa, took
	e_{29} .	karate, movie, kid, watched
	e_{30} .	congrats, blue, jay, name, jaywalkers
	e_{31} .	boss, centre, thursday, suntec, fair
	e_{32} .	facebook, internet, camera, photo, photos
	e_{33} .	cup, world, opening
11	e_{34} .	mexico, africa, goal, scored, south
	e_{35} .	uruguay, france
12	e_{36} .	park, ji, sung, jisung
	e_{37} .	argentina, nigeria, usa, england, messi
	e_{38} .	captain, steve, gerrard, scores
	e_{39} .	greece, korea, koreans, han, min
	e_{40} .	goalkeeper, robe, green, ball, mistake

Chapter 8

Conclusion and Future Work

I have never let my schooling interfere with my education.

– Mark Twain

8.1 Summary

The online social media has played an increasing important role for web users in the past few years. It provides an interactive platform for individuals and communities to create, share, and spread user-generated content. The explosive growth of user-generated content in online social media further aggravates the problem of information overload suffered by web users. Two critical issues emerge along with the urgent demands of new IR and NLP techniques for online social media: (i) efficient solutions for IR and NLP tasks to deal with the astronomical volume of digital information; (ii) effective solutions for IR and NLP tasks to deal with the noisy and dynamically changing user-generated content.

In this thesis, we focus on developing novel solutions in several concrete IR and NLP tasks. The intensive interactions enabled in online social media make it a multi-dimensional social network. We first investigate the explicit and implicit relations obtained from different perspectives in the context of Wikipedia. Different similarity measures for Wikipedia articles focusing on different perspectives of the collaborative knowledge building system are examined. We show that each similarity measure discovers

some specific facet about the semantic relation among Wikipedia articles. These similarity measures are complementary to each other, leading to a better understanding of dynamics of the knowledge building. Specifically, we study the source of controversy from different aspects of Wikipedia, by utilizing each similarity measure of a particular perspective. In the context of online social media, it has an important implication and provides guideline for many IR and NLP related tasks. That is, the possibility of using different aspects of relations to determining new information, which may not be derivable or misleading by considering only a specific relation aspect.

Traditional IR technologies are developed based on the BOW model, where the word order is ignored. This kind of document representation method loses a lot of rich semantic information such as polysemy, synonym, abbreviations and associative relations, resulting in an moderate performance. More importantly, the high number of dimensions incurred by BOW makes the systems often inefficient. It is desirable to abstract the documents with semantic information units of higher level. Wikipedia is the largest online encyclopedic that each Wikipedia article describes a single concept. Due to its exhaustive coverage, the recognized high quality and semantic structure (*e.g.*, hyperlink, redirect pages and disambiguation pages), Wikipedia has become a natural choice as a external knowledge base to transform documents from BOW to concept representations. Given the semantic relations collaboratively encoded in Wikipedia, the concept representation based on Wikipedia can support many IR tasks. In Chapter 4, we propose a generic word sense disambiguation framework based on Wikipedia, called *TSDW*. *TSDW* parses the content of each document and identifies the keyphrases that would be referred to some Wikipedia article. Each ambiguous keyphrase is then associated with one candidate sense by *TSDW*, by considering both unambiguous keyphrases as well as other ambiguous keyphrases. The valuable semantic information contained in the ambiguous keyphrases help alleviate the data sparsity problem encountered by using

the unambiguous keyphrases alone. The selection of representative unambiguous and ambiguous keyphrases enables *TSDW* to achieve better disambiguation accuracy with moderate computation cost. Also, *TSDW* is language independent and adaptive to different context settings. These unique characteristics make it a desirable solution for the semantic applications on large data volume. As to evaluation, we conduct extensive experiments over different datasets of two language: English and Traditional Chinese. To the best of our knowledge, our work is the first to study the performance of existing state-of-the-art approaches and *TSDW* using datasets in multiple (two) languages, in terms of both effectiveness and efficiency. This has significant implications for any future related works. In Chapter 5, we then study the task of tag recommendation in the context of social tagging by applying concept model, of which the concept space is derived based on Wikipedia. This investigation has two-fold meanings. Firstly, we can examine the performance of concept model in information retrieval task, like tag recommendation. Secondly, we can investigate the possibility of harnessing the wisdom of crowds emerged in the social tagging environments. Existing works showed that social tags are quite helpful in many IR tasks, such as web object clustering and classification, web search, etc. Thus, an efficient and effective tag recommendation system would benefit not just the users in the social tagging systems, but support other IR tasks. We apply *TSDW* to derive the concepts from each document. We attempt to emulate human tagging behavior by adopting a probabilistic framework for tag recommendation. Through the experiments, we find out that concept model can capture the tagging behaviors more accurately and bring significant reduction in computation cost. Moreover, the study of different document weighting schemes offers us a consistent conclusion that we derive in Chapter 3: combining different information from different aspects can help extract accurate knowledge and only a single relation in a multi-dimensional network may result in a biased impact.

The user-generated content in online social media is error-prone. For example, these adverse features are very prevalent in user comments and microblogs. However, understanding the users' opinion in these domains is very useful for many different fields, such as product marketing, politic elections and crisis response. In Chapter 6, we propose an novel 2-step unsupervised method for named entity recognition in targeted Twitter stream, called *TwNER*. It does not depend on the unreliable local linguistics features. In the first step, it leverages on the *global context* obtained from Wikipedia and Web N-Gram corpus to partition tweets into valid segments (phrases) using a dynamic programming algorithm. Each such tweet segment is a candidate named entity. It is observed that the named entities in the targeted stream usually exhibit a *gregarious* property, due to the way the targeted stream is constructed. In the second step, *TwNER* constructs a random walk model to exploit the *gregarious* property in the *local context* derived from the Twitter stream. The highly-ranked segments have a higher chance of being true named entities. The evaluation based on two sets of real-life targeted streams shows that *TwNER* achieves comparable performance as with conventional approaches in both streams.

Traditional approaches for event detection mainly focus on the bursty features by investigating the temporal pattern and are BOW model based. However, the content in online social media like microblogging systems are often noisy and diverse. And these user-generated content are always changing, leading to a mixture of pointless babble, conversation, and event-related reports, etc. In Chapter 7, we introduce a segment-based event detection system for tweets, called *twevent*. *Twevent* first detects bursty tweet segments as event segments and then clusters the event segments into events considering both their frequency distribution and content similarity. After clustering bursty segments into candidate events, Wikipedia is exploited to identify the realistic events and to derive the most newsworthy segments to describe the identified events. We evaluate

Twevent and compare it with the state-of-the-art method using 4.3 million tweets published by Singapore-based users in June 2010. In our experiments, *Twevent* outperforms the state-of-the-art method by a large margin in terms of both precision and recall. More importantly, the events detected by *Twevent* can be easily interpreted with little background knowledge because of the newsworthy segments. We also show that *Twevent* is efficient and scalable.

8.2 Future Work

As the work in this thesis has a broad scope over IR and NLP tasks on online social media, we plan to continue working on either extending our previously developed approaches and resolving possible drawbacks, or moving to other open issues. We briefly discuss them below:

- The generic word sense disambiguation framework, *TSDW*, proposed in Chapter 4 is designed by considering a specific similarity measure. From the error analysis conducted in Section 4.3.4.4, we know that the discriminative power offered by a specific similarity measure of a single perspective is limited in some cases. Take the case study about keyphrase *joey* in Wikipedia article *Don Porter* discussed in Section 4.3.4.4 as an example. The similarity measure based on the wikilink structure would give the close similarity scores for both candidate topics *Joey (1985 film)* and *Joey (TV series)*, which leads to a wrong disambiguation (*i.e.*, *Joey (TV series)*). However, using *expert-based similarity* could give us a preference to the true topic *Joey (1985 film)*. Since the actor *Don Porter* played in the film *Joey (1985 film)*, the similarity based on the authorship could give *Joey(1985 film)* a much higher score than *Joey (TV series)*. That is, the authors of *Joey (1985 film)* are expected to be familiar with the actor *Don Porter*, and hence contribute to the

story of the actor, and vice versa. We plan to develop a mechanism to incorporate multiple kinds of relations of different perspectives in the framework of *TSDW*.

- In Chapter 5, we emulate human tagging behavior by incorporating the concept model with the probability framework (*i.e.*, Naive Bayes). Naive bayes model designed for BOW model ignores the semantic relations among single words. Such probability model often leads to poor performance when deals with short text. Recently, the translation-based language model was proposed to deal with the data sparsity problem in the domain of question and answer (CQA) retrieval [70, 157]. The proximity for each pair of words is calculated based on the association patterns in the corpus, and integrated with the existing language model. Similarly, the concepts identified from each document could be also sparse and limited, which make the traditional language model performs poorly. On the other hand, the concept model derived from Wikipedia naturally provides with more information. It is reasonable to derive the semantic relations for each pair of Wikipedia concepts based on the semantic structure of Wikipedia (*e.g.*, wikilinks and categorization). It would be interesting to exploit the semantic relations between Wikipedia concepts in the task of tag recommendation.
- We are also interested in extending *TwNER* proposed in Chapter 6 for entity type classification. The possible methodology would be exploiting the local context of each named entity detected as well as the types of its frequent co-occurrences. Moreover, the strategy for identify the suitable K value is also scheduled for future work. The possible features related to the choice of K should include the topic cohesiveness, *gregarious* property and the size of tweet collection.
- Chapter 7 proposes an event detection system for tweets (*Twevent*). *Twevent* focuses on the temporal property of tweet segments. Other features, like retweet rate

and hashtags, could be complementary indicators besides the tweet segments. Also the mechanism to quantify the newsworthy of a detected event that are not covered well by Wikipedia is an interesting topic. For example, retweet rate and the quality of the related hashtags could be useful indicators in such situation.

- The current work utilizing the wisdom of crowds of online social media often focuses on a single system. Different kinds of online social media could provide us with different knowledge that are complimentary to each other. For example, transferring learning has become a hot topic in research community, which tries to leverage the knowledge of other domains to address the problems of the targeted domain. We wish to aggregate the wisdom from different online social media for more challenging issues and tasks, *e.g.*, short text classification and clustering, user opinion mining, etc.

Appendix A

Appendix

A.1 Keyphrase Recognition

Let $k = w_1w_2...w_m$ be a keyphrase of length m , and $K = \{k\}$ be all keyphrases in the Wikipedia inventory. We group the keyphrases based on their prefix words, so that each group K_w is a set of keyphrases with the same prefix word: $K_w = \{k | k = w_1w_2...w_m, w_1 = w, m \geq 1\}$. Then, any group K_w can be accessed in constant time by looking up a hashtable with the key being the prefix word w . For each group K_w , we build a prefix tree for all keyphrases within this group. Specifically, given a keyphrase in K_w , we start creating a path from the root, where the node at level i denotes the word at position $i + 1$ of the keyphrase. The root of the tree is the prefix word w at level 0. Each node has a boolean mark indicating whether it is the last word of some keyphrases, i.e., the path from the root to the node constitutes a keyphrase in K_w . In this way, we build an index for the Wikipedia inventory as a forest, where each prefix tree of the forest corresponds to a keyphrase group K_w with the same prefix word w . We call such a prefix tree a *keyphrase tree*.

Figure A.1 illustrates an example keyphrase tree based on the prefix word “java”. In this figure, the tree contains 9 keyphrases: java, java virtual machine, java virtual machine heap, java virtual machine tools interface, java sdk, java speech api, java speech api markup language, java speech markup language, java swing. Algorithm 2 outlines the keyphrase

Figure A.1: Example of Keyphrase Tree of initial word *java*

recognition algorithm using the Wikipedia inventory. Given an input text, we process the recognition word after word. Let the word at position i of the input text is w_i , the corresponding keyphrase tree is looked up via a hashtable for key w_i (lines 4). If no such tree exists, we skip the current word and proceed to the next word (lines 7, 26). When such a keyphrase tree exists, we dive into the tree and identify the matched child node with the next word w_{i+1} (lines 13-14). The child node search operation can be realized efficiently by using a hashtable or a binary tree data structure. If a child node matches w_{i+1} , the search process continues to match the next word w_{i+2} . During the search process, if a node is marked as the last word, we store the corresponding keyphrase in a variable s (lines 17-18). Then, variable s contains the longest keyphrase identified so far (along some path of the keyphrase tree). The recognition process continues until we reach the leaf node of the path, or fail to match the next word with any child node (lines 13-24). After the search process terminates, the keyphrase contained in s is returned as the recognized longest keyphrase (lines 22). If a keyphrase with length m is recognized, the recognition process continues with the word at position $i + m$ (line 22). The algorithm has a complexity of $O(n)$, where n is the length of the input text in number of words.

```
input :
    A text:  $t = w_1w_2...w_n$ ;
    A hash table:  $f(w, tree)$  indexes the forest of Wikipedia inventory;
output:
    A set of matched keyphrases  $\mathbb{S}$ ;
 $\mathbb{S} = \{\}$ ;  $m = 1$ ; //  $m$  is either 1 or the length of last recognized keyphrase
for  $i = 1; i \leq n; i++ = m$  do
     $w = w_i$ ;
     $t_w = f.get(w)$ ; // look up the prefix tree associated with word  $w$ 
     $s = ""$ ;
     $pre = ""$ ;
    if  $t_w \neq null$  then
         $node = t_w.root()$ ;
         $pre.append(w)$ 
        if  $node.lastword()$  then
             $s = pre.string()$ ;
        for  $j = i + 1; j \leq n; j++$  do
             $w = w_j$ ;
             $node = node.child(w)$ ;
            if  $node \neq null$  then
                 $pre.append(node.word())$ ;
                if  $node.lastword()$  then
                     $s = pre.string()$ ; // store the longest keyphrase matched so far
            else
                break; // no further word can be matched
        if  $s \neq ""$  then
             $\mathbb{S}.add(s)$ ;  $m = s.length()$ ; // skip all words of the matched keyphrase
        else
             $m = 1$ ;
    else
         $m = 1$ ;
return  $\mathbb{S}$ ;
```

Algorithm 2: Keyphrase Recognition

A.2 List of Publications

- **Chenliang Li**, Aixin Sun, Anwitaman Datta. *TSDW: Two-Stage Word Sense Disambiguation to Wikipedia*. Journal of the American Society for Information Science and Technology (**JASIST**), Volume 64, Issue 6, Pages 1203 – 1223, June 2013.
- **Chenliang Li**, Aixin Sun, Anwitaman Datta. *Twevent: Segment-based Event Detection from Tweets*. The 21st ACM International Conference on Information and Knowledge Management (**CIKM'12**), Pages 155 – 164, Maui, USA, Oct 2012.
- **Chenliang Li**, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun and Bu-Sung Lee. *TwiNER: Named Entity Recognition in Targeted Twitter Stream*. The 35th Annual International ACM SIGIR Conference (**SIGIR'12**), Pages 721 – 730, Portland, Oregon, USA, 12-16 August 2012.
- **Chenliang Li**, Anwitaman Datta, Aixin Sun. *Mining Latent Relations in Peer-Production Environments: A Case Study with Wikipedia Article Similarity and Controversy*. Social Network Analysis and Mining (**SNAM**), Volume 2, Issue 3, Pages 265 – 278, Springer, 2012.
- **Chenliang Li**, Anwitaman Datta, Aixin Sun. *Semantic Tag Recommendation Using Concept Model*. The 34th Annual International ACM SIGIR Conference (**SIGIR'11**), Pages 1159 – 1160, Beijing, China, July, 2011.
- **Chenliang Li**, Aixin Sun, Anwitaman Datta. *A Generalized Framework for Word Sense Disambiguation based on Wikipedia*. The 33rd European Conference on Information Retrieval (**ECIR'11**), Pages 653 – 664, Dublin, Ireland, April, 2011.

Bibliography

- [1] Bibsonomy. <http://www.bibsonomy.org/>.
- [2] Citeulike. <http://www.citeulike.org/>.
- [3] Delicio.us. <http://www.delicious.com/>.
- [4] Digg. <http://digg.com/>.
- [5] Facebook. <http://www.facebook.com/>.
- [6] Netflix. <http://www.netflix.com/>.
- [7] Open directory project. <http://www.dmoz.org/>.
- [8] Twitter. <http://www.twitter.com/>.
- [9] Weibo. <http://www.weibo.com/>.
- [10] Wikipedia. <http://www.wikipedia.org/>.
- [11] Yahoo! answer. <http://answers.yahoo.com/>.
- [12] Youtube. <http://www.youtube.com/>.
- [13] S. F. Adafre and M. de Rijke. Discovering missing links in wikipedia. In *Proc. of LinkKDD*, pages 90–97. ACM, 2005.

- [14] B. T. Adler and L. de Alfaro. A content-driven reputation system for the wikipedia. In *Proc. of WWW*, 2007.
- [15] E. Agirre and O. L. de Lacalle. Supervised domain adaption for wsd. In *Proc. of EACL*, 2009.
- [16] E. Agirre and A. Soroa. Personalizing pagerank for word sense disambiguation. In *Proc. of EACL*, 2009.
- [17] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proc. of SIGIR*, 1998.
- [18] G. W. Allport. *The nature of prejudice*. Basic books, 1979.
- [19] K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu. Can all tags be used for search? In *Proc. of CIKM*, 2008.
- [20] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.
- [21] J. E. Blumenstock. Size matters: Word count as a measure of quality on wikipedia. In *Proc. of WWW*, 2008.
- [22] U. Brandes, P. Kenis, J. Lerner, and D. van Raaij. Network analysis of collaboration structure in wikipedia. In *Proc. of WWW*, 2009.
- [23] U. Brandes and J. Lerner. Visual analysis of controversy in user-generated encyclopedias. *Information Visualization*, 7(1):34–48, 2007.
- [24] T. Brants, F. Chen, and A. Farahat. A system for new event detection. In *Proc. of SIGIR*, 2003.

- [25] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proc. of EACL*.
- [26] L. S. Buriol, C. Castillo, D. Donato, S. Leonardi, and S. Millozzi. Temporal analysis of the wikigraph. In *Proc. of WI*, 2006.
- [27] D. Carmel, H. Roitman, and N. Zwerdling. Enhancing cluster labeling using wikipedia. In *Proc. of SIGIR*, 2009.
- [28] G. Chrupala and D. Klakow. A named entity labeler for german: Exploiting wikipedia and distributional clusters. In *Proc. of LREC*.
- [29] R. L. Cilibrasi and P. M. B. Vitanyi. The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):370–383, Mar. 2007.
- [30] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proc. of EMNLP-CoNLL*, 2007.
- [31] J. F. da Silva and G. P. Lopes. A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora. In *the 6th Meeting on Mathematics of Language*, 1999.
- [32] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. of SIGKDD*, 2001.
- [33] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, July 1945.
- [34] P. Dondio, S. Barrett, S. Weber, and J.-M. Seigneur. Extracting trust from domain analysis: A case study on the wikipedia project. In *Proc. of ATC*, 2006.

- [35] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the essence: improving recency ranking using twitter data. In *Proc. of WWW*, 2010.
- [36] D. Downey, M. Broadhead, and O. Etzioni. Locating complex named entities in web text. In *Proc. of IJCAI*, 2007.
- [37] G. Dupret. Latent concepts and the number orthogonal factors in latent semantic analysis. In *Proc. of SIGIR*.
- [38] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of SIGKDD*, pages 226–231, 1996.
- [39] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entites in twitter data with crowdsourcing. In *Proc. of the Workshop on Creating Speech and Language Data With Mechanical Turk at NAACL-HLT*, 2010.
- [40] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of ACL*, 2005.
- [41] G. P. C. Fung, J. X. Yu, H. Liu, and P. S. Yu. Time-dependent event hierarchy construction. In *Proc. of SIGKDD*, 2007.
- [42] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *Proc. of VLDB*, 2005.
- [43] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge. In *Proc. of AAAI*, 2006.

- [44] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of IJCAI*, 2007.
- [45] Y. Ganjisaffar, S. Javanmardi, and C. Lopes. Review-based ranking of wikipedia articles. In *Proc. of CASON*, 2009.
- [46] J. Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, Dec 2005.
- [47] K. Gimpel, N. Schneider, B. O’Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proc. of ACL*, 2011.
- [48] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32, 2006.
- [49] M. Grineva, M. Grinev, and D. Lizorkin. Extracting key terms from noisy and multitheme documents. In *Proc. of WWW*, 2009.
- [50] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.
- [51] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *Proc. of WWW*, 2007.
- [52] B. Han and T. Baldwin. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proc. of ACL*, 2011.
- [53] X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proc. of CIKM*, 2009.

- [54] D. Hasan Dalip, M. André Gonçalves, M. Cristo, and P. Calado. Automatic quality assessment of content created collaboratively by web communities: a case study of wikipedia. In *Proc. of JC DL*, 2009.
- [55] Q. He, K. Chang, and E.-P. Lim. Analyzing feature trajectories for event detection. In *Proc. of SIGIR*, 2007.
- [56] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*, 1992.
- [57] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarking improve web search? In *Proc. of WSDM*, 2008.
- [58] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *Proc. of SIGIR*, 2008.
- [59] T. Hofmann. Probabilistic latent semantic indexing. In *Proc. of SIGIR*, 1999.
- [60] S. Homma, K. Aikawa, and S. Sagayama. Improved estimation of supervision in unsupervised speaker adaptation. In *Proc. of ICASSP*, 1997.
- [61] J. Hu, L. Fang, Y. Cao, H.-J. Zeng, H. Li, Q. Yang, and Z. Chen. Enhancing text clustering by leveraging wikipedia semantics. In *Proc. of SIGIR*, 2008.
- [62] J. Hu, G. Wang, F. Lochovsky, J.-t. Sun, and Z. Chen. Understanding user’s query intent with wikipedia. In *Proc. of WWW*, 2009.
- [63] M. Hu, E.-P. Lim, A. Sun, H. W. Lauw, and B.-Q. Vuong. Measuring article quality in wikipedia: models and evaluation. In *Proc. of CIKM*, 2007.
- [64] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou. Exploiting wikipedia as external knowledge for document clustering. In *Proc. of SIGKDD*, 2009.

- [65] A. Huang, D. N. Milne, E. Frank, and I. H. Witten. Clustering documents using a wikipedia-based concept representation. In *Proc. of PAKDD*, 2009.
- [66] L. Huang, D. Milne, E. Frank, and I. H. Witten. Learning a concept-based document similarity measure. *Journal of the American Society for Information Science and Technology*, 2012.
- [67] P. Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.
- [68] R. A. Jarvis and E. A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Trans. Comput.*, 22(11):1025–1034, Nov. 1973.
- [69] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proc. of SIGKDD*, 2002.
- [70] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *Proc. of CIKM*, 2005.
- [71] D. W. Johnson and F. P. Johnson. *Joining together : group theory and group skills*. Allyn & Bacon, 8th edition, 2002.
- [72] J. Kamps and M. Koolen. The importance of link evidence in wikipedia. In *Proc. of ECIR*, 2008.
- [73] J. Kamps and M. Koolen. Is wikipedia link structure different? In *Proc. of WSDM*, pages 232–241, 2009.
- [74] L. Kassner, V. Nastase, and M. Strube. Acquiring a taxonomy from the german wikipedia. In *Proc. of LREC*, 2008.

- [75] J. Kazama and K. Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Proc. of EMNLP-CoNLL*, 2007.
- [76] A. Kittur and R. E. Kraut. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proc. of CSCW*, 2008.
- [77] A. Kittur, B. Suh, B. A. Pendleton, and E. H. Chi. He says, she says: Conflict and coordination in wikipedia. In *Proc. of SIGCHI*, 2007.
- [78] J. Kleinberg. Bursty and hierarchical structure in streams. In *Proc. of SIGKDD*, 2002.
- [79] D. E. Knuth. *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.
- [80] M. Komachi, T. Kudo, M. Shimbo, and Y. Matsumoto. Graph-based analysis of semantic drift in espresso-like bootstrapping algorithms. In *Proc. of EMNLP*, 2008.
- [81] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proc. of WWW*, 2010.
- [82] M.-T. Le, H.-V. Dang, E.-P. Lim, and A. Datta. Wikinetviz: Visualizing friends and adversaries in implicit social networks. In *Proc. of ISI*, 2008.
- [83] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proc. of SIGDOC*, 1986.
- [84] C. Li, A. Datta, and A. Sun. Mining latent relations in peer-production environments: a case study with wikipedia article similarity and controversy. *Social Netw. Analys. Mining*, 2(3):265–278, 2012.

- [85] C. Li, A. Sun, and A. Datta. A generalized method for word sense disambiguation based on wikipedia. In *Proc. of ECIR*, 2011.
- [86] Y. Li, K. Huang, S. Tsuchiya, F. Ren, and Y. Zhong. Exploring words with semantic correlations from chinese wikipedia. In *Intelligent Information Processing IV, 5th IFIP International Conference on Intelligent Information Processing*, IFIP, pages 103–108. Springer, 2008.
- [87] M. D. Lieberman and J. Lin. You are where you edit: Locating wikipedia contributors through edit histories. In *Proc. of ICWSM*, 2009.
- [88] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *Proc. of ACL*, 2011.
- [89] C. Lu, X. Hu, X. Chen, J.-R. Park, T. He, and Z. Li. The topic-perspective model for social tagging systems. In *Proc. of KDD*, 2010.
- [90] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of Berkeley Symposium on Mathematical Statistics and Probability*, pages 1:281–297, 1967.
- [91] P. Malo, A. Sinha, J. Wallenius, and P. Korhonen. Concept-based document classification using wikipedia and value function. *Journal of the American Society for Information Science and Technology*, 62(12):2496–2511, 2011.
- [92] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [93] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

- [94] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proc. of HT*, 2006.
- [95] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *Proc. of SIGMOD*, 2010.
- [96] O. Medelyan, E. Frank, and I. H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proc. of EMNLP*, 2009.
- [97] O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. In *Proc. of AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.
- [98] O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. In *Proc. of the first AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.
- [99] R. Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proc. of HLT*, 2005.
- [100] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proc. of CIKM*, 2007.
- [101] J. S. Mill. *On Liberty*. Penguin Classics, 1982.
- [102] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proc. of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence*, 2008.
- [103] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proc. of CIKM*, 2008.
- [104] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proc. of CIKM*, 2008.

- [105] D. Myers. *Social Psychology*. McGraw-Hill, 10th edition, 2009.
- [106] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39(2-3):103–134, May 2000.
- [107] S. Nunes, C. Ribeiro, and G. David. Wikichanges: exposing wikipedia revision activity. In *Proc. of WikiSym*, 2008.
- [108] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.
- [109] PearAnalytics. Twitter study, August 2009. Available online <http://www.pearanalytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf>.
- [110] P. Pecina and P. Schlesinger. Combining association measures for collocation extraction. In *Proc. of COLING/ACL*, 2006.
- [111] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. In *Proc. of HLT-NAACL*, 2010.
- [112] S. Phuvipadawat and T. Murata. Breaking news detection and tracking in twitter. In *Proc. of WI-IAT*, 2010.
- [113] B. Poblete, R. Garcia, M. Mendoza, and A. Jaimes. Do all birds tweet the same?: characterizing twitter around the world. In *Proc. of CIKM*, 2011.
- [114] A.-M. Popescu, M. Pennacchiotti, and D. Paranjpe. Extracting events and event descriptions from twitter. In *Proc. of WWW*, 2011.

- [115] M. Potthast, B. Stein, and R. Gerling. Automatic vandalism detection in wikipedia. In *Proc. of ECIR*, 2008.
- [116] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In *Proc. of EMNLP*, 2009.
- [117] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina. Clustering the tagged web. In *Proc. of WSDM*, 2009.
- [118] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proc. of CoNLL*, 2009.
- [119] L.-A. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proc. of ACL*, 2011.
- [120] S. Rendle, L. Balby Marinho, A. Nanopoulos, and S.-T. Lars. Learning optimal ranking with tensor factorization for tag recommendation. In *Proc. of SIGKDD*, 2009.
- [121] A. E. Richman and P. Schone. Mining wiki resources for multilingual named entity recognition. In *Proc. of ACL*, 2008.
- [122] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *Proc. of EMNLP*, 2011.
- [123] L. Rui, L. Kin, K. Ravi, and C. Kevin. Tedas: a twitter based event detection and analysis system. In *Proc. of ICDE*, 2012.
- [124] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. of WWW*, 2010.

- [125] C. Santamaría, J. Gonzalo, and J. Artiles. Wikipedia as sense inventory to improve diversity in web search results. In *Proc. of ACL*, 2010.
- [126] U. Scaiella, P. Ferragina, A. Marino, and M. Ciaramita. Topical clustering of search results. In *Proc. of WSDM*, 2012.
- [127] P. Schone and D. Jurafsky. Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proc. of EMNLP*, 2001.
- [128] S. Sen, S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl. tagging, communities, vocabulary, evolution. In *Proc. of CSCW*, 2006.
- [129] M. Shirakawa, K. Nakayama, E. Aramaki, T. Hara, and S. Nishio. Relation extraction between related concepts by combining wikipedia and web information for japanese language. In *Proc. of AIRS*, 2010.
- [130] J. Silva, Z. Kozareva, V. Noncheva, and G. Lopes. Extracting named entities. a statistical approach. In *Proc. of TLAN*, 2004.
- [131] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.-C. Lee, and C. L. Giles. Real-time automatic tag recommendation. In *Proc. of SIGIR*, 2008.
- [132] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proc. of AAAI*, 2006.
- [133] J. Su, J. S. Shirab, and S. Matwin. Large scale text classification using semi-supervised multinomial naive bayes. In *Proc. of ICML*, 2011.
- [134] F. M. Suchanek, M. Vojnovic, and D. Gunawardena. Social tags: meaning and suggestions. In *Proc. of CIKM*, 2008.

- [135] B. Sun and V. T. Ng. Identifying influential users by their postings in social networks. In *Proc. of MSM*, 2012.
- [136] Y. Sun, J. Han, J. Gao, and Y. Yu. itopicmodel: Information network-integrated topic modeling. In *Proc. of ICDM*, 2009.
- [137] S. Tamagawa, S. Sakurai, T. Tejima, T. Morita, N. Izumi, and T. Yamaguchi. Learning a large scale of ontology from japanese wikipedia. In *Proc. of WI-IAT*, 2010.
- [138] B. Tan and F. Peng. Unsupervised query segmentation using generative language models and wikipedia. In *Proc. of WWW*, 2008.
- [139] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., 2005.
- [140] J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: a comparison of microblog search and web search. In *Proc. of WSDM*, 2011.
- [141] R. Tinati, L. Carr, W. Hall, and J. Bentwood. Identifying communicator roles in twitter. In *Proceedings of the 21st international conference companion on World Wide Web, WWW '12 Companion*, 2012.
- [142] D. Turdakov and P. Velikhov. Semantic relatedness metric for wikipedia concepts based on link analysis and its application to word sense disambiguation. In *Proceedings of the SYRCODIS 2008 Colloquium on Databases and Information Systems*, volume 355, 2008.
- [143] P. Turek, A. Wierzbicki, R. Nielek, and A. Datta. WikiTeams: How Do They Achieve Success? *IEEE Potentials*, 30, 2011.

- [144] P. Turek, A. Wierzbicki, R. Nielek, A. Hupa, and A. Datta. Learning about the quality of teamwork from wikiteams. In *Proc. of SocialCom/PASSAT*, 2010.
- [145] B.-Q. Vuong, E.-P. Lim, A. Sun, M.-T. Le, and H. W. Lauw. On ranking controversies in wikipedia: models and evaluation. In *Proc. of WSDM*, 2008.
- [146] K. Wang, C. Thrasher, E. Viegas, X. Li, and P. Hsu. An overview of microsoft web n-gram corpus and applications. In *Proc. of HLT-NAACL*, 2010.
- [147] P. Wang and C. Domeniconi. Building semantic kernels for text classification using wikipedia. In *Proc. of SIGKDD*, 2008.
- [148] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *Proc. of SIGKDD*, 2007.
- [149] Y. Wang. Annotating and recognising named entities in clinical notes. In *Proc. of the ACL-IJCNLP 2009 Student Research Workshop*, 2009.
- [150] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proc. of SIGIR*, 2006.
- [151] J. Weng and B.-S. Lee. Event detection in twitter. In *Proc. of ICWSM*, 2011.
- [152] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proc. of WSDM*, 2010.
- [153] R. West, D. Precup, and J. Pineau. Completing wikipedia’s hyperlink structure through dimensionality reduction. In *Proc. of CIKM*, 2009.
- [154] D. Willett, A. Worm, C. Neukirchen, and G. Rigoll. Confidence measures for hmm-based speech recognition. In *Proc. of ICSLP*, 1998.

- [155] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on twitter. In *Proc. of WWW*, 2011.
- [156] Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proc. of SIGIR*, 2009.
- [157] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. In *Proc. of SIGIR*, 2008.
- [158] Y. Yang, T. Ault, T. Pierce, and C. W. Lattimer. Improving text categorization methods for event tracking. In *Proc. of SIGIR*, 2000.
- [159] Y. Yang, J. G. Carbonell, R. D. Brown, T. Pierce, B. T. Archibald, and X. Liu. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems*, 14(4):32–43, July 1999.
- [160] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *Proc. of SIGIR*, 1998.
- [161] J. Yao, B. Cui, Y. Huang, and Y. Zhou. Bursty event detection from collaborative tags. *World Wide Web*, 15(2):171–195, Mar. 2012.
- [162] E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa. Wikiwalk: random walks on wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, 2009.
- [163] D. Yin, Z. Xue, L. Hong, and B. D. Davison. A probabilistic model for personalized tag prediction. In *Proc. of SIGKDD*, 2010.
- [164] M. Yoshida, M. Ikeda, S. Ono, I. Sato, and H. Nakagawa. Person name disambiguation by bootstrapping. In *Proc. of SIGIR*, 2010.

- [165] T. Zesch, I. Gurevych, and M. Mühlhäuser. Comparing wikipedia and german word-net by evaluating semantic relatedness on multiple datasets. In *Proc. of NAACL*, 2007.
- [166] Y. Zhang, A. Sun, A. Datta, K. Chang, and E.-P. Lim. Do wikipedians follow domain experts?: a domain-specific study on wikipedia knowledge building. In *Proc. of JCDL*, 2010.
- [167] P. Zhao, J. Han, and Y. Sun. P-rank: a comprehensive structural similarity measure over information networks. In *Proc. of CIKM*, 2009.
- [168] G. Zhou and J. Su. Named entity recognition using an hmm-based chunk tagger. In *Proc. of ACL*, 2002.
- [169] G. K. Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Hafner Pub. Co, 1949.
- [170] A. Zubiaga, A. P. García-Plaza, V. Fresno, and R. Martínez. Content-based clustering for tag cloud visualization. In *Proc. of ASONAM*, 2009.