User-level Sentiment Analysis on Microblogging by Incorporating Target Specific Knowledge and User Network

Mr. Yongyos Kaewpitakkun

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

User-level Sentiment Analysis on Microblogging by Incorporating Target Specific Knowledge and User Network

Mr. Yongyos Kaewpitakkun

Supervisor: Professor Kiyoaki, SHIRAI

School of Information Science
Japan Advanced Institute of Science and Technology

September, 2016

Abstract

Microblogging services have been becoming increasingly popular over the last decade. Many people express their opinions and feeling about anything in the famous microblogging service, Twitter. These people's opinions can be grabbed easily and publicly through the interface provided by Twitter. Both individuals and organization are increasingly using this data for decision making. Customers want to know the opinion of other users before making purchase decision. Companies want to know the feedback from users about their products and also their competitors. Therefore, opinion mining and sentiment analysis become one of the major research topics in the field of natural language processing. Early work on the sentiment analysis proposed methods of classifying the sentiment on the traditional social network, i.e. forum, webboard and review. However, it is more difficult to analyze sentiments on tweets. Tweets are very short and contain a lot of informal expressions, i.e. slang, emoticon, typographical errors and a lot of words that are not compiled in a dictionary. The solution and method of traditional sentiment analysis system cannot be applied directly because of these unique characteristic of the microblogging. Moreover, existing sentiment analysis approaches mainly focus on measuring the sentiment of individual tweets or predict the massive opinions for a specific target. However, identification of the opinions of individual users is another important task that is often required in practical opinion mining systems.

In this thesis, we research and develop several methods of classifying the sentiments on microblogging, aiming to address the problems described above. We do not only focus on classifying the sentiment of each tweet by considering the textual information, which is usually short and hard to interpret. We aim to seek other characteristics in microblogging to extract the extra knowledge for boosting the performance of the sentiment analysis. Three main sentiment analysis tasks are considered, namely tweet-level sentiment analysis, target-dependent sentiment analysis, and user-level sentiment analysis. In the following, we describe each analysis one-by-one.

First, in the tweet-level sentiment analysis, we introduce a hybrid approach that uses a

lexicon for sentiment words to alleviate the data sparseness problem inherent in machine learning approaches and improve sentiment classification in tweets. The data sparseness problem can be reduced by the following two methods. We first estimate the potential polarity of objective and out-of-vocabulary (OOV) words and use these words as additional information of the existing sentiment lexicon. The polarity scores of OOV words are estimated based on the assumption that the polarities of words are coincident with the polarity of their associated sentences, using a collection of labeled sentences with their polarity. Then, we introduce a novel feature weighting method by interpolating sentiment lexicon score into uni-gram score in the feature vectors of SVM.

Second, in the target-dependent sentiment analysis, we propose a method for incorporating on-target sentiment information and user sentiment information into a machine learning classifier for the target-dependent sentiment analysis of the tweets. Three extra resources, the add-on lexicon, the extended target list, and the competitors list, are automatically constructed from the unlabeled tweets. The target specific training data is created based on heuristic rules and the lexicon-based sentiment analysis method. Two new features for training the sentiment classifier are introduced. One is the on-target sentiment feature, giving greater weight to the sentiments of the words near the target; the other is the user sentiment feature that captures the tendency of the sentiment expressed by the same user.

Third, in the user-level sentiment analysis, we propose a novel graph-based method that incorporates the information of both textual information, as well as the explicit and implicit relationships between the users, into a heterogeneous factor graph for the sentiment analysis of the tweets at the user level. Our framework takes into consideration not only the explicit connections such as follow, mention and retweet but also the implicit connections between users. An implicit connection refers to the relations of users who share similar topics of interest. The implicit relations among the users are extracted from their historical tweet corpus. Since the presence of the explicit relations in some social network is limited, the implicit relations allow us to utilize the data in social network more effectively. We also propose a new enhanced pooling method, "Hashtag-PMI", to more precisely infer the latent topics by the conventional LDA (Latent Dirichlet Allocation) from the tweet corpus.

Both public and real-life tweet corpora are used in our experiments. The results of experiments show that our method achieves 64-70%, 59-62% and around 65% accuracy on the tweet-level, target dependent and user-level sentiment analysis tasks respectively. The proposed method is effective and significantly improves the performance compared to the several baselines and existing methods.

Keywords: Sentiment Analysis, Machine Learning, Knowledge Acquisition, Topic Modeling, Microblogging

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Sentiment analysis is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes [33]. Sentiment analysis has become an important task because a large amount of user-generated content is published over the Internet. The existing sentiment analysis approaches mainly focus on identifying the opinion of the user on review data such as product and movie reviews. The lexicon-based and machine learning-based methods are popular in the previous work achieving the successful results [53, 80, 27, 21].

More recently, microblogging services, such as Twitter[1], have become a popular data source in the domain of sentiment analysis because of its efficient and low-cost for preparing a large data set. The increasingly popular use of microblogging services drastically raises the numbers of messages posted by users. The statistics indicate that there are 500 million tweets per day[2] and 304 million monthly active users[3]. As for consumers, the microblogging messages can lead to the decision making for buying or ignoring something. Nowadays, a consumer believes the opinions expressed by other consumers more than an advertising from the sellers. As for enterprises, they can improve the quality of products and services by analyzing a true voice of their customers. Microblogging is a

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

[2] http://www.internetlivestats.com/twitter-statistics/

[3] http://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

rich resource for extracting this useful information. Moreover, the microblogging services are the domain-mixing environment that allow users to post anything that they want without restrictions. It contains many user sentiment expressions either positive or negative to many different topics and domains. In addition, the users on the microblogging are different in background and preferences as well as topics the users are interested in. Therefore, it is a valuable source for flexibly analyzing people's feeling in a wide range of topics, group of users and languages.

However, automatically analyzing sentiments on tweets is still difficult because tweets are very short and contain slang, informal expressions, emoticons, typographical errors and many words not found in a dictionary. These characteristics negatively influence the performance of sentiment classifier. Many researchers have adopted the existing sentiment analysis methods of normal text, such as machine learning and lexicon based approaches, for sentiment analysis on Twitter. Most of these approaches aim at identifying the sentiment of the tweet, but not that addressed to a specific target in the tweet [55, 6, 7, 14, 68]. In other words, they classify the sentiment of the tweet, not of the target. In Twitter, however, it is common that a user expresses several sentiments in one tweet, or shows the sentiment about things other than the target and is neutral to the target. For example, the sentiment of the tweet "ugh I hate jazmin she got an iPhone 6s" is clearly negative at the tweet level, but neutral to the target "iPhone". These target-independent approaches may be insufficient for the practical use of sentiment analysis, since it is often required to know the sentiments towards a specific target, such as a product, brand, or person. For example, the users may want to know the opinion of other people about the products they are interested in, before making a purchase decision. Companies also want to know the opinions of potential users with respect to a product.

Moreover, existing sentiment analysis approaches mainly focused on the classification based on textual information and ignore the information of network relations. Many individual users' tweets are difficult to classify, but their overall opinion can be determined by considering their related tweets and their social relations. In fact, identification of the opinions of individual users is important. Such user's voice is really essential for supporting management of the enterprise to improve their brand royalty, as well as for the other customers' decision making of buying or ignoring the products.

In this thesis, we research and develop several methods of classifying the sentiments on microblogging, mainly aiming to address the problems described above. We do not only focus on sentiment classification of each tweet by considering the textual information, which is usually not enough for using in many practical systems. We aim to seek other characteristics in microblogging to extract the extra knowledge for improvement of the performance of the sentiment analysis classifier. Three kinds of extra knowledge are extracted and incorporated to the sentiment classifier to produce reliable and robust results. First, the sentiment tendency of words that have no sentiment or are not contained in the public sentiment lexicon should be revised based on the co-occurrence between words and sentiment of the sentences. Since more than 90% of words in the famous public sentiment lexicon, SentiWordNet [4], are objective words and many words in microblogging are not contained in it, this information is very important and significantly reduces the data sparseness problem. Second, the target-specific knowledge, such as target-specific training data, extended target lists and competitor lists, are extracted via heuristic, statistical and lexicon-based methods. This information is useful to help the sentiment classifier to predict the sentiment toward a certain target. Third, we also consider the link information between users in microblogging. We do not only consider the explicit relations, such as friend in Facebook or follow in Twitter, but also the implicit relations extracted by analysis of users' past tweets. Whereas the presence of the explicit relations in social networks are limited, the implicit relations allow us to utilize the data in social networks more effectively.

Twitter is used as a representative case study of microblogging services in this thesis because of three main reasons. First, Twitter becomes one of the popular social network services containing a very huge data (around 200 billion tweets per year). Second, due to the privacy policy of Twitter, the fact that the default privacy setting of tweet messages is set to public, more user generated texts are available than other social network service, i.e. Facebook. Twitter allows us to capture the past tweets of the users easily through the interface and API provided by them. Third, since many previous work on sentiment analysis on microblogging uses Twitter as the case study, conducting the experiment on the Twitter data enable us to compare our proposed method with the previous approaches. Please note that the solutions proposed in this thesis are not restricted to only Twitter.

Since our proposed methods are common to the most microblogging services, we believe that many findings are transferable to other microblogging services as well.

We investigate our proposed methods and evaluate its effectiveness in multiple sentiment analysis tasks: tweet-level sentiment analysis to detect the sentiment of the individual tweet, target-dependent sentiment analysis to detect the sentiment toward a given target, and user-level sentiment analysis to detect the overall sentiment of users about a given target. In addition, we also investigate the method to estimate the implicit users' preference similarity extracted from their past tweets which enable us to utilize the major part of data in social network effectively.

1.2 Problem Statement

Microblogging services such as Twitter are much different from other traditional media (e.g. reviews, forum and blogs). They are short and contains many abbreviations, special symbols, such as emoticons, and casual expressions. These characteristics make the solution for classifying sentiment on microblogging data become more difficult. Sentiment analysis on microblogging faces many challenges caused by the following reasons:

Noisy data According to the propose of microblogging services, the users are forced to write their message within the limited space. Twitter has a 140-character limit, while Facebook allow 420 characters limit in one status post. The users often omit words in ungrammatical manner, use unfamiliar abbreviation and short emoticons to write their statement in short messages. Moreover, because it is users' private space, users usually post their message without spelling checking attention and use informal and casual expressions along with many daily chat messages. Therefore, the messages in microblogging are short and contain many slang, casual expressions, emoticons, typographical errors and many words not found in a dictionary, especially the hashtag, a special coarse-grained topic generated by users, which are created newly every minute. These characteristics make the data in microblogging be noisier than other media.

Open-domain Unlike some traditional media such as reviews, users are freely expressing their opinions with respect to any topics and domains without restriction in microblogging. However, the sentiment orientations of some words are dependent on the target or domain. The word "unpredictable" has a positive sentiment in movie domain but this word is quite negative in the stock marketing domain. Moreover, there is common that users express their opinion but not truely about a certain target. For example, let us consider the tweet "I hate when people start to tweet about political things, you're just a teenager with an iPhone." The author of this tweet expresses strong negative feeling for "a teenager", not "iPhone". It means the sentiment analysis at tweet level may give us unsatisfied results if the goal is to detect the sentiment about the targets or entities.

Lack of labeled data In many review and bulletin board websites, there is a rating system that allows users to evaluate the products or services. The users' rating can be used as gold label for training the machine learning based classifier. However, there is no such a system in Twitter. Some previous work uses the emoticon such as ":)" and ":(" as the noisy label. But, the presence of these emoticons is limited and there is common that users express their feeling opposite to the emoticon that they used, especially in the sarcasm sentence like "Nice perfume. Must you marinate in it? :)". Moreover, classifier trained from one domain usually loss their performance when applied to other domains. Therefore, it is very time consuming to label training data for every target domain, which requires much human labor.

User personalized style Because there is no restriction for posting a message in microblogging, users tend to use their own personalized sentiment words when expressing their opinion. For example, the word "small size" has a good sentiment for a person who wants a light weight phone but bad for a person who wants to see the data in the big screen. Moreover, the word "good" may refer to the very positive feeling for one user, but a little bit positive for another user. Therefore, the user-sentiment consistency should be considered.

Although some of the above characteristics are true for other traditional media such as blogs, the nosiness caused by the shortage of the messages is the most serious problem. Anyway, it is necessary to explore special techniques to deal with these characteristics for sentiment analysis on microblogging.

1.3 Research Objective

In order to capture the users' sentiment and mood expressed in microblogging, several researches on sentiment analysis for Twitter are proposed [55, 6, 7, 20, 52, 50, 12, 5, 28, 14, 68, 9, 75]. Several practical open-source Twitter sentiment analysis tools are also developed and their performances are comparable to that of a commercial software [61]. Sentiment140[4], proposed by a group of students in Stanford University, applied distant supervision techniques to tracking the sentiment of a given target. SentiStrength[5] adopted the enhanced lexicon-based approach to detect the sentiment strength in the social web. SenticNet[6] combined the common-sense reasoning, psychology, linguistics, and machine learning for analyzing and summarizing sentiments and emotion of a given target on Twitter.

However, these approaches mainly focus on measuring the sentiment of individual tweets or recognizing the massive opinion for a specific target, but not consider the information of users who expressed the opinion[7]. The information of 'users' and 'their opinion on a specific target' are demanded in many practical situations and more difficult to capture, compared to sentiment analysis of the individual tweet or target. The enterprise can use this information for supporting a management strategy such as finding a target customer or improving their brand royalty.

The work presented in this thesis addresses the problem of detecting the sentiment of users about a specific topic. In others words, our end goal is to figure out *"What people think about X"*, where X can be a target such as brand, product, company or celebrity. This outcome is very important and a final objective in many opinion mining systems as mentioned above. In order to achieve our final goal, the most straightforward solution is adopting the tweet-level sentiment classifier to classify each user's tweet one by one and using a majority vote approach to infer the sentiment of the user on the topic. However, the individual tweets are usually ambiguous and hard to interpret. Let

[4] http://www.sentiment140.com/

[5] http://sentistrength.wlv.ac.uk/

[6] http://sentic.net/

[7] A number of methods have been proposed to extract opinion holders, users who expressed an opinion in question, from given texts. However, the studies introduced in the previous paragraph did not pay attention to the users to identify the polarity of the texts.

us consider an example “Watching Obama debate. I still don’t see any strength points of GOP candidate for our president!!.” To infer the positive sentiment to “Obama” in this tweet, we need target-specific knowledge that GOP refers to the Republican Party which is considered as a competitor of Obama. To precisely judge the polarity expressed to the given target, a sentiment analyzer should have target-specific knowledge. In other words, the sentiment analyzer should be optimized for the given target. In addition, only the textual information posted by the users may not be effective enough for analysis of the overall users’ opinion. The network relationship between the users can be also used as a clue to figure out the true sentiment of the users, since the similar users often tend to be connected in the social network [2].

Therefore, we propose an approach (1) automatically acquiring target-specific knowledge and (2) employing both textual information and user-network information created in Twitter for user-level sentiment analysis.

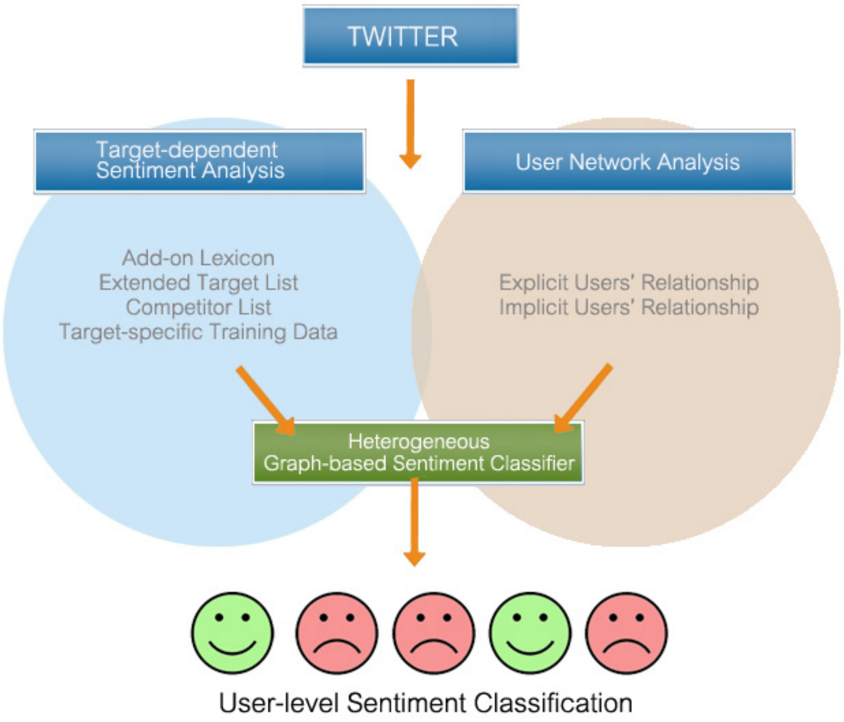

TWITTER
Target-dependent Sentiment Analysis
User Network Analysis
Add-on Lexicon
Extended Target List
Competitor List
Target-specific Training Data
Explicit Users' Relationship
Implicit Users' Relationship
Heterogeneous Graph-based Sentiment Classifier
User-level Sentiment Classification

Figure 1.1: An overview of the proposed user-level sentiment analysis system

Figure 1.1 shows the overview of the proposed method. In Target-dependent Sentiment Analysis part, an add-on lexicon, extended target list, competitor list and target-specific training data are automatically constructed. These are used to infer the polarity of individual tweets toward the given target. In User Network Analysis part, explicit and implicit users' relationship are derived from Twitter corpus. These network information as well as the textual information obtained by target-dependent sentiment analysis are integrated into Heterogeneous Graph-based Sentiment Classifier. It identifies the polarity of each user toward the given target.

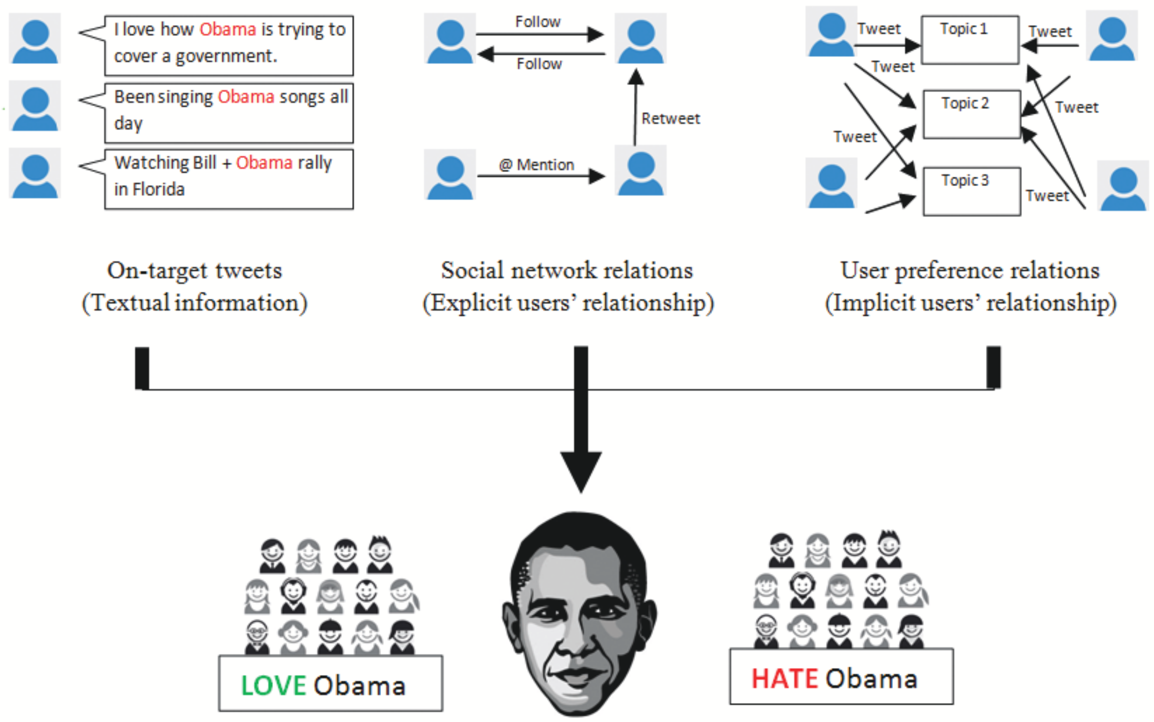

I love how Obama is trying to cover a government.
Been singing Obama songs all day
Watching Bill + Obama rally in Florida
Follow
Follow
Retweet
@ Mention
Tweet
Topic 1
Tweet
Tweet
Topic 2
Tweet
Tweet
Topic 3
Tweet
On-target tweets
(Textual information)
Social network relations
(Explicit users' relationship)
User preference relations
(Implicit users' relationship)
LOVE Obama
HATE Obama

Figure 1.2: An example of the input and expected outcome of the user-level sentiment analysis system

Figure 1.2 gives an example of user-level sentiment prediction taking the textual tweet information, social relations and user preference relation into consideration. Note that we call the social network relation such as retweet, follow and mention as the explicit relation, and the user preference relation, which is the connection between the users who are interested in similar topics, as the implicit relation. The Twitter users will be classified if they have positive or negative opinion to the target (love or hate 'Obama' in the example of Figure 1.2) using these three kinds of information.

In the rest of this chapter, we state the research questions that we address in this thesis, followed by the research methodologies, and end with the chapter organization of the thesis.

1.4 Research Questions

In order to overcome the problems mentioned before, the main research question of this thesis is shown below:

"How to build the user-level sentiment classifier from short and sparse text, without any human intervention, which is able to incorporate prior knowledge extracted from historical tweet corpus and network relationship?"

In order to achieve the objective, this research aims to investigate the following research questions:

Q1: How to overcome the data sparseness problem due to the informal language usage. One possible way to overcome this problem is to revise the polarity of words that "originally" have no sentiment in the public sentiment lexicon or are not contained in the public lexicon, which are called out-of-vocabulary (OOV) words. SentiWordNet or "SWN" [4] has become one of the most famous and widely used sentiment lexicon due to its huge vocabulary coverage. SentiWordNet is an extended version of WordNet[8], where words and synsets in WordNet are augmented with their sentiment score. SWN 3.0 contains more than 100,000 synsets. However, more than 90% of them are classified as objective words [23]; which are usually considered less important in the classification process. Furthermore, lexicon-based sentiment analysis over Twitter faces several challenges due to the short informal language used. Tweets are usually short and contain lots of slang, emoticons, abbreviations or mistyped words. Most of them are not contained in the public lexicon. Both objective and OOV words may have implicit sentiment, especially in some specific domains or group of users; thus, it could be better to modify an existing public sentiment lexicon, such as SentiWordNet, by incorporating the polarity of objective and OOV words. One possible way to revise SentiWordNet is to estimate the polarity scores of sentiment unknown words based on the polarity of the sentences including them in the corpus. The solution to solve this problem is presented in Chapter 4.

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

Q2: How to develop effective methods to predict the sentiment toward a certain target. In twitter, it is common that users may express several sentiments in one tweet or express the sentiment to other things but neutral to the target. Jiang et al. [25] reported that 40% of error of twitter sentiment analysis are because of this reason. Therefore, the target-independent approaches may get the unsatisfied result for classifying the sentiment toward some certain topic. In this thesis, we propose the approach to overcome this problem by incorporating target specific sentiment information and user-sentiment information into a machine learning classifier. First, three extra resources, an add-on lexicon, an extended target list, and a competitors list, are automatically constructed from the unlabeled tweets. Then, target-specific training data is created based on heuristic rules and the lexicon-based sentiment analysis method. Two new features for training the sentiment classifier are introduced. One is the on-target sentiment feature, giving greater weight to the sentiments of the words near the target; the other is the user sentiment feature, which captures the tendency of the sentiment expressed by the same user. Note that 'on-target sentiment' implies the sentiment about a given target, while 'user sentiment' implies the sentiment expressed by a user. The solution to solve this problem is presented in Chapter 5.

Q3: How to extract the preference similarity of users from the historical tweet corpus. In order to extract the user preference in Twitter, one possible way is to figure out *"What is a list of topics that users are interested in and usually mention to?"*. Topic modeling has been widely used to extract hidden latent topics from the document corpus. Several researches have been proposed the topic modeling methods, such as Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Indexing (PLSI) and Latent Dirichlet Allocation (LDA), and successfully discover the latent topics over the large document corpus. However, these traditional topic modeling techniques do not perform well when applied on the tweet corpus because of the sparseness and noise in the short and informal language tweets. One simple but effective way to tackle this problem is to aggregate tweets and represents them as a larger document and then train with the conventional topic models. It is called 'pooling method'. In this thesis, we propose a new enhanced pooling method, "Hashtag-PMI", to more precisely infer the latent topics by the conventional LDA from the tweet corpus. Then, the interested-related topics can be used as the input for the implicit user relationship extraction process, which will be incorporated into the sentiment analysis classifier. The solution to solve this problem is presented in Chapter 6.

Q4: How to incorporate explicit and implicit user relationship into the sentiment classification algorithm and predict the overall opinion of users about a target. In Twitter, besides of the textual information from tweet messages, there is a big part of network relationship information. It would be better if we take the advantage from this big data and use it to improve the performance of the sentiment analysis classifier. Previous approaches that have incorporated network information into a classifier have mainly focused on "a link" defined by the explicitly connected network, such as follow, mention, or retweet. However, the presence of explicit link structures in some social networks is limited. The statistics for Twitter in 2009[9] indicate that 55.50% of the users were not following anyone, 52.71% had no follower, and only 1.44% of the tweets are retweets. Therefore, in real-life situations, a large part of the social network does not contain explicit links, and so the current opinion mining systems do not derive any benefit from the network information. In order to overcome this limitation, we propose a novel graph-based framework that incorporates the "implicit connections", based on similarities between users. An implicit connection refers to the relations of users who share similar topics of interest, which is extracted from their historical tweet corpus. This will enable us to use more data for sentiment classification. To the best of our knowledge, it is the first work that incorporates both explicit and implicit relation in social network into the classifier for sentiment analysis at the user-level. The solution to solve this problem is presented in Chapter 6.

1.5 Chapter Organization

The remaining chapters of this dissertation are organized into 3 main parts.

[9] http://www.webpronews.com/wonder-what-percentage-of-tweets-are-retweets-2009-06/

Part 1: Literature Review

Chapter 2 We first describe the characteristic of the microblogging as well as the background and fundamentals of opinion mining and sentiment analysis. Then, we discuss the previous work in the area of sentiment analysis, topic modeling and social recommendation on the microblogging data.

Part 2: Sentiment Analysis on Microblogging

Chapter 3 We describe the overview and the pipeline of our methodology for sentiment analysis.

Chapter 4 We present our approach to improve the performance of the sentiment analysis at tweet-level by using the sentiment lexicon interpolation and polarity estimation of objective and out-of-vocabulary words.

Chapter 5 We present our approach to improve the performance of the target-dependent sentiment analysis by incorporating several target specific knowledge.

Chapter 6 We present our approach to improve the performance of the sentiment analysis at user-level by incorporating an implicit and explicit user similarity network.

Part 3: Discussion and Conclusion

Chapter 7 We discuss and conclude the work presented in this thesis as well as the direction for the future work.

Chapter 2

Background and Literature Review

In this chapter, we provide an overview of the previous work on sentiment analysis. First, the characteristics of the microblogging are discussed in detail. Next, the fundamental background knowledge about sentiment analysis will be discussed. Then, the work on Twitter sentiment analysis, which is the main focus of this thesis, will be explored. Finally, the previous work on topic modeling and social recommendation is introduced, since they are related to this study. In the end of each section or subsection except for 2.1 and 2.2, differences between previous work and this study are clarified.

2.1 Microblogging Characteristics

Microblogging is a web service that allows the user to broadcast short messages to other users of the service. Microposts can be made public on a web site and/or distributed to a private group of the users. The users can read microblog posts online or request that updates are delivered in real time to their desktop as an instant message or sent to a mobile device as an SMS text message. An important feature of microblogging is that the posts are brief or short, typically 140 - 200 characters[1]. Social networking service (SNS), like Facebook, also use a microblogging feature, called "Status updates", which allow users to publish short text updated in their profiles. Recently, microblogging is growing rapidly and moving to be the mainstream communication media. The number of the users in microblogging services is increasing dramatically. Therefore, the data in

[1] http://searchmobilecomputing.techtarget.com/definition/microblogging

microblogging is very valuable and too large to ignore.

Unlike the traditional social media data, such as reviews, blogs and forums, the microblogging data has special characteristics and difficulties to handle as shown below:

Short text length The text length of microblogging messages is short because of the limitation provided by the microblogging services. Due to the limitation of the text length, the user can easily write or receive their posts via a variety of platforms and devices, such as mobile phone, tablet and laptop. With the shortness of the posts, sentiment analysis becomes more difficult.

Informal language Because of the character limitation constraint, users tend to use informal language and non-standard text to emphasize the point that they want to claim within their limited messages. The abbreviation (e.g. *LOL* means laughing out loud), slang (e.g. *Headdesk* means supreme frustration), misspelling (e.g. tommorrow), emphatic lengthening (e.g. gooood) and emoticon (e.g. ^-^) are often used. Consequently, the microblogging data becomes noisy. Moreover, the special tokens used in the microblogging, such as hashtag (e.g. #iphone), mention (e.g. @John) and URL (e.g. fb.me/MZkHqr42), should be carefully treated.

Topic/domain variation Apart from the traditional social media like forums which focused on one interested domain, the microblogging services are the domain-mixing environment that allow users to post any topics in any domains without restrictions. Sometimes, even two or more topics are referred in the same message or sentence. For example, let us consider the post "800 million people don't have access to clean drinking water. 1.5 billion people have an iPhone." This post mentions two topics, which are people without clean water and people with iPhone. Moreover, the sentiment of some words are dependent on domain. For example, the words "short" is good for "restaurant queue", but bad for "battery life". This causes the problem of the sentiment ambiguity of the words, that is the sentiment of words depends on domain and context. In addition, the classifier trained from one domain usually loss their performance when applied to other domains. This can lead to problems of training data annotation for supervised learning approach.

Language style variation Because of the number of users using the microblogging service is very large, there is a mixing of users with difference background and preference as well as variety of writing styles. The microblogging post data can be very formal like *"I have recently published a short essay on the vital role that randomness plays in nature and in human life,"* or hardly to understand like *"RT @marc1919 @FBueller: FYI...2 HPD motorcycle cops #OH !!"*. Moreover, the sentiment of words may vary on the user preferences and characteristics. All of these facts can cause the problem of sentiment personalization and user-sentiment consistency.

Big and real-time data As discussed above, more than 2 billion users are using the microblogging services, and the number of posts is much larger than that. The statistics in 2015 indicate that 10 billion Facebook messages[2] and 600 million tweets are sent each day[3]. Moreover, microblogging services operate on the real-time data stream where data is transferred, viewed and updated immediately. This causes the problem of the big data analysis, which have to deal with the limitation of resources such as time and storage space.

Multi-language Nowadays, the microblogging service has become popular in various countries around the world. Unlike the reviews or forums that usually written by a single language, the microblogging post are mixed with many languages. In some countries, such as Indonesia, people use the English characters to represent their language but the meaning of words is totally different. With the short length of microblogging post, multi-language detection is more difficult.

2.1.1 Twitter Characteristics

In this thesis, Twitter is used as a representative case study of microblogging services. However, our proposed solutions are transferable to other microblogging services as well. This subsection describes the special characteristics and definitions of Twitter.

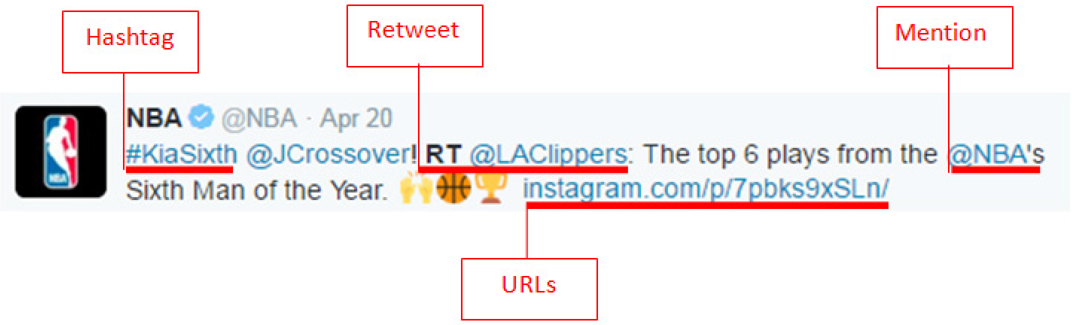
Twitter is currently one of the most popular microblogging services that allows their users to post the short messages, called *tweets*, which are displayed to the *follower* on

[2] http://blog.wishpond.com/post/115675435109/40-up-to-date-facebook-facts-and-stats

[3] http://www.internetlivestats.com/twitter-statistics/

their *timeline* in a real-time manner via Twitter website[4]. The length of tweet messages is limited to 140 characters. The default privacy setting for tweets is set to public but users can change the privacy so that the messages are displayed to just their followers. Users can describe about themselves within 160 characters that appears in a user profile, call *bio*. Unlike friend in Facebook which is a symmetric relationship, the connection in Twitter, called *follow*, can be considered as an asymmetric (one-way) relationship that may or may not be mutual. Users in Twitter are freely following other users without seeking any permission. *Following* is the action of subscription to see other Twitter user's posts. *Follower* is a person who receives other people's Twitter updates and *followee* is a person who is followed by someone. For example, when user A follows user B, Twitter refers to A as B's follower, while B as A's followee. Twitter provides a public application programming interface, called *REST APIs*[5] and *Streaming APIs*[6], which allow programmers and third-party applications to interact with the data in Twitter easily.

Twitter also provides the special language that makes users more convenient to express their feeling within a limited space. Figure 2.1 shows an example of tweet with hashtags, mentions, URLs and retweet. The following paragraphs describe these Twitter-specific languages, which users can embed in their tweet message, one by one.



NBA @NBA · Apr 20
#KiaSixth @JCrossover! RT @LAClippers: The top 6 plays from the @NBA's Sixth Man of the Year. 🙌🏀🏆 instagram.com/p/7pbks9xSLn/

Figure 2.1: An example of Twitter-specific language

[4] http://www.twitter.com

[5] https://dev.twitter.com/rest/public

[6] https://dev.twitter.com/streaming/overview

Hashtag is a word or phrase starting with the "#" symbol which freely generated by Twitter users. The hashtag can be categorized into 3 types [78]. (1) Topic hashtag, which is used for identifying the topic of a tweet annotated by a user, such as in "My headphones always do this!!! Always wonder if I'm going deaf #iphone". (2) Sentiment hashtag, which is used for expressing the user's sentiment or feeling, such as in "I hate putting a case on my iPhone its annoying #suck". (3) Topic-sentiment hashtag, which is used for expressing the sentiment about a certain topic, such as in "Since i got my iPhone i have found no need to use my laptop much #loveiphone". Moreover, the hashtag can be used for extracting the *trending topics*, a hot topic discussed by many people during a particular period of time. For example, around 150,000 tweets were sent during the NBA finals 2013 period[7] as well as at least 1,200 tweets per minute were sent with the hashtag #Tsunami in the starting period of tsunami in 2011[8].

Mention is a tweet that contains another users *@username* in the body of the tweet messages. For example, "@anthony I always do this too!!!" includes the mention @anthony. It is normally used for replying comments or referring to other users. The notification will be sent to the mentioned users. This action creates the series of conversations between users. Therefore, besides the follow-network, mentions can be also used to form a user-user relationship network via an "@-reference" [69].

URL Beside the text message, Twitter allows user to refer to the external contents, such as news or photos, via the short URLs. According to the short length of tweet, URLs enable a user to give more information about their thinking by redirecting to the website that they want. Liu et al. reported that URLs to the picture sites (e.g. instagram.com) or video sites (e.g. youtube.com) are often subjective, while other URLs (mostly linking to news articles) are usually objective [35].

Retweet is an action to repost or forward message posted by other users on Twitter. Its format is 'RT @username' where username is the twitter name of the person who wrote the message reposted by the other. Because the retweet message cannot be changed, it

[7] http://twitter.github.io/interactive/tpms/

[8] http://www.journalism.org/2011/03/17/twitter-responds-japanese-disaster/

implies that the retweet user usually agree with the content of the original tweet message [57].

2.2 Background of Opinion Mining and Sentiment Analysis

Sentiment analysis, also called opinion mining, generally aims to identify the attitude of a speaker or writer using natural language processing, text analysis and computational linguistics [56]. This area can be considered as interdisciplinary research including data mining, knowledge discovery, and computational linguistics (or natural language processing). Most of studies on sentiment analysis are focused on sentiment detection (including subjectivity and polarity classification) and emotion recognition. The subjectivity detection aims to identify whether a given text is objective (i.e. no sentiment) or subjective while polarity detection aims to classify the sentiment orientation in text into positive or negative. The emotion recognition further investigates characteristics of the texts in more detail. It aims to identify not the polarity (positive or negative) but the human emotions and feelings expressed in text, such as joy, sadness, surprise and fear [45].

Research on sentiment analysis has been investigated from different perspectives. In general, sentiment analysis can be categorized as five kinds of the tasks as follows.

Document level This task aims to determine the overall sentiment of an entire document. It is usually done by combining the sentiment of all sentences inside the document. This level considers the document as a single topic. Therefore, it is applicable for the document where the sentiment is expressed toward only one entity such as product reviews.

Sentence level This task aims to classify the sentiment of a given sentence into positive, neutral or negative. If there are both positive and negative feelings inside the sentence, a stronger expression is selected.

Target-dependent/entity-dependent This task aims to classify the sentiment toward a given target or entity. It is common that users may express several sentiments toward multiple targets. The target-dependent approach can be considered as the effective way for the practical use of sentiment analysis, since it is often required to know the sentiment towards a specific entity, such as product, brand and celebrity.

Aspect-dependent This task is fine-grained analysis toward a certain target. It aims to analyze the sentiment for each aspect or feature. For example, CPU, memory, OS, disk, screen, keyboard, battery etc. are the aspects or features of a note PC. Aspect-dependent sentiment analysis can be divided into 2 sub tasks. One is a task to extract the aspects of a given target, the other is a task to classify the sentiment of each aspect.

User level This task aims to determine the overall sentiment of a user toward a certain target. It is usually done by averaging the sentiments expressed in multiple sentences written by the same user. Several methods are proposed to enhance the result by taking the information of the users network into consideration. This task can be considered as a final objective in many opinion mining systems, since the end goal is to figure out *What people think about X*, where *X* can be any targets or entities.

In terms of data source, in the early state, there have been several attempts of sentiment analysis on general text, such as blogs [42, 48, 19] and customer reviews [54, 34, 36]. The common characteristic of such data sources is that the language in them is quite formal and well-structured. Recently, microblogging service, such as Twitter, becomes a popular data source for analyzing the public sentiment due to a large amount of user generated contents. However, the language used in microblogging text is very short and informal. Therefore, the traditional approach for conventional text may be inefficient when it is applied to microblogging data. In the next section, the detailed review of the previous work will be discussed, especially those work on Twitter, which has been mainly used as the data source in this thesis.

2.3 Sentiment Analysis on Microblogging

Recently, the sentiment analysis on microblogging like Twitter is the upcoming trend in the current studies. This section provides a detailed overview of previous studies on sentiment analysis of microblogging focusing on tweet-level, target-dependent and user-level.

2.3.1 Tweet-level Sentiment Analysis

Early work on Twitter sentiment analysis used two approaches in traditional sentiment analysis on normal texts: machine learning-based and lexicon-based approaches.

Machine learning-based approach employs supervised machine learning, such as Naive Bayes (NB), Maximum Entropy (ME) and Support Vector Machine (SVM) as the learning algorithm. This approach consists of 2 phases: training phase and predicting phase. In the training phase, a classification model is generated by learning from a set of features extracted from a training data which is usually manually labeled. Then, the sentiment labels of unseen data in test set are predicted via the trained classification model. Most of work mainly focused on feature engineering, including feature extraction and selection. Some of features that can be used for sentiment analysis on Twitter are n-grams, bag-of-word, part-of-speech (POS), lexicon, syntactic and Twitter-specific features, such as hashtag and emoticons [55, 6, 7]. The problems with this approach are (1) it needs labeled training data, which requires much human labor, and (2) the classifier trained for one domain does not usually work well for another domain.

In order to overcome the first problem, several attempts to automatically collect the training data without manual annotation, called distant supervision, were reported. The pioneer work was introduced by Go et al. [20]. They used emoticons such as ":)" and ":(" to construct a training corpus consisting of 1.6 million positive and negative tweets. They reported that SVM with uni-gram features achieved the best performance at 82.9%. Pak et al. used the similar approach to solve the problem of 3-class sentiment analysis [52]. They used emoticons as the noisy labels for collecting positive and negative tweets while several newspapers were used for generating neutral tweets. Both n-gram and POS were used as their features to compute the posterior probability in the Naive Bayes models. Beside using emoticons, Davidov et al. introduced an approach that used a hashtag to label the dataset of O'Connor et al. [50]. They used 50 hashtags such as #sucks and #notcute as well as 15 smiley emoticons as noisy labels to classify tweets into positive and negative [12]. In the similar way, Kouloumpis et al. utilized a set of hashtags to label tweets in Edinburgh Twitter corpus[9] into positive (i.e. #success), negative (i.e. #fail) and neutral (i.e. #omgfacts) [28]. Several feature sets including n-grams, lexicon, POS and microblogging features were used in their work.

In addition to use emoticons and hashtags as the sentiment indicators, Barbosa at el. proposed a slightly different approach by using the result from third-party sentiment analysis websites such as Twendz[10], TweetFeel[11], and Sentiment140[12] to create a dataset with noisy labels [5]. They proposed 2-step classification where the system first classified messages as subjective and objective, and further distinguished the subjective tweets as positive or negative. The results indicated that the meta-information of the words (negative polarity, positive polarity and verbs) were more important for the polarity detection step, while the tweet syntax features (good emoticons and upper case) were more significant for subjectivity detection. However, Kun-Lin Liu et al. [35] and Speriosu et al. [67] argued that using the distant supervision approach alone is often inaccurate and may harm the performance of sentiment classifiers. Moreover, both emoticons and hashtags are sparse for preparing a large amount of training data for some target keywords.

On the other hand, lexicon-based approach uses pre-defined external resources, such as a polarity dictionary or lexicon like SentiWordNet[13], ANEW[14], or MPQA[15] to determine the sentiment orientation in texts [14, 68]. The effectiveness of this approach highly depends on the sentiment lexicon and the algorithm to calculate the overall sentiment tendency. O'Connor et al. [50] and Bollen et al. [9] used the MPQA sentiment lexicon to detect the sentiment of tweets by simply counting whether these tweets contain more positive or negative words according to the sentiment lexicon. Brody et al. found that emphatic lengthening words, such as 'coooooool', were strongly associated with subjectivity and sentiment [9]. Therefore, they proposed a lexicon-based approach to detect the sentiment of tweets by including lengthening words as additional opinionated words to the MPQA lexicon. They applied the label propagation method on a graph of the lengthening words to calculate the final polarity of the tweets.

[9] http://demeter.inf.ed.ac.uk

[10] http://twendz.waggeneredstrom.com

[11] http://www.tweetfeel.com

[12] http://www.sentiment140.com/

[13] http://sentiwordnet.isti.cnr.it/

[14] http://neuro.imm.dtu.dk/wiki/A_new_ANEW/

[15] http://mpqa.cs.pitt.edu/

Thelwall et al. proposed a lexicon-based sentiment strength detection system on microblogging, called SentiStrength [75]. They used their own sentiment lexicons which consist of 298 positive and 465 negative terms as well as lists of emoticons, negations and boosting words. They applied several lexical rules that designed for dealing with the informal language in tweet messages. Thelwall et al. proposed the enhanced version of SentiStrength [74]. They expanded the number of terms in their sentiment lexicon to 2,310 words. The idiom lists and the strength boosting algorithm were added. The evaluation results showed that SentiStrength performed significantly above the baseline across six social web data sets, such as Twitter, YouTube and MySpace. However, the drawback of the lexicon-based approach is that it highly depends on pre-built lexicons and language models. Terms that are not included in the sentiment lexicon are usually ignored. In other words, the performance of this approach degrades drastically with the exponential growth of the lexicon size [73]. As the result, even this approach can show high precision but low recall [62].

Recently, some studies have combined these two approaches and achieved relatively better performance in two ways. The first is to develop two classifiers based on these two approaches separately and then integrate them into one system. The second is to incorporate lexicon information directly into a machine learning classification algorithm.

In the first way, Kumar et al. used a machine learning-based method to find the semantic orientation of adjectives and used a lexicon-based method to find the semantic orientation of verbs and adverbs [29]. The overall tweet sentiment is then calculated using a linear interpolation of the results from both methods.

In the second way, Saif et al. utilized knowledge of not only words but also semantic concepts obtained from a lexicon as features to train a Naive Bayes classifier [63]. Fang et al. automatically generated domain-specific sentiment lexicon and incorporated it into the SVM classifier [17]. They applied this method for identifying sentiment classification in product reviews. Mudinas et al. presented concept-level sentiment analysis system, which are called pSenti [46]. Their system used a lexicon for detecting the sentiment of words and used these sentiment words as features in the machine learning-based method. Results from both lexicon and machine learning were combined together to calculate the final overall sentiment scoring. Recently, Hung et al. reported that more than 90 percent of words in SentiWordNet are objective words that are often considered useless in sentiment classification [23]. So, they reassigned proper sentiment values and tendency of such objective words in a movie review corpus and incorporated these sentiment scores into the machine learning-based method.

In this thesis, we reevaluate the sentiment score of not only objective words but also out-of-vocabulary (OOV) words, which are common in informal language in the tweets. We also propose an alternative way to incorporate the sentiment lexicon knowledge into the machine learning algorithm. We will propose sentiment interpolation weighting method that interpolates lexicon scores into uni-gram scores in the vector representation of the SVM classifier. Our method is described in detail in Chapter 4.

2.3.2 Target-dependent Sentiment Analysis

Most of the previous approaches aim at target-independent sentiment analysis, that is, classifying not the target but the tweet according to the polarity. However, as discussed in Chapter 1, a tweet can often have two or more sentiments towards multiple targets. Therefore, target independent approaches may be inappropriate because it is often required to classify the sentiment toward a certain topic.

Chen et al. examined whether a topic dependent model improves the polarity classification of microblogging [10]. They observed that, for some topics, topic dependent models achieved significantly better performance than a general model. Jiang et al. incorporated target dependent features into the SVM classifier [25]. These features were extracted by rules based on syntactic relations in the result of the dependency parser. To boost the performance of the classification, they also used a graph based optimization by considering the sentiment labels of the related tweets. Dong proposed an Adaptive Recursive Neural Network (AdaRNN) for target dependent Twitter sentiment classification by propagating the sentiment of a word to the target based on the context and syntactic relationship in the dependency tree [15]. The common disadvantage of [25] and [15] is that it requires manually labeled training data for each specific topic. Moreover, the performance of their approach greatly depends on the dependency parser, which is not quite accurate when applied to informal language like tweets.

Alternatively, some ways of performing unsupervised target-dependent sentiment analysis on Twitter have been proposed. Chen et al. presented an optimization based approach to automatically extract sentiment expressions for a given target from a corpus of unlabeled tweets [11]. Then, they applied a lexicon based method to classify the sentiment by summing up the score of the sentiment expression extracted from the previous step. Zhang et al. proposed an entity-level sentiment analysis for Twitter by combining lexicon based and learning based methods [30]. Their method first adopted a lexicon based approach to perform an entity-level sentiment analysis. After that, additional tweets that were likely to have opinions about the given entity were automatically identified through the Chi-square test based on the association between words and sentiment label of the tweets. Then, the classifier was trained to classify the polarity of the entities in the additional tweets extracted from the previous step.

In this thesis, we present a sophisticated method to automatically create target-specific training data, not simply applying an existing sentiment classification tool to unlabeled data. We incorporate the procedures to (1) change the polarity of the tweets that are not truly related to the target into neutral, (2) invert the polarity of tweets expressing an opinion about a competitor of the target by heuristic rules and (3) automatically construct target-specific lexicon when the target-specific training data is created. This is used for training the SVM classifier with uni-gram, on-target sentiment and user sentiment features for the prediction of the sentiment to the target. Note that the use of user sentiment features is one of the advantages of the proposed method. Another difference between our work and previous approaches ([11] and [30]) is how to identify the neutral tweets. Unlike previous work, where the neutral tweets are extracted by looking up indicator words obtained by statistical methods, in our method, they are identified by machine learning. Since the majority of the errors of target-dependent sentiment classification are caused by the tweets that show the user's opinion but not about the target [25], it would be appropriate to classify neutral-to-target tweets by sophisticated machine learning rather than a simple heuristic.

2.3.3 User-level Sentiment Analysis

Although most previous work on sentiment analysis in Twitter has mainly focused on understanding the sentiments of individual messages as discussed in Subsection 2.3.1, there have been several attempts to identify the sentiment of the users. They were based on the assumption that the overall sentiment of the users can be estimated by aggregating the sentiments of the individual tweets in their history corpus [26, 65]. However, many individual tweets are difficult to classify, due to shortness and ambiguity of the meaning of the tweets. Moreover, the simple aggregation of the sentiment of the tweets may cause a lot of noise and errors. To overcome this problem, some researchers have proposed solutions that incorporate the network relation data into their model, such as 'follow', 'mention', and 'retweet'. Tan et al. used a friendship network such as that from 'follow' and from the 'mention' graph to perform a user-level sentiment analysis [69]. Pozzi et al. used the approval relation based on the retweet graph to solve the same problem, and got satisfying results [57]. Nozza et al. follow the idea of Pozzi's approval graph for improving the sentment prediction of both tweets and users at the same time [49]. Unfortunately, in some social networks, presence of such explicit link structures is limited.

In our approach, we incorporate not only explicit links but also implicit links, which will be extracted from users' historical tweet corpora, in the social network. This enables us to effectively use more information in the social network.

2.4 Topic Modeling for Short Texts

Probabilistic topic models, like Latent Dirichlet Allocation (LDA) [8], are widely used and give a successful results for discovering the hidden topics from a large collection of documents. However, previous research has found that the standard topic modeling techniques do not work well with the short and ambiguous form of tweets [82]. To overcome this problem, a number of extensions of LDA have been proposed in two directions. One is to modify the LDA mechanism to deal with short texts, such as Labeled LDA [59] or Twitter-LDA [82]. The other, which is simpler and more popular, is to aggregate the tweet messages into more lengthy documents before applying the standard LDA model. This is called the 'message pooling scheme'.

The popular message pooling schemes are to merge all tweets under the same author [22, 79], the tweets published in a similar time period, or the tweets with the same hashtag [41]. Regarding the pooling schemes, Mehrotra et al. have reported that merging the tweets sharing the same hashtag into one document performed better than other pooling schemes [41]. On the other hand, Wang et al. reported that only 14.6% of the tweets contain a hashtag, that is, the remaining un-hashtagged tweets were not used effectively in the hashtag-pooling method [78]. Mehrotra et al. [41] and Schinas et al. [64] tackled this problem by merging messages without a hashtag to the most similar document, and found that this method improved the performance of the conventional LDA and achieved the best performance. They used cosine similarity with TF-IDF to measure the textual similarity.

In the present thesis, we propose an alternative way to extract the potential relationships between tweets that should belong to the same topic. The difference between our approach and previous approaches ([41], [64]) is that instead of assigning the un-hashtagged tweets to the document with the highest textual similarity, we consider the co-occurrence between a hashtag and a term, based on Point-wise Mutual Information (PMI), which explicitly captures the relation between them.

2.5 Social Recommendation

As social networks have become so popular, some research of recommendation system, which could recommend a user an item that he/she might be interested in, combined a user-item matrix for a regular collaborative filtering (CF) and social matrix-factorization (MF) with social network analysis. Several papers reported that these methods outperformed the recommender system without social information. Since the idea of combining users' preference and social network analysis is somehow related to our proposed method, we give a brief review of them in this section.

Ma et al. introduced the Social Recommendation (SoRec) model to solve the data sparsity and poor prediction accuracy problems by employing both users' social network information and rating records [38]. One year later, the same research group proposed a recommendation system with the Social Trust Ensemble (STE) which naturally fused the users' tastes and their trusted friends' favors together [37]. Jamali et al. proposed the enhanced model-based approach for recommendation in social networks by incorporating the mechanism of trust propagation into the matrix factorization model, called SocialMF [24]. The experimental results indicated that their proposed method outperformed the SoRec and STE in term of RMSE. Ma et al. proposed a social regularization method by incorporating social network information into the training procedure of the social matrix-factorization [39]. They estimated the similarity between users by using Pearson Correlation Coefficient (PCC) and Vector Space Similarity (VSS) of commonly rated items among users. However, in the real-life situation in the social network, the number of commonly rated items between friends could be very small. To tackle this problem, Yu et al. proposed an Adaptive Social Similarity (ASS) function based on the matrix factorization technique [81]. They estimated the similarity between users based on their latent features between friends which was not lose the information even these two friends did not buy any products in common. Tang et al. proposed a fine-grained approach, called mTrust, to capture multi-faceted trust relationships between users for the tasks of rating prediction, facet-sensitive ranking, and strengthening status theory [72].

Even through the idea of the integration of users' preference and network information is roughly common to recommendation systems, however, this idea is first applied for the sentiment analysis of microblogging. Moreover, the goal of our research and the recommendation system is also different. User-level sentiment analysis is the task to detect the sentiment of users toward a specific target, while the recommendation system is the task to predict an item or content that a user might be interested in. In addition, in recommendation research work, the user's opinion of the items is extracted from the rating scale (i.e. review star), while our work relies on the classifier that is automatically trained and tuned for the specific target.

Chapter 3

Overview of Proposed Method

This chapter presents an overview of the proposed approach. The system accepts a collection of tweets, a set of users and a list of topics as input. Then, three kinds of sentiment analysis are performed: tweet-level sentiment analysis to detect the sentiment of the individual tweet, target-dependent sentiment analysis to detect the sentiment toward a given target, and user-level sentiment analysis to detect the sentiment of users about a given target. An overview of the system architecture and a pipeline of our work are shown in Figure 3.1. Chapters in this thesis that corresponds to each task are also shown. The overview of our proposed method for each task is described below.

Task 1: Tweet-level Sentiment Analysis

Our two-step hybrid sentiment analysis system has been developed by combining lexicon-based and machine learning-based approaches. In the first step, the data-oriented add-on lexicon has been created. In this thesis, we define 'add-on lexicon' as a special additional lexicon that complies specific terms with their polarity and compensates a public sentiment lexicon. It is automatically constructed by reevaluating the polarity scores of objective words and out-of-vocabulary (OOV) words extracted from a specific tweet corpus. After that, at 'Feature Extraction' step, the score from both the public lexicon (SentiWordNet; SWN) and add-on lexicon will be incorporated into a feature vector as extra prior knowledge in four different ways. At 'Supervised Classification' step, Support Vector Machine is trained from the extracted feature vectors to identify the polarity of a given tweet. The main advantage of our approach is the extra sentiment polarity in-

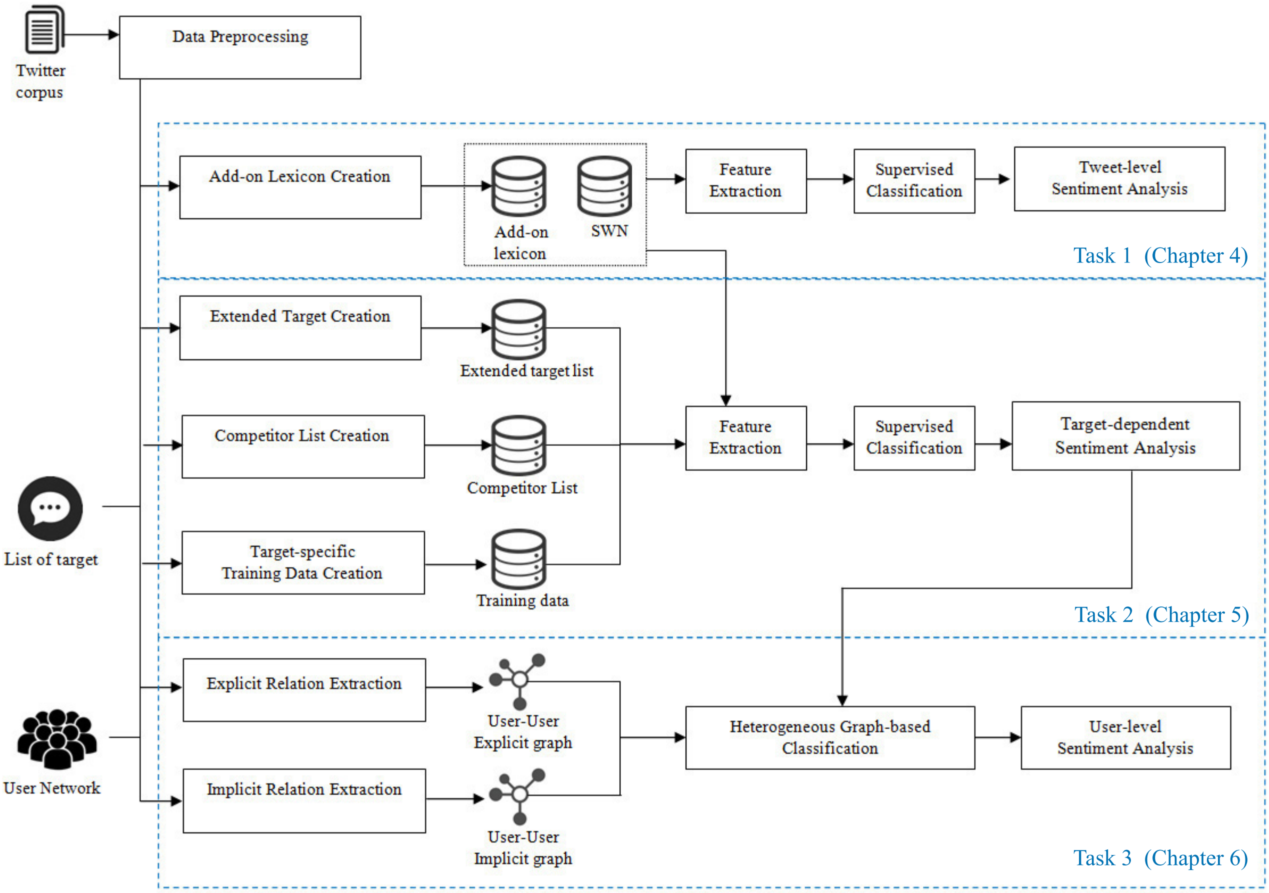


Figure 3.1: The overview of the proposed system architecture

formation from both the public and add-on lexicon will be incorporated to the powerful machine learning algorithm. It can help the supervised learned classifier to identify the sentiment of tweets more precisely, even when tweets contain words that are not found in the public lexicon or less frequently appeared in the training set. For more detail, see Chapter 4.

Task 2: Target-dependent Sentiment Analysis

We develop a method of classifying the sentiments (positive, negative or neutral) of a given target in the tweets. Our method relies on supervised machine learning. However, the proposed method does not require any human intervention, such as annotation of the labeled data. This enables us to apply our method to the sentiment analysis of various targets. Several techniques will be proposed to improve the performance of target dependent sentiment classification. First, a target specific add-on lexicon is automatically constructed. It is an additional sentiment lexicon built by automatically identifying the polarity of the objective and OOV words. Second, an extended target list and competitor list are built. These are extra target specific knowledge. Third, not general but target specific training data is constructed for learning the sentiment classifier. It is automatically created by a lexicon-based method and several heuristics with the extended target list and the competitor list from unlabeled tweets. At 'Feature Extraction' step, samples in the target specific training data are represented as the feature vectors. Note that the add-on lexicon and SWN are used in this step. Furthermore, a user sentiment feature, where the other tweets of the same user that expresses an opinion about a given target are considered, is also incorporated. At 'Supervised Classification' step, SVM is trained from the obtained feature vectors. It is applied for target-dependent sentiment analysis of a newly given tweet. For more detail, see Chapter 5.

Task 3: User-level Sentiment Analysis

We develop a method of classifying the sentiments (positive or negative) of users about a certain topic by using textual information as well as both explicit and implicit relationships between users in the social network. First, the retweet connections are extracted as explicit relations between the users. Second, the implicit relations between the users are extracted by finding similar users in terms of their interested topics. Third, the sentiment of the on-target tweets is classified by a target-dependent sentiment analysis incorporating target specific knowledge. Note that in this thesis 'on-target tweet' is defined as a tweet that addresses a given target. After that, the information about the implicit relationship, the explicit relationship and the sentiment of the on-target tweets are incorporated into a heterogeneous factor-graph model. Finally, loopy belief propagation is applied to identify the sentiment of the users. In addition, in the step of implicit relation extraction, we propose an improved method to discover latent topics in the tweets via an enhanced pooling scheme with the conventional LDA, called the Hashtag-PMI pooling scheme. Note that the whole process does not require any human intervention, such as annotation of labeled data. This enables us to apply our method to the sentiment analysis of various targets. For more detail, see Chapter 6.

In addition, some components developed in one chapter are applied to another chapter, indicated by the arrow across the chapters shown in Figure 3.1. The final goal of the thesis is to develop a system for Task 3, i.e. user-level sentiment analysis. To incorporate the textual information (the user's sentiment toward the target) into the heterogeneous graph-based model, the classifier of target-dependent sentiment analysis developed in Task 2 is used. To develop the target-dependent sentiment classifier, the target specific add-on lexicon plays an important role. It is constructed by the method used in Task 1. In this way, the different components of three sentiment analysis tasks are combined. In other words, the several techniques for user-level sentiment analysis are applied to other task. This enable us to evaluate the effectiveness of our proposed methods when they are applied for the different tasks.

We evaluate the effectiveness of our proposed methods compared to several baselines and conduct the experiments on several datasets in multiple sentiment analysis tasks, including tweet-level, target-dependent and user-level sentiment analysis, which allow us to understand the problem of sentiment analysis in different views.

From another points of view, as illustrated in Figure 3.2, each proposed method of three sentiment analysis tasks consists of the following three steps.

Step 1. Data preprocessing

Several data preprocessing process are executed to clean the raw tweet dataset. The preprocessing step consists of part-of-speech tagging, lemmatizing, and stop word and URL removal.

Step 2. Knowlege extraction

Extra knowledge is automatically extracted from a collection of the tweets.

Sentiment lexicon expansion In Task 1 and 2, we revise the polarity of the objective and OOV words in public lexicon based on the co-occurrence of words in the tweet collection.

Target-specific knowledge extraction In Task 2, by several sophisticated methods, we extract the information about a given target, such as target-specific training data, extended target and competitor list, which will be used as the extra knowledge for classifying the true sentiment about the target.

Network relationship extraction In Task 3, we extract the link information between users in Twitter, both explicit and implicit relations. The explicit relation, such as follow or retweet, can be grabbed directly through TwitterAPI; while the topic modeling based algorithm is applied to extract the implicit relation among users.

Step 3. Sentiment classification

We investigate and design several methods for classifying the sentiment of the tweets by incorporating several knowledge extracted from previous processes. The machine learning-based with feature engineering as well as graph-based algorithms are applied to produce reliable and robust sentiment analysis results.

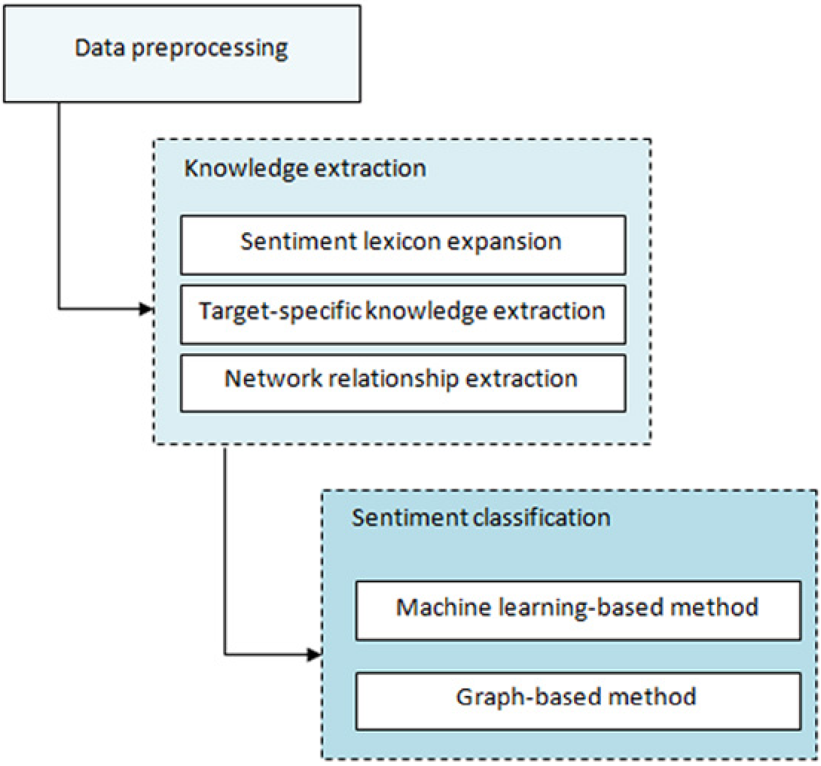

Data preprocessing
Knowledge extraction
Sentiment lexicon expansion
Target-specific knowledge extraction
Network relationship extraction
Sentiment classification
Machine learning-based method
Graph-based method

Figure 3.2: General flowchart of the proposed methods of three tasks

Chapter 4

Tweet-level Sentiment Analysis

This chapter presents an approach to improve the performance of the sentiment analysis at tweet-level. We introduce a hybrid approach that incorporates sentiment lexicons into a machine learning approach to improve sentiment classification in tweets. We automatically construct an *Add-on lexicon* that compiles the polarity scores of objective words and out-of-vocabulary (OOV) words from tweet corpora. We also introduce a novel feature weighting method by interpolating sentiment lexicon score into uni-gram vectors in the Support Vector Machine (SVM). Results of our experiment show that our method is effective and significantly improves the sentiment classification accuracy compared to a baseline uni-gram model.

4.1 Background and Motivation

There are two main approaches to sentiment analysis: lexicon-based and machine learning-based techniques. Several researchers have combined these two techniques [29, 46, 63, 17, 23]. This study adopts a similar approach; we seek to combine the prior polarity knowledge from the lexicon-based method and the powerful classification algorithm from the machine learning-based method. Two main motivations of this approach are discussed below.

The initial motivation is to revise the polarity of objective and out-of-vocabulary words in the public sentiment lexicon to improve Twitter sentiment classification. In the lexicon-based approach, sentiment classification is done by comparing the group of positive and negative words looked up from the public lexicon. For example, if the document contains more positive words than negative words, it will be classified as positive. Several public lexical resources such as ANEW[1], MPQA Subjectivity Lexicon[2], General Inquirer[3], SentiWordNet[4] and SenticNet[5] lexicon are available for this type of analysis. SentiWordNet or "SWN" [4] has become one of the most famous and widely used sentiment lexicons because of its huge vocabulary coverage. SentiWordNet is an extended version of WordNet[6], where words and synsets in WordNet are augmented with their sentiment score. SWN 3.0 contains more than 100,000 synsets. However, more than 90% of them are classified as objective words [23], which are usually considered less important in the classification process. Furthermore, lexicon-based sentiment analysis over Twitter faces several challenges due to the short informal language used. Tweets are usually short and contain lots of slang, emoticons, abbreviations or mistyped words. Most of them are not contained in the public lexicon, which are called out-of-vocabulary (OOV) words. Both objective and OOV words may have implicit sentiment, especially in some specific domains or group of users; thus, it could be better to modify an existing public sentiment lexicon, such as SentiWordNet, by incorporating the polarity of objective and OOV words. One possible way to revise SentiWordNet is to estimate the polarity scores of sentiment unknown words based on the polarity of the sentences including them in the corpus. For example, let us suppose that the objective word "birthday" appears more times in positive tweets than in neutral or negative tweets. This word could be revised as a positive word in the sentiment lexicon. On the other hand, when the OOV word "ugh" appears more times in negative tweets than in neutral or positive tweets, it could be newly classified as a negative word. In this work, we aim to build a data-oriented add-on lexicon covering the estimated polarity scores for both objective words and OOV words in the SentiWordNet.

The second motivation is to incorporate the prior polarity knowledge from the sentiment lexicon into powerful machine learning classifier, such as the Support Vector Machine (SVM), as extra information. Among many machine learning techniques, SVM

[1] http://neuro.imm.dtu.dk/wiki/A_new_ANEW/

[2] http://mpqa.cs.pitt.edu/

[3] http://www.wjh.harvard.edu/~inquirer/

[4] http://sentiwordnet.isti.cnr.it/

[5] http://sentic.net/

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

has achieved the great performance in the sentiment classification task. The uni-gram feature has been widely and successfully used in sentiment analysis, especially in user review datasets. Since tweets are much shorter than user reviews, however, the use of only the uni-gram feature may cause a data sparseness problem. One possible way to solve this problem is to integrate the information from the sentiment lexicon to supervised algorithms as extra knowledge. Recently, some researchers incorporate information derived from a lexicon into machine learning by augmenting sentiment lexicon as extra polarity group feature to uni-gram [51] or simply replacing uni-gram with a lexicon score [23]. In this work, we present an alternative way to incorporate lexical information into a machine learning algorithm by interpolating a score in the sentiment lexicon into a score of uni-gram feature in vector weighting. Our experiment results show that the proposed lexicon interpolation weighting method with revised polarity estimation of objective and OOV words is effective and significantly improves the sentiment classification accuracy compared to the baseline uni-gram model.

4.2 Proposed Method

Our proposed method for tweet-level sentiment analysis consists of two steps: one is creation of a data-oriented add-on lexicon, the other is predicting the polarity of the tweets by a supervised trained SVM classifier where the information of sentiment lexicon is incorporated. The overall system framework is shown in Figure 4.1. Note that the proposed method is based on supervised learning and the input training data is a collection of the tweets annotated with polarity tags.

4.2.1 Data Preprocessing

The data preprocessing process consists of part-of-speech (POS) tagging, lemmatizing, and stop word and URL removal. In the first step, tweets are POS tagged by the TweetNLP POS Tagger[7], which is trained specially from Twitter data. SentiWordNet contains only four open-class words: noun, verb, adjective and adverb. Therefore, we have to map the POS tag from the tagger into the SentiWordNet tag as shown in Table

[7] http://www.ark.cs.cmu.edu/TweetNLP/

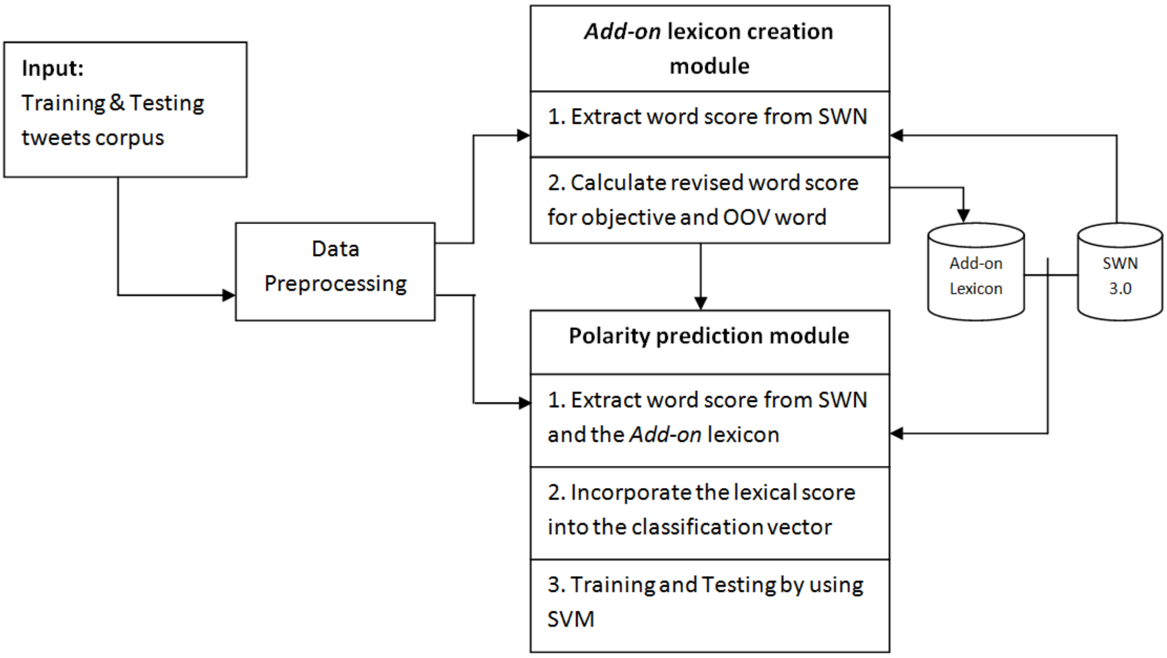

Input:
Training & Testing tweets corpus
Data Preprocessing
Add-on lexicon creation module
1. Extract word score from SWN
2. Calculate revised word score for objective and OOV word
Add-on Lexicon
SWN 3.0
Polarity prediction module
1. Extract word score from SWN and the *Add-on* lexicon
2. Incorporate the lexical score into the classification vector
3. Training and Testing by using SVM

Figure 4.1: System framework of the tweet-level sentiment analysis

4.1. The SWN tag is used as POS feature for training SVM in the polarity classification step. After that, all words are lemmatized by the Stanford lemmatizer[8]. We also reduce the number of letters that are repeated more than twice, i.e. "heellllooooo" is replaced by "heelloo". Finally, the common stop words[9] and URL are removed because they represent neither sentiment nor semantic concept.

Table 4.1: Mapping POS tag into SentiWordNet tag

POS Meaning	**POS Tag**	**SWN Tag**
Verb	VB, VBD, VBG, VBN, VBP, VBZ	V
Noun	NN, NNS, NNP,NNPS	N
Adverb	RB, RBR, RBS	R
Adjective	JJ, JJR, JJS	A
OOV words	Other remaining tags	OTHER

[8] http://nlp.stanford.edu/software/

[9] http://xpo6.com/list-of-english-stop-words/

4.2.2 Data-oriented Add-on Lexicon Creation

A data-oriented add-on lexicon is built by compiling both objective and OOV words on SentiWordNet with their newly estimated sentiment score. Word scores are estimated based on the assumption that the polarities of words are coincident with the polarity of their associated sentences, which seems reasonable due to the short length of tweet messages. In other words, if the word frequently appears in the positive (or negative) tweets, its polarity might be positive (or negative).

In the creation of the add-on lexicon, the sentiment score of a word is calculated based on the probability that the word appears in positive or negative sentences in a sentiment tagged corpus. There are two steps. In the first step, objective words and OOV words are extracted from pre-processed tweets with their SentiWordNet score $SWNScore$. It is defined as Equation (4.1). $SWNScore_{POS}$ is the positive score while $SWNScore_{NEG}$ is the negative score of words in SentiWordNet. Note that $SWNScore$ is used as a weight of a feature vector in the succeeding step of polarity classification, but it is just used for checking if the word has no polarity in the add-on lexicon creation step. That is, SentiWordNet is just used to check if the word is an objective word ($SWNScore(w_i) = 0$) or OOV word, then objective and OOV words will be sent to the revised polarity estimation step. The revised scores for these words are calculated by Equation (4.2).

$$SWNScore(W_i) = SWNScore_{POS}(w_i) - SWNScore_{NEG}(w_i) \tag{4.1}$$

$$Score(w_i) = \begin{cases} Score_{POS}(w_i), \\ \qquad \text{if } Score_{POS}(w_i) > Score_{NEG}(w_i). \\ (-1) \times \; Score_{NEG}(w_i), \\ \qquad \text{if } Score_{POS}(w_i) < Score_{NEG}(w_i). \end{cases} \tag{4.2}$$

where,

$$\begin{aligned}
Score_{POS}(w_i) &= \frac{P(positive|w_i)}{P(positive)} \\
Score_{NEG}(w_i) &= \frac{P(negative|w_i)}{P(negative)} \\
P(positive|w_i) &= \frac{No.\ of\ w_i\ in\ positive\ tweets}{No.\ of\ w_i\ in\ dataset} \\
P(negative|w_i) &= \frac{No.\ of\ w_i\ in\ negative\ tweets}{No.\ of\ w_i\ in\ dataset} \\
P(postitive) &= \frac{No.\ of\ positive\ tweets}{No.\ of\ all\ tweets} \\
P(negative) &= \frac{No.\ of\ negative\ tweets}{No.\ of\ all\ tweets}
\end{aligned}$$

Please note that $Score_{POS}(w_i)$ indicates the sentiment tendency of this word in positive orientation. $P(positive|w_i)$ is the probability that the tweet has positive sentiment when the word w_i appears in the tweet, while $P(positive)$ indicates the probability of the positive tweets in the corpus. $Score_{NEG}(w_i)$, $P(negative|w_i)$ and $P(negative)$ are also defined in the same way for estimating negative orientation of the word.

In the second step, since scores in SentiWordNet are in the range of -1 to 1, we have to convert the revised word scores into the same interval. In this case, we use a Bipolar sigmoid function [18] because it is continuous and returns a value from -1 to 1. The conversion formula is shown in Equation (4.3).

$$Score(w_i)^{'} = sigmoid(Score(w_i)) \tag{4.3}$$

where, $sigmoid(x) = \frac{2}{(1+e^{-x})} - 1$

The revised polarity score may be unreliable if the frequency of the word is too low, or the difference between positive and negative tendency is not great enough. Therefore, two thresholds are introduced. Threshold 1 (T_1) is the minimum number of words in the dataset and threshold 2 (T_2) is the minimum difference between positive and negative word orientation scores ($Score_{POS}(w_i)$ and $Score_{NEG}(w_i)$). The objective and OOV words with their scores are added to the add-on lexicon only when Equation (4.4) is fulfilled. The way how to determine T_1 and T_2 will be explained in Subsection 4.3.2.

$$\begin{aligned} Frequency\ of\ w_i\ in\ dataset &\geq T_1 \\ |Score_{POS}(w_i) - Score_{NEG}(w_i)| &\geq T_2 \end{aligned} \tag{4.4}$$

4.2.3 Lexicon Score Incorporation and Feature Weighting Methods

In this subsection, the word scores from both SentiWordNet and the add-on lexicon will be incorporated into the SVM classification features as extra prior information in four different ways: sentiment weighting, sentiment augmentation, sentiment interpolation and sentiment interpolation plus. We start with the baseline uni-gram and POS features, followed by our proposed sentiment lexicon incorporation method. Note that we ignore word sense disambiguation problem although the sentiment score is associated not with a word but with a synset in SWN. When SWN is consulted to obtain a sentiment score for a polysemous word, the first word sense in SWN is always chosen because it is the most representative sense of each word.

Uni-gram and POS Features

Uni-gram and POS features are common and widely used in the domain of sentiment analysis. There are many feature weighting schemes for the uni-gram. In this work, we use the pair of uni-gram (word) and its POS as the feature with term presence weighting. As a result, the weight of (word,POS) is 1 if it is present, otherwise 0. It is regarded as baseline in our experiment.

Sentiment Weighting Features

In this method, the feature weights of uni-gram binary vectors will be simply replaced with the word sentiment scores (Equation (4.1) or (4.3)) from the lexicon. Note that the weight is set to 0 if the word does not appear in the tweet.

Sentiment Augmentation Features

In this method, words will be classified into 3 groups: positive, neutral and negative, based on their scores in the lexicon. Then, these sentiment group features are augmented to the original uni-gram vector. There are three additional features that are the percentage of positive, neutral and negative words in a tweet, where the sum of the weights of these three features would be equal to one.

Sentiment Interpolation Features

In this method, we proposed a new incorporation method where the word score from the lexicon will be interpolated into the original uni-gram feature weight. The weight of the new interpolated vector is shown in Equation (4.5). Note that uni-gram score is always 1 in our model.

$$Weight = \alpha \ \textit{Uni-gram score} + (1 - \alpha) \ \textit{Lexicon score} \tag{4.5}$$

The parameter α ($0 \leq \alpha \leq 1$) is used for controlling the influence between the uni-gram model and the sentiment lexicon model. When α is equal to 1, the weight is the fully uni-gram model, and when α is 0, the weight is the fully sentiment weighting model.

Sentiment Interpolation Plus Features

In this method, we combine sentiment interpolation and sentiment augmentation together. Therefore, three additional augmentation features will be added to the sentiment interpolation vector as the extra features.

The summary of all features and weight values is shown in Table 4.2. Please note that the weight of the feature is always 0 if it does not appear in the tweets.

4.3 Evaluation

In this section, we present the results of two experiments. The first experiment was conducted with Positive-Neutral-Negative classification over full datasets (3-way classification). In the second experiment, we discarded neutral tweets and conducted the experiment with Positive-Negative classification over datasets of only positive and negative tweets. In addition, we used LIBLINEAR[10] [16] with default setting for training the SVM classifier.

[10] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

Table 4.2: Summary of feature vector construction methods

Methods	Feature weight value	Additional features
Uni-gram + POS	1	No
Sentiment Weighting	Lexicon score (Equation (4.1) or (4.3))	No
Sentiment Augmentation	1	percentage of positive, neutral and negative word in a tweet
Sentiment Interpolation	Equation (4.5)	No
Sentiment Interpolation Plus	Equation (4.5)	percentage of positive, neutral and negative word in a tweet

4.3.1 Data set

Sanders Dataset

The Sanders corpus[11] consists of 5,512 tweets on four different topics (Apple, Google, Microsoft, and Twitter). Each tweet was manually labeled as positive, negative, neutral or irrelevant. After removing irrelevant and duplicate tweets, 2,661 tweets are remained. Then, the dataset was randomly divided into two subsets. The first subset was used for the add-on lexicon creation part and training part, while the second was used for the testing (evaluation) part. Detailed information of this corpus is shown in Table 4.3. We used the Sanders dataset as a representative of small and domain-specific corpus.

SemEval 2013 Dataset

The SemEval 2013 corpus [47] consists of about 15,000 tweets that were created for the task 2 (Sentiment Analysis in Twitter) in the International Workshop on Semantic Evaluation (SemEval) 2013. Each tweet was manually labeled as positive, negative or neutral by Amazon Mechanical Turk workers. This dataset consists of a variety of topics.

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

Table 4.3: Sanders corpus

Subset	Used for	# Pos	# Neu	# Neg	# Total
1	Add-on lexicon creation, Training	319	1,319	345	1,983
2	Testing	109	455	114	678

Among the full dataset, only 10,534 tweets could be downloaded, because some of them were protected or deleted. This dataset was also randomly divided into three subsets. Detailed information on this corpus is shown in Table 4.4. Note that the development set was used for optimization of the parameter tuning. We used the SemEval 2013 dataset as a representative of a large and general corpus.

Table 4.4: SemEval 2013 corpus

Subset	Used for	# Pos	# Neu	# Neg	# Total
0	Development (parameter tuning)	1,297	1,401	475	3,173
1	Add-on lexicon creation, Training	2,272	3,083	884	6,239
2	Testing	372	441	187	1,000

In addition, the percentages of objective words and OOV words after data preprocessing in both corpora are shown in Table 4.5. We can find that the objective and OOV words form a majority in the corpora.

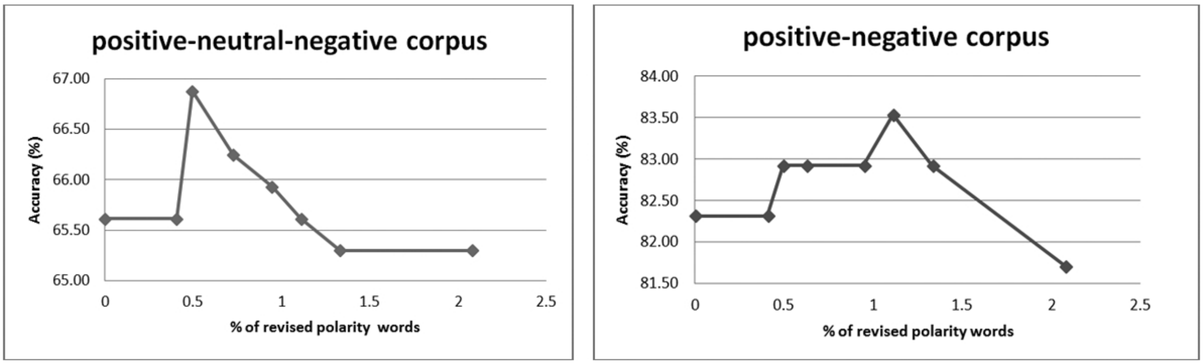
4.3.2 Parameter Optimization

As described in Subsection 4.2.2, in the add-on lexicon creation process, two thresholds can play an important role to control the number of revised polarity words. The objective and OOV words should not be revised if their estimated scores are not reliable enough. First, the threshold T_2 was set to 0.2 based on empirical observations. To investigate an

Table 4.5: Percentages of objective and OOV words in the two corpora

Corpus	**Objective words**	**OOV words**
Sanders	26.61%	57.73%
SemEval 2013	24.01%	66.55%

optimal value for the threshold T_1, we conducted a sensitivity test on the SemEval 2013 development dataset (subset 0 in Table 4.4).



(a) positive-neutral-negative classification

(b) positive-negative classification

Figure 4.2: The classification accuracy vs. number of revised polarity words on the development dataset

Figures 4.2 (a) and (b) show the accuracy of our method for various values of T_1 using interpolation plus weighting method in a 3-way and a positive-negative classification, respectively. In these graphs, the horizontal axis indicates the ratio of the number of words in the add-on lexicon to that of the corpus. The results show that, in 3-way classification, the classifier achieved better performance when the numbers of revised polarity words were smaller than the case of positive-negative classification. The accuracy reached its peak with the percentage of revised polarity words set around 0.5% (in 3-way classification) and 1.2% (in positive-negative classification). We did not investigate the optimum for the threshold T_1 in the Sanders corpus due to the insufficient number of tweets, but set T_1 so that the percentage of the number of the add-on lexicon is the same as in the optimized value in the SemEval 2013 dataset. Based on this observation, two thresholds were set as shown in Table 4.6.

Table 4.6: Optimized parameters in two corpora

Corpus	**Task**	T_1	T_2	***1**	***2**	***3**
Sanders	3-way	45	0.20	5,145	24	0.46%
	pos-neg	25	0.20	5,145	60	1.17%
SemEval 2013	3-way	60	0.20	15,366	78	0.50%
	pos-neg	35	0.20	15,366	173	1.12%

*1=number of types of the words in the corpus,

*2=number of the words added to the add-on lexicon,

*3=proportion of the words added to the add-on lexicon

Inspection of Parameter Optimization of T_1

In order to investigate the effectiveness of our parameter optimization method, we conducted the sensitivity test of T_1 over the test set of both SemEval 2013 and Sander corpora. Figures 4.3 a) and b) show the change of accuracy of our proposed sentiment interpolation plus method on two dataset in the 3-way and positive-negative classification, respectively. The red dot indicates the optimized value of T_1. For the SemEval 2013 dataset, the chosen point was near to the true peak on the test data. In addition, the optimized value on Sander data was also close to the peak in the positive-negative classification and almost same in the 3-way classification. These results showed that our method for the parameter optimization of T_1 was appropriate.

4.3.3 Results

Tables 4.7 and 4.8 show the results of the 3-way and positive-negative classification, respectively. They reveal the average of precision, recall and F1-measure over positive and negative classes as well as accuracy (Acc) for both Sanders and SemEval 2013 datasets. Five methods (including the baseline) described in Subsection 4.2.3 with and without the add-on lexicon are compared. In the experiment, the coefficient α in Equation (4.5) was initially set to 0.5 for maintaining the balance of uni-gram and lexicon score. The sensitivity of α will be investigated in Subsection 4.3.6. The results show that our pro-

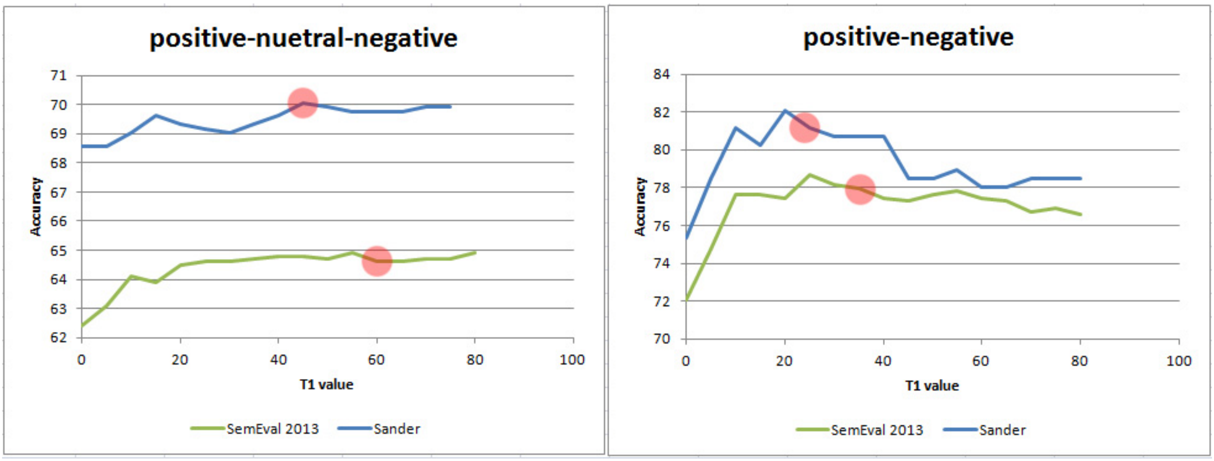


Figure 4.3: Inspection of parameter optimization of T_1 over the test data

Table 4.7: Results of 3-way classification task

Methods		**Sanders**				**SemEval 2013**			
Feature	**Lexicon**	**Precision**	**Recall**	**F1**	**Acc**	**Precision**	**Recall**	**F1**	**Acc**
Uni-gram + POS	No	0.454	0.444	0.446	0.667	0.575	0.482	0.518	0.617
Sentiment Weighting	SWN	0.306	0.392	0.306	0.423	0.485	0.478	0.464	0.531
	+Addon	0.323	0.315	0.300	0.541	0.554	0.425	0.472	0.606
Sentiment Augmentation	SWN	0.496	0.452	0.471	0.690	0.611	0.487	0.536	0.628
	+Addon	0.485	0.452	0.466	0.684	0.620	**0.491**	0.542	0.635
Sentiment Interpolation	SWN	0.451	0.407	0.427	0.671	0.588	0.471	0.514	0.621
	+Addon	0.467	0.425	0.443	0.676	0.595	0.476	0.519	0.622
Sentiment Interpolation Plus	SWN	0.511	**0.439**	**0.471**	0.702	0.646	0.484	0.547	0.644
	+Addon	**0.522**	0.430	0.469	**0.705**	**0.650**	0.487	**0.550**	**0.646**

Table 4.8: Results of positive-negative classification task

Methods		**Sanders**				**SemEval 2013**			
Feature	**Lexicon**	**Precision**	**Recall**	**F1**	**Acc**	**Precision**	**Recall**	**F1**	**Acc**
Uni-gram + POS	No	0.767	0.764	0.762	0.762	0.699	0.688	0.692	0.733
Sentiment Weighting	SWN	0.741	0.734	0.733	0.735	0.642	0.642	0.642	0.682
	+Addon	0.723	0.722	0.722	0.722	0.697	0.661	0.670	0.730
Sentiment Augmentation	SWN	0.776	0.773	0.771	0.771	0.719	0.700	0.707	0.750
	+Addon	0.765	0.763	0.762	0.762	0.725	0.712	0.717	0.755
Sentiment Interpolation	SWN	0.772	0.772	0.771	0.771	0.712	0.695	0.701	0.744
	+Addon	0.800	0.799	0.798	0.798	0.740	0.715	0.724	0.766
Sentiment Interpolation Plus	SWN	0.785	0.785	0.785	0.785	0.740	0.715	0.724	0.766
	+Addon	**0.813**	**0.812**	**0.812**	**0.812**	**0.759**	**0.728**	**0.739**	**0.780**

Table 4.9: Statistical test of the difference between Sentiment Interpolation Plus and baseline (uni-gram + POS)

	The two-tailed, P-Value	
Task	**Sander**	**Semeval 2013**
3-Way	0.005	0.0041
Positive-Negative	0.0455	0.0012

posed Sentiment Interpolation Plus and the use of the add-on lexicon achieved the highest accuracy in both 3-way and positive-negative classification over both Sanders and SemEval2013 corpora. Table 4.9 shows P-values of McNemar's test to evaluate the significance of the differences between our proposed method (Sentiment Interpolation Plus with add-on lexicon) and the baseline (uni-gram with POS). The result indicated that our method significantly outperformed the baseline at 99% confident level on 3-way classification task and 95-99% confident level on positive-negative classification task over both corpora. In addition, Table 4.10 shows the examples of tweets correctly identified only by the proposed Sentiment Interpolation Plus. It shows the source corpus, the gold label, the output of Uni-gram model and the output of Sentiment Interpolation Plus. We found that the revised polarity terms in the add-on lexicon, represented in bold, incorporated the correct sentiment of the words to the classifier. On the other hand, these words were not contained in SWN and the baseline (Uni-gram) could not predict the correct sentiment results.

4.3.4 Effect of the Add-on Lexicon

This section investigates the performance of the add-on lexicon. Comparing the results of the models with and without add-on lexicon in Table 4.7 and 4.8, it is found that the add-on lexicon can contribute to gain the performance of the sentiment analysis in most cases. To clarify the contribution of the add-on lexicon at a glance, the average accuracy of the models with four different feature weighting methods with only original SWN or both SWN and the add-on lexicon is shown in Figure 4.4. Here the average accuracy means the average of both 3-way and positive-negative classification tasks and

Table 4.10: Examples of tweets correctly identified by the proposed method

	Corpus	Gold	Uni-gram	Sentiment Interporaltion Plus	Tweet
1	SemEval	−	+	−	#pause I bet the clippers are gonna get in the Lakers **ass** today
2	SemEval	−	±	−	But tonight when I went to see Madonna at the Scottrade when I walked in I started **crying** because I thought about last Saturday.
3	SemEval	+	±	+	Colts game tonight! **Yay**!
4	SemEval	+	±	+	Pacers fans are going to have **fun** on Saturday...
5	Sander	−	+	−	@CBM: Lies @apple. the **battery** on this new iPhone4S is definitely not any better.
6	Sander	−	±	−	Having major **battery** drain issue since **updating** iPhone 4 to **iOS** 5. Anyone else? @AppStore @iPhone @apple
7	Sander	+	±	+	#google #galaxynexus #icecream **great**
8	Sander	+	−	+	New calendar app with pinch to zoom capabilities. Far superior to the current one! #Google **#Android** **#ICS**

(+=positve, −=negative, ±=neutral)

both datasets. It indicates that the add-on lexicon significantly improved the accuracy in the sentiment weighting and slightly improved the accuracy in the sentiment interpolation and sentiment interpolation plus. In the case of sentiment augmentation, the accuracy was almost the same. In addition, the combination of sentiment interpolation plus and the use of the add-on lexicon achieved the highest accuracy.

When the add-on lexicon was applied, the performance was improved more in positive-negative classification than in positive-neutral-negative (3-way) classification. Table 4.11 shows the average of both datasets of accuracy improvement in 3-way and positive-negative classification with and without the add-on lexicon when using two sentiment interpolation weighting methods. The result shows that when the add-on lexicon was applied, the accuracy was increased about 2% compared to applying only SWN in positive-negative classification, while only 0.25% in 3-way classification. Therefore the add-on lexicon is more suitable for positive-negative sentiment classification than positive-neutral-negative sentiment classification. The reason may be that in the case of 3-way classification, some neutral tweets were misclassified as subjective tweets when objective or OOV words were revised to subjective words.

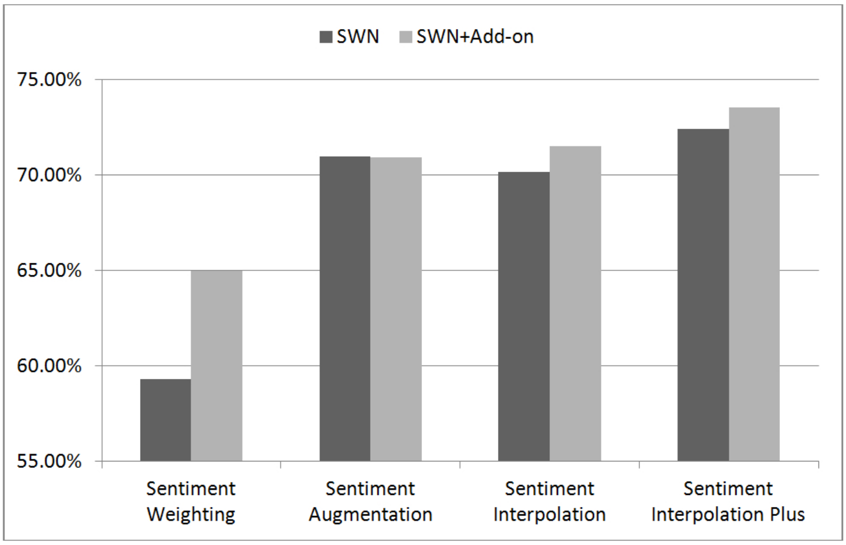


Figure 4.4: Contribution of the add-on lexicon

Table 4.12 compares the performance of the add-on lexicon over the Sanders and SemEval 2013 corpus when using sentiment interpolation plus weighting method. The average accuracy of two classification tasks are shown in this table. It seems that the add-on lexicon performed better over the domain specific corpus (Sanders) than the general corpus (SemEval 2013). Using the add-on lexicon, the average accuracy was improved by 1.49% on the Sanders corpus and 0.82% on the SemEval 2013 corpus.

Table 4.11: Comparison of the contribution of the add-on lexicon in two classification tasks

Classification	**Sentiment Interpolation**	**Sentiment Interpolation Plus**
3-Way	+0.27%	+0.25%
Positive-Negative	+2.42%	+2.06%

Table 4.12: Comparison of the contribution of the add-on lexicon in two corpora

Corpus	**SWN**	**+Add-on**	**Improvement**
Sanders	74.34%	75.83%	1.49%
SemEval 2013	70.48%	71.30%	0.82%

Table 4.13 and Table 4.14 show examples of the revised positive and negative words with their POSs and scores obtained from the Sanders and SemEval 2013 corpora, respectively. It can be observed that the revised polarity words in the Sanders corpus are more domain-specific than those in the SemEval 2013 corpus since the Sanders corpus is a collection of tweets associated with only four keywords: Apple, Android, Microsoft and Twitter. Most of words in the add-on lexicon seem reasonable. For example, in Sander corpus, the words 'battery', 'ios' and 'update' are revised to negative words since there are a lot of complaint about the battery of iPhone and ios update at that time, while '#ics' and '#android' are revised to positive words because 'ics' refer to the new release android os (Ice Cream Sandwich) which received a lot of compliment from their users. Note that complaint against android OS often implies positive opinion toward its competitor, i.e. Apple company.

Table 4.13: Examples of revised positive / negative words in the Sanders corpus.

Positive word	**Revised score**	**Negative word**	**Revised score**
#ics#OTHER	0.9223	battery#N	-0.9526
look#V	0.9211	customer#N	-0.9253
power#N	0.8926	update#N	-0.9109
:)#OTHER	0.8851	dear#OTHER	-0.9074
#android#N	0.8698	lot#N	-0.8931
help#V	0.8698	send#V	-0.8931
user#N	0.8664	#ios#OTHER	-0.8776
great#A	0.8252	service#N	-0.8049
game#N	0.8041	wait#V	-0.7434
thank#V	0.7994	ass#N	-0.7086

Table 4.14: Examples of revised positive / negative words in the SemEval 2013 corpus.

Positive word	**Revised score**	**Negative word**	**Revised score**
thank#V	0.8637	:(#OTHER	-0.9920
fun#A	0.8628	fuck#N	-0.9900
luck#N	0.8560	cancel#V	-0.9872
great#A	0.8442	damn#OTHER	-0.9864
:D#OTHER	0.8421	niggas#N	-0.9690
yay#OTHER	0.8341	die#V	-0.9554
pakistan#OTHER	0.8265	dont#V	-0.9329
:)#OTHER	0.8170	ass#N	-0.9272
yeah#OTHER	0.7999	cry#V	-0.9168
celebrate#V	0.7928	russia#OTHER	-0.9039

4.3.5 Comparison of Feature Weigthing Methods

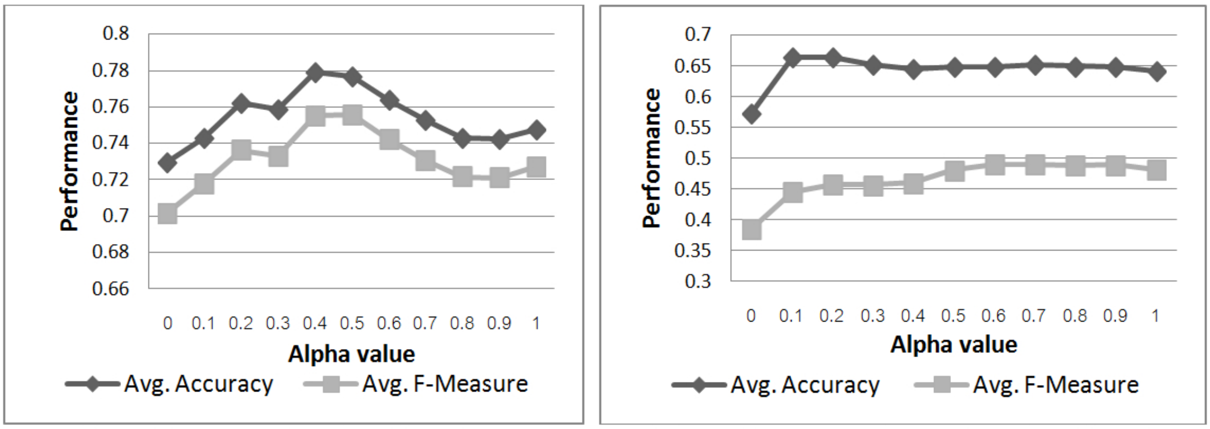
Table 4.15 shows the comparison among four feature weighting methods and the baseline uni-gram. It reveals the average accuracy of the methods on both Sanders and SemEval corpora in both 3-way classification and positive-negative classification tasks, where both SentiWordNet and the add-on lexicon are used as the sentiment lexicon. First, the accuracy of the sentiment weighting method (the score in the lexicon is used as the weight) was 4.51% worse than the uni-gram method. It may be because, unlike uni-gram weighting, the weights of objective and OOV words were set to 0 even when they appeared in the tweets. It means that the classifier loses the information about these words. Sentiment augmentation, where three lexicon scores were added to original uni-gram as extra features, improved the accuracy by 1.43%. Sentiment interpolation, where lexicon scores were interpolated into uni-gram vector weights, further improved the accuracy 2.05% compared to baseline. Finally, the combination of sentiment interpolation and sentiment augmentation, called sentiment interpolation plus, achieved the highest accuracy among all methods with average accuracy improvement 4.08% compared to baseline uni-gram.

Table 4.15: Comparison among feature weighting methods

Methods	**Avg. Acc**	**Improvement**
Uni-gram + POS	69.49%	-
Sentiment Weighting	64.98%	-4.51%
Sentiment Augmentation	70.92%	1.43%
Sentiment Interpolation	**71.53%**	**2.05%**
Sentiment Interpolation Plus	**73.57%**	**4.08%**

4.3.6 The Sensitivity of α Parameter

In the sentiment interpolation method, the α parameter in Equation (4.5) plays an important role for controlling the influence of uni-gram and sentiment lexicon scores. To analyze the effect of the α parameter, different values of the α parameter were applied. Note that when α is equal to 1, the vector weight becomes a fully uni-gram model (only term presence are used as feature weight), and when α is equal to 0, the vector weight value becomes a fully sentiment weighting model (only lexicon scores are used as feature weight).



(a) positive-negative classification

(b) positive-neutral-negative classification

Figure 4.5: Effect of the α parameter in the sentiment interpolation plus method

Figures 4.5 a) and b) show the change of the average accuracy and F1-measure of the sentiment interpolation plus method on two datasets in the 3-way and positive-negative classification, respectively. In the positive-negative classification, the result clearly shows that the integration of uni-gram and lexicon score outperformed either uni-gram or sentiment weighting. The sentiment interpolation plus method performed well with large rage of α values (0.2 to 0.7). On the other hand, in the 3-way classification, it seems that the sentiment interpolation plus method only slightly increased the performance compared to uni-gram or sentiment weighting in most of the α values. As discussed earlier, the sentiment interpolation plus method was more suitable for the positive-negative classification than the 3-way classification task.

4.3.7 Comparison with the Participating System on SemEval 2013 Task 2

In this subsection, we compare our proposed method with other tweet-level sentiment analysis methods. SemEval 2013 Task 2 includes two subtasks: task 2A (contextual polarity disambiguation) and task 2B (message polarity classification). Since the task of the experiment in this chapter is same as the task 2B, our system should be compared with the participating systems of this task. However, the direct comparison is impossible. Because the development and training data of SemEval 2013 task 2B is freely available, but the test data is only opened for the participants.

The developers of two participating systems, NRC-Canada [44] and ECNUCS [83], reported the performance of their systems by 10 fold cross validation on the training data. We also conducted 10 fold cross validation on the training data of SemEval 2013 dataset and compared our method with them. Please note that the number of tweets used in their evaluation (8,258) and our evaluation (7,239) is different, because some of tweets were protected or deleted. Table 4.16 reveals the performance of NRC-Canada, ECNUCS trained by Maximum Entropy, ECNUCS trained by SVM and our proposed Sentiment Interpolation Plus. The second to fourth columns show F1-measure of positive, neutral and negative tweets; the fifth column shows the average of F1-measure of positive and negative tweets; the last column shows the accuracy. Our method outperformed ECNUCS but worse than NRC-Canada.

In the SemEval 2013 task 2B, NRC-Canada got the top-rank and ECNUCS was ranked at 18th among all 51 participating system[12]. From the results of the comparison in Table 4.15, our Sentiment Interpolation Plus works better than the middle ranked system but does not perform very well comparing to the best system. However, we only used simple features for machine learning. The combination of the other useful features proposed by the other methods can further improve the performance of our method.

Table 4.16: Results of NRC-Canada, ECNUCS and Sentiment Interpolation Plus

Method	**F-pos**	**F-nue**	**F-neg**	**aveage F** **(pos and neg)**	**Acc** **(%)**
NRC-Canada	-	-	-	**0.672**	-
ECNUCS (MaxEnt)	0.6488	0.7083	**0.4587**	0.5538	64.89
ECNUCS (SVM)	0.6593	0.7288	0.4481	0.5537	66.41
Sentiment Interpolation Plus	**0.6826**	**0.7571**	0.4156	0.5642	**69.45**

[12] https://www.cs.york.ac.uk/semeval-2013/index.php%3Fid=evaluation-results.html

4.3.8 Error Analysis

Table 4.17: Example of misclassified tweet

	Gold	Uni-gram	Sentiment Interporaltion Plus	Tweet
1	±	±	+	Gonna go to zumba with y mom Mondays Wednesdays nd **Fridays** then zumba at school on **Fridays** nd **Saturdays** kettle bell work outs!
2	±	±	+	#HalloweenSong now playing X Files, Phantom of the Opera, Halloween, **Friday** the 13th, Psycho, Devil's Rejects - what are you listening to?"
3	+	+	−	My iPhone 4S **battery** lasted longer than a day. That hasn't happened since my edge iPhone. Nice job!
4	+	+	−	#**iOS**5 **update** submitted to @apple! Thanks for all the support!
5	−	−	+	I kicked off Type 1 Diabetes Awareness Day with a high blood sugar **thanks** to too many chocolates last night. **Well**, there's always tomorrow

(+=positve, −=negative, ±=neutral)

To better understand the limitations of our system, we manually inspected classification errors. Table 4.17 shows examples of misclassified output of our proposed method. In this table, the gold label, the output of Uni-gram model and the output of Sentiment Interpolation Plus are compared. The common errors can be grouped into 3 cases. (1) Neutral tweets are misclassified as subjective tweets because of the revised polarity words as shown in tweet #1 and #2. The word 'Friday' and 'Saturday' are revised to positive word in the add-on lexicon because most of the time they refer to holiday and show some positive sentiment tendency. However, these words seem neutral in the tweet #1 and #2. (2) Some words have strong sentiment tendency at a certain period of time, but not always. Tweet #3 and #4 show the example of misclassified tweet because of word ‘ios’, ‘update’ and ‘battery’. These 3 words are revised to extremely negative because a lot of complaints about the battery of iphone and iOS update are posted. Once the problems on the battery or iOS update are fixed, these words may not indicate the negative sentiment. (3) In sarcasm tweet, the sentiment of the tweet is usually opposite to the real meaning that a user wants to express as shown in tweet #5. The noun ‘thank’ and the interjection ‘well’ are the positive words in the add-on lexicon. It leads the error that this negative tweet is wrongly classified as positive by our method. On the other hand, the baseline (Uni-gram) can classify the tweet correctly.

4.4 Summary

In this chapter, we have shown an alternative hybrid method that incorporated sentiment lexicon information into the machine learning method to improve the performance of Twitter sentiment classification. There are two main contributions of our proposed method. First, we estimated the implicit polarity of objective and OOV words and used these words as additional information for the public sentiment lexicon. We described how we revised the polarity of objective and OOV words based on the assumption that the polarities of words are coincident with the polarity of their associated sentences, which seem reasonable due to the short length of tweets. Second, we proposed an alternative way to incorporate sentiment lexicon knowledge into a machine learning algorithm. We proposed the sentiment interpolation weighting method that interpolated lexicon score into uni-gram score in the feature vectors of SVM.

Our results indicate that the data-oriented add-on lexicon improved the classification accuracy on average compared to the system using only the original public lexicon. The proposed sentiment interpolation weighting method performed well and the combination of sentiment interpolation and sentiment augmentation, called sentiment interpolation plus, with SentiWordNet and the add-on lexicon achieved the best performance and significantly improved the classification accuracy compared to the uni-gram model. The experiments show that the add-on lexicon performed better over the domain-specific corpus than the general corpus. Although our tweet-level classifier performed less than the state-of-the-art method, the main contribution of this chapter is to propose a method to create the add-on lexicon and empirically evaluate its effectiveness. The method to construct the add-on lexicon is employed in our target-dependent sentiment analysis in Chapter 5.

Our results indicate that the proposed approach was more appropriate for positive-negative classification than positive-neutral-negative (3-way) classification. Therefore, we plan to apply the subjective classification as our future work in order to filter the neutral tweets before the polarity classification. Since negation words such as “not” and “less” are simply treated as uni-gram features in this work, another interesting issue is investigation on how special treatments of negation affect the polarity classification. Furthermore, we plan to find a method to reestimate the word polarity from unlabeled data or noisy labeled data instead of labeled data that is time consuming to create.

Chapter 5

Target-dependent Sentiment Analysis

This chapter presents an approach to improve the performance of the sentiment analysis for the specific target by incorporating several target specific knowledge. We propose a method that incorporates the on-target sentiment features and user sentiment features into the classifier trained automatically from the data created for the specific target, called **Ta**rget **S**pecific **K**nowledge **Sen**timent Classification (TASK-SEN). An add-on lexicon, extended target list, and competitor list are also constructed as knowledge sources for the sentiment. The results of our experiment show that our method is effective and improves on the performance of sentiment classification compared to the baselines.

5.1 Background and Motivation

Many researchers have adopted both machine learning and lexicon based approaches for sentiment analysis on Twitter. Most of these approaches aim at identifying the sentiment of the tweet, but not that addressed to a specific target in the tweet. In other words, they classify the sentiment of the tweet, not the target. In Twitter, however, it is common that a user expresses several sentiments in one tweet or the sentiment about not the target but other things. For example, the sentiment of the tweet "I hate when my mom annoying me with questions about her iphone" is clearly negative at the tweet-level, but neutral to the target "iPhone". These target-independent approaches may be insufficient for the practical use of sentiment analysis, since it is often required to know the sentiments towards a specific target, such as a product, brand, or person. The users may want to know the opinion of other people about the products they are interested in, before making a purchase decision. Companies also want to know the overall opinion of their products of their potential users.

The goal of this research is to develop a method of classifying the sentiments (positive, negative or neutral) of a given target in the tweets. Our method relies on supervised machine learning. However, we try to develop our system without any human intervention, such as annotation of the labeled data. This enables us to apply our method to the sentiment analysis of various targets. Several techniques will be proposed to improve on the performance of target dependent sentiment classification. First, not general but target dependent training data is constructed for learning the sentiment classifier. It is automatically created by a lexicon-based method and several heuristics from unlabeled tweets. Second, a target-specific add-on lexicon is automatically constructed. A public sentiment lexicon is insufficient for target specific sentiment analysis, since the words used to express an opinion of the target are often not compiled in it. In this research, an additional sentiment lexicon is built by automatically identifying the polarity of the objective and out-of-vocabulary words. Third, a user sentiment feature is considered. The theory of Sentiment Consistency [1] indicates that the sentiment of two messages posted by the same user are more likely to be consistent than those of two randomly selected messages. Therefore, it would be better to take into consideration the other tweets of the same user that express an opinion about a given target. This user specific information can imply how likely are positive and negative opinions of the user to be expressed about the target. Finally, an extended target list and competitor list are introduced into the model. The former is the list of synonyms of the target. It is used to identify the target when expressed by different words or phrases. The latter is a list of the competitors of a given target (e.g. a product). People sometimes give their comments not only about the target itself but also its competitors, especially when they are comparing competing products. The competitor list can contribute to distinguishing whether the tweet is expressing an opinion about the target or its competitor.

5.2 Proposed Method

This section presents the proposed approach. An overview of the system framework is shown in Figure 5.1. For a given target, tweets containing the target word are retrieved by Twitter API. They are classified as positive, negative or neutral in several steps to create the target-specific training data. Then, uni-gram, on-target and user sentiment features are extracted. Finally, the SVM is trained to classify the sentiment towards the target in the tweet.

5.2.1 Data Preprocessing

Preprocessing in TASK-SEN is same as the process described in Subsection 4.2.1. We normalize all the tweets as follows: 1) tokenization and lemmatizing, 2) character repetition replacement, 3) stop word and URL removal and 4) part-of-speech (POS) tagging.

5.2.2 Creating a Target-specific Add-on Lexicon

A target-specific add-on lexicon is constructed by a similar method explained in Subsection 4.2.2. However, instead of using labeled corpus as a data source, the sentiment tendency of objective and OOV words are estimated from the corpus of unlabeled historical tweets.

After the preprocessing, the objective and OOV words of SentiWordNet are extracted from the retrieved tweets through TwitterAPI[1]. In this step, only adjectives, interjections and hashtags are extracted, because they are the most informative for sentiment classification. For each word, the relevant tweets are retrieved by a query, that is combination of the objective or OOV word and the given target. The tweets that contain URLs are discarded because they commonly refer to some external resources, and re-tweet messages are also ignored because they are copies of the original tweets. Next, the polarity of each tweet is identified by the SentiStrength tool [74]. SentiStrength is a state-of-the-art lexicon based method for classifying the sentiment of short social texts, and it has been applied in much related research[46, 60]. Finally, the sentiment score of the objective and OOV words are calculated using Equation (5.1). Note that $P(positive|w_i)$ and $P(positive)$ are the ratio of the number of the tweets positively classified by SentiStrength to the number

[1] https://dev.twitter.com/rest/public

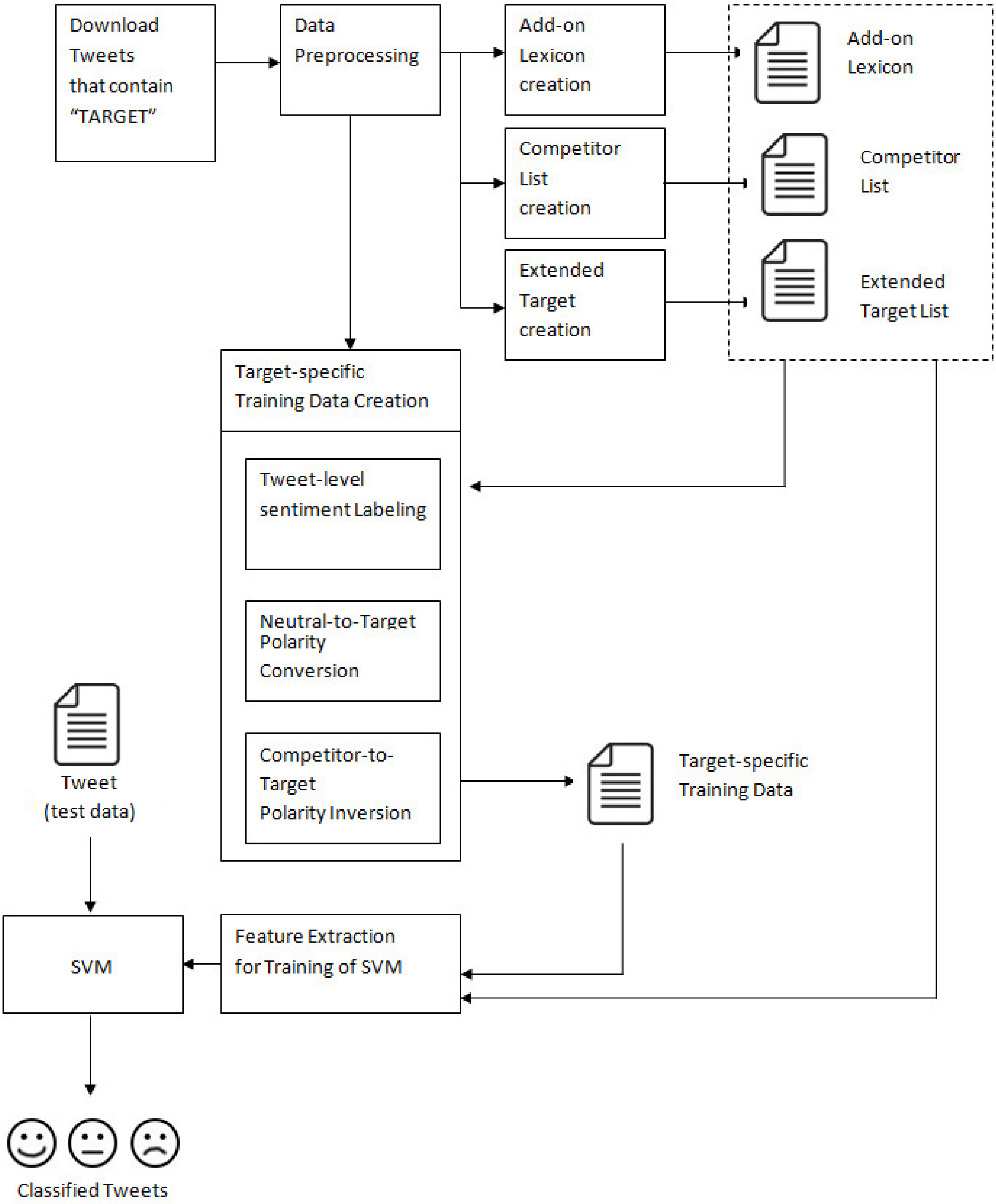

Download Tweets that contain "TARGET"
Data Preprocessing
Add-on Lexicon creation
Competitor List creation
Extended Target creation
Add-on Lexicon
Competitor List
Extended Target List
Target-specific Training Data Creation
Tweet-level sentiment Labeling
Neutral-to-Target Polarity Conversion
Competitor-to-Target Polarity Inversion
Target-specific Training Data
Tweet (test data)
SVM
Feature Extraction for Training of SVM
Classified Tweets

Figure 5.1: TASK-SEN System Framework

of the tweets containing w_i and all tweets, respectively. $P(negative|w_i)$ and $P(negative)$ are defined in the same way.

$$Score(w_i) = \begin{cases} Score_{POS}(w_i), \\ \qquad \text{if } Score_{POS}(w_i) > Score_{NEG}(w_i). \\ (-1) \times \ Score_{NEG}(w_i), \\ \qquad \text{if } Score_{POS}(w_i) < Score_{NEG}(w_i). \end{cases} \tag{5.1}$$

where

$$Score_{POS}(w_i) = \frac{P(positive|w_i)}{P(positive)}$$
$$Score_{NEG}(w_i) = \frac{P(negative|w_i)}{P(negative)}$$

Next, since scores in SentiWordNet are in the range of -1 to 1, we have to revise our sentiment score in the same interval. A Bipolar sigmoid function defined in Equation (4.3) is used.

Similar to Subsection 4.2.2, the polarity score may be unreliable if the frequency of the word is too low, or the difference between the positive and the negative tendency is not large enough. Therefore, threshold T_1, which is the minimum number of words in the dataset and threshold T_2, which is the minimum difference between positive and negative word orientation scores, are also applied. For the experiment in Section 5.3, we set the threshold T_1 to 10 and T_2 to 0.4, based on empirical observations.

Note that the difference between this module and Subsection 4.2.2 is the resource to create the add-on lexicon, that is the former and the latter use unlabeled and labeled tweets respectively. Another important difference is that the target-specific add-on lexicon is designed for a certain target, while the add-on lexicon in Subsection 4.2.2 is designed for a general use.

5.2.3 Extended Target Creation

In Twitter, users might not express their opinion about a given target with the target keyword exactly. Sometimes, they comment about its features, concept, or things related to the target. Therefore, it would be better to create an extended target list, consisting of terms that can be used as a representative of the target. For example, let us consider the tweet "I hate all Apple products." It can be guessed that this user also hates the targets "iPhone", "iPad" and "iPod". So, the term "Apple" should be added to the extended target list of iPhone and so on. In [10, 25], the extended target list was created by measuring the Pointwise Mutual Information (PMI) between the candidate terms and the target on a corpus containing 20 million tweets. However, the performance of PMI is quite sensitive to the corpus size and it is very time-consuming to download a tweet corpus that contains enough data for the various candidate terms. Therefore, we propose a method to estimate the PMI using statistics obtained from Twitter API without a pre-downloaded tweet corpus.

First, the nouns and proper nouns are selected as the candidate terms of the extended targets. Next, we estimate the relatedness between the candidate term C and the target T by approximating the PMI as in Equation (5.2). The functions $n(T)$, $n(C)$ and $n(T, C)$ are, respectively, the number of tweets containing T, C, and both T and C, while $time(T)$, $time(C)$ and $time(T, C)$ are the time ranges in which these tweets were posted. These statistics can be immediately obtained by TwitterAPI. $n(all\_tweets)$ and $time(all\_tweet)$ are estimated from the Twitter statistic that there are around 6,000 tweets per second on average[2]. The extended target list is built from all candidate terms whose PMI is greater than a threshold. For the experiment in Section 5.3, we set the threshold to 0 based on empirical observations.

$$\begin{aligned} PMI(T,C) &= log\frac{p(T,C)}{p(T)p(C)} = log\frac{p(T|C)}{p(T)} \\ P(T|C) &= \frac{\frac{n(T,C)}{time(T,C)}}{\frac{n(C)}{time(C)}} \\ P(T) &= \frac{\frac{n(T)}{time(T)}}{\frac{n(all\_tweet)}{time(all\_tweet)}} \end{aligned} \tag{5.2}$$

[2] http://www.internetlivestats.com/twitter-statistics/

5.2.4 Competitor List Creation

In Twitter, users might comment not only on the target itself, but also express their sentiment by comparing the target with its competitors. For example, a tweet "I'm fucking pissed I broke my iPhone and have to use this shitty Android" is clearly negative to "Android" and seems neutral to "iPhone". If we know that Android is a competitor of iPhone, we can infer that this user is expressing their positive expression to the target "iPhone". Therefore, it would be effective for sentiment analysis to create a list of the terms for which the sentiment is opposite to the target, called the competitor list. To the best of our knowledge, no general method to obtain the competitor-to-target terms has been reported. This research presents a novel method to automatically create the list of terms that can be considered as competitors to the target by use of the word "VS" (versus) as the main keyword. It is usually used when people compare two things.

First, we build the queries "TARGET vs" and "vs TARGET" and enter them into the Search API of Twitter. Then, the retrieved set of tweets is cleaned by discarding the duplicate tweets and re-tweets, removing stop words and one character words. Next, we extract the two words connected immediately before or after the term "VS" as the candidate terms. More specifically, the terms are selected only when they are located on the opposite side of "VS" from the target term without ":". Next, we measure the relatedness between the target and the candidate term by the PMI formula as shown in Equation (5.2), and select as the competitors those terms where the PMI is greater than a threshold. For the experiment in Section 5.3, we set the threshold to 0 based on empirical observations. Finally, the terms in the extended target (described in Subsection 5.2.3) will be removed from the competitor list.

5.2.5 Target-specific Training Data Creation

As discussed above, the performance of machine learning is sensitive to the domain of the training data. A classifier usually does not perform well when it is trained from the training data of a different domain [3]. This research presents a novel method to create a target-specific training data set without manual annotation. We first use the state-of-the-art lexicon-based sentiment analysis tool that performs well at the tweet-level sentiment analysis. Then, we use heuristic rules to convert the sentiment to neutral if the sentiment score at the tweet level is very different from that at the target level. In other words, the sentiment of the tweets where the users express their opinion but not truly about the target will be converted to neutral. Finally, the sentiment labels of comparison tweets, where the users express their opinion of a competitor, will be inverted to the opposite orientation.

Tweet-level Sentiment Labeling

In this step, we create a set of sentiment-classified tweets at the tweet level. Several researchers have used emoticons, such as :) and :(, or hashtags, such as #fail, to create data labeled with sentiments [20, 28]. However, both emoticons and hashtags are sparse for preparing a large amount of training data for some target keywords. In our proposed method, the tweets related to the target are first retrieved and preprocessed as described in Subsection 5.2.1. The tweets containing URLs or re-tweets are discarded from the data, since they could express a sentiment not about the target but about the contents of the linked page or other tweet. Then, similar to the add-on lexicon creation in Subsection 5.2.2, we use SentiStrength to classify the tweets as positive, negative or neutral.

Neutral-to-target Polarity Conversion

The labels of the tweets in the corpus created by the previous step are the sentiments of the whole tweet, not the target. We can usually regard the sentiment of the tweet as coinciding with that of the target, but the positive or negative tweets sometimes indicate a sentiment about other things, not the target. In such cases, the sentiment should be revised to neutral. Algorithm 1 shows the method of this neutral-to-target polarity conversion. First, two scores, called $score_{tw}$ and $score_{tg}$, are calculated by looking up in the public sentiment lexicon SentiWordNet and in our add-on lexicon. $score_{tw}$ is the summation of the lexicon score of all words in the tweet, called the "tweet-level lexicon score". On the other hand, $score_{tg}$ is the summation of the lexicon score for the words that are probably related with the target or its extended target, called the "on-target lexicon score". $DWLS(x, y)$ is the "distance weighted lexicon score" between the two words. It is defined as the lexicon score of x weighted by the reciprocal of the distance between x and y, where the distance is the length of the path from x to y in the dependency tree. Note that $score_{tg}$ is estimated based on $DWLS(w_i, tg_i)$. Finally, the sentiment label of the tweets will be converted to neutral if the relative difference between $score_{tw}$ and $score_{tg}$ is greater than a threshold. For the experiment in Section 5.3, we set the threshold to 0.66, based on empirical observations.

Algorithm 1: Neutral-to-target Polarity Conversion

```
Input: Tweet corpus with label $TC_L$, threshold $T_1$, the SWN, the add-on lexicon,
       the extended target
Output: Tweet corpus with new label of tweet $TC_N$
while $not\ at\ end\ of\ the\ TC_L$ do
    read current tweet $tw_i$ ;
    if $tw_i$ contains more than one noun then
        for word $w_i \in tw_i$ do
            $score_{tw} = score_{tw} + lexicon\_score(w_i)$;
            for target $tg_i \in Extended\_Target\_List$ do
                $DWLS(w_i, tg_i) = \frac{lexicon\_socre(w_i)}{distance(w_i, tg_i)}$;
            end
            $score_{tg} = score_{tg} + max_{tg_i}(DWLS(w_i, tg_i))$;
        end
        if $\frac{|score_{tw} - score_{tg}|}{|score_{tw}|} > T_1$ then
            Set $label(tw_i) = Nuetral$ ;
        end
    end
end
```

Competitor-to-target Polarity Inversion

As discussed in Subsection 5.2.4, a user might express a sentiment by comparing the target with its competitors. If the sentiment label at the tweet level stands for the opinion about the competitors of the target, it should be inverted at the target level. Algorithm 2 shows the method of the competitor-to-target polarity inversion. First, we select only the tweets that contain the terms in the competitor list. Then, two scores, called $score_{tg}$ and $score_{cp}$, are calculated by looking up in SentiWordNet and our add-on lexicon. $score_{tg}$ is the "on-target lexicon score" in the neutral-to-target polarity conversion, and $score_{cp}$ is the summation of the lexicon score for the words that are probably related to the competitors, called the "on-competitor lexicon score". Both $score_{tg}$ and $score_{cp}$ are calculated by the lexical score weighted by the reciprocal of the distance in the dependency tree. The sentiment label will be inverted if the difference between $score_{tg}$ and $score_{cp}$ is greater than a threshold. Moreover, the polarity of the tweet should be inverted only when the main opinion is expressed about the competitor. More specifically, the tweet label will be inverted only when the sign of the original label (denoted by $sign(tw_i)$ in Algorithm 2) is the same as the sign of $score_{cp}$, and the sign of $score_{tg}$ and $score_{cp}$ is not the same. For the experiment in Section 5.3, we set the threshold to 0.2, based on empirical observations.

5.2.6 Feature Extraction

In this subsection, we will explain how to represent a tweet as a feature vector to train a classifier for target-dependent sentiment classification.

Uni-gram and POS Features

Uni-gram and POS features are common and widely used in the domain of sentiment analysis. Although there are many feature weighting schemes for uni-gram, binary weighting is used as the baseline method in this work. That is, the weights of a pair of a word and its POS is 1 if it is present in the tweet, otherwise 0.

On-target Sentiment Features

The polarity score of the sentiment lexicon is widely used as a feature in sentiment classification, too. In order to perform a sentiment analysis at the target level, *on-target_lexicon_score* of the sentiment words in both SentiWordNet and the add-on lexicon are defined by Equation (5.3). tg' is the closest target (or its extended target) to the sentiment word w_i, while cp' is the closest term in the competitor list. The score is weighted by the reciprocal of the distance between the sentiment word w_i and tg' if tg' is closer than cp', otherwise between w_i and cp' with sign inversion. The sentiment words in the tweet are then classified into two classes: positive and negative, based on their score.

Algorithm 2: Competitor-to-target Polarity Inversion

```
Input: Tweet corpus with label $TC_L$, threshold $T_2$, the SWN, the add-on lexicon,
       the extended target list, the competitor list
Output: Tweet corpus with new label of tweet $TC_N$
initialization;
while $not\ at\ end\ of\ the\ TC_L$ do
    read current tweet $tw_i$ ;
    if $tw_i\ contains\ competitors$ then
        for $word\ w_i \in tw_i$ do
            for $target\ tg_i \in Extended\_Target\_List$ do
                $DWLS(w_i, tg_i) = \frac{lexicon\_socre(w_i)}{distance(w_i, tg_i)}$;
            end
            $score_{tg} = score_{tg} + max_{tg_i}(DWLS(w_i, tg_i))$;
            for $competitor\ cp_i \in Competitor\_List$ do
                $DWLS(w_i, cp_i) = \frac{lexicon\_socre(w_i)}{distance(w_i, cp_i)}$;
            end
            $score_{cp} = score_{cp} + max_{cp_i}(DWLS(w_i, cp_i))$;
        end
        if $\left|score_{tg} - score_{cp}\right| > T_2$ and $sign(tw_i) =$
        $sign(score_{cp})$ and $sign(score_{tg}) \neq sign(score_{cp})$ then
            Set $label(tw_i) = oppsite(label(tw_i))$ ;
        end
    end
end
```

On-target sentiment features are two additional features for positive and negative classes whose weights are defined as the sum of the *on-target_lexcion_score* of the positively and negatively classified words.

$$\textit{On-target\_lexicon\_score}(w_i,\textit{target}) \\ = \begin{cases} DWLS(w_i, tg'), \text{if } DWLS(w_i, tg') \geq DWLS(w_i, cp'). \\ (-1) \times DWLS(w_i, cp'), \text{otherwise.} \end{cases} \tag{5.3}$$

user sentiment features

user sentiment features represent the latent opinion of the user about a given target. It is often difficult to understand the opinion of the user from one short tweet. In our method, other tweets of the user are taken into account to guess the user's latent opinion of the target. SentiStrength is used to classify the tweets which contain the target word and are posted by the same user, as either positive, negative, or neutral. The user sentiment features are three additional features for positive, negative and neutral classes whose weights are defined as the percentage of the positive, negative and neutral tweets of the user.

Sarcasm feature

One of the characteristics of Twitter is that a considerable number of the tweets is sarcasm. Sarcasm is a kind of irony that is used to mock or express contempt. Since the true meaning of sarcasm is usually opposite to its literal meaning, sarcastic tweets might be difficult to identify its polarity. Therefore, it requires special technique to deal with. We introduce sarcasm feature, one additional binary feature indicating whether a tweet is a sarcastic or not. That is, the weight of the sarcasm feature is 1 if the tweet is classified as sarcastic tweet, otherwise 0.

We use the sarcasm identification system proposed by [76] to classify sarcasm tweets. The system is based on supervised learning framework that focuses on several features: 1) concept expansion, 2) semantic sentiment analysis, 3) coherence identification and 4) N-grams. According to the definition of sarcasm, it often occurs in a contradictory form of communication or the use of words to express something opposite to the intended meaning

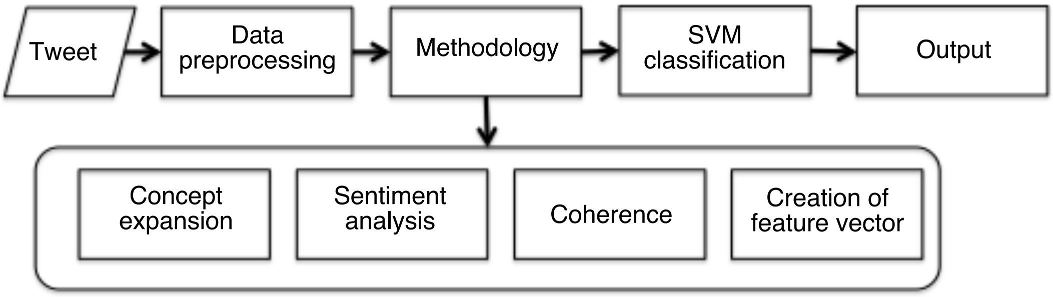


Figure 5.2: Overview of sarcasm recognition method [76]

[58]. The system attempts to use sentiment analysis to find contradiction in sentiment polarity between words in a tweet. Furthermore, a way to expand concepts of unknown sentiment words is also presented to compensate for insufficiency of a sentiment lexicon. The method uses a concept lexicon called "ConceptNet 2.0"[3] to expand the concepts for the words whose sentiment score is unknown. This allows the system to recognize sarcasm of the sentence in the concept level. Coherence in sentences in a tweet is considered to identify sentiment contradiction. They also propose a new method to identify coherence among multiple sentences based on unsupervised clustering in this system. After the process of feature extraction, Support Vector Machine (SVM) will be used to classify sarcastic tweet based on the proposed features as well as ordinary N-grams. The output from the classifier is based on an ensemble of two SVMs with two different feature sets. Figure 5.2 shows the overall procedures of sarcasm identification method. For more detail, see [76].

5.3 Experimental Setup

5.3.1 Dataset

Because people usually express their opinion about products, brands, companies and celebrities, we selected "iPhone", "Xbox", "Nike" (products/brands), "Google", "Verizon", "Sony" (companies) and "Obama", "Beyonce", "Messi" (person) as the targets for sentiment analysis. In order to create the training data, we downloaded, via the Twitter Search API, the collection of those tweets that contain the target keyword. After the

[3] http://alumni.media.mit.edu/˜hugo/conceptnet/

creation of the target-specific training dataset (as in Subsection 5.2.5), we balanced the number of positive, negative and neutral tweets so that the training data would consist of equal numbers of tweets for each class. Due to the limitation of Twitter API[4], we select one representative target for each domain (iPhone, Google and Obama) to evaluate the effectiveness of the user sentiment feature. For each user in the training data, 3,200 tweets posted by that user were downloaded. Then, only the tweets containing the target (on-target tweets) were used to obtain the user sentiment feature. For the test data, another 300 tweets (100 for each class) including the target keyword were retrieved. They were manually annotated with the sentiment for the target. To investigate the reliability of the manual annotation of the test data, the second annotator also judged the polarity of the tweets of three targets, iPhone, Google and Obama. Then the inter-annotator agreement was measured. The agreement ratio was 88.44%, and Kappa coefficient was 0.842. There results indicated that the quality of the gold labels was good enough. Statistics of the dataset is shown in Table 5.1.

Table 5.1: Statistics of the dataset

Target	**Tweets in training set**	**Tweets in test set**	**Users**	**On-target tweets**
iPhone	10,500	300	10,500	64,260
Xbox	15,000	300	-	-
Nike	15,000	300	-	-
Google	12,000	300	12,000	69,240
Verizon	15,000	300	-	-
Sony	9,000	300	-	-
Obama	10,500	300	10,500	258,510
Beyonce	13,500	300	-	-
Messi	13,500	300	-	-
Total	**114,000**	**2,700**	**33,000**	**392,010**

[4] The limit for getting timeline tweets of the users is set to 180 requests per 15 minutes.

5.3.2 Evaluation Methods

We conducted several experiments to evaluate the effectiveness of our proposed method. The average of F1 measure (harmonic mean of the precision and recall) over the sentiment classes as well as accuracy are used as our evaluation criteria. The performance of the following methods was measured.

Sentiment140[5]: a supervised method that discovers the current sentiment for a brand, product, or topic on Twitter, developed by graduate students at Stanford University. This is the baseline.

SentiStrength: a state-of-the-art lexicon based method for classifying the sentiment of short social texts. This is another baseline.

SVM-SS: an SVM classifier trained from the training data labeled by SentiStrength with uni-gram features. No other technique described in Section 5.3 was applied.

SVM-Our: an SVM classifier trained from our target-specific training data described in Subsection 5.2.5 with uni-gram features.

SVM-Our_Sen: an SVM classifier trained from our target-specific training data with uni-gram and on-target sentiment features.

SVM-Our_Usr: an SVM classifier trained from our target-specific training data with uni-gram and user sentiment features.

SVM-Our_All: an SVM classifier trained from our target-specific training data with all the features described in this section.

Note that the sarcasm feature is not used in all systems. The contribution of the sarcasm feature will be solely investigated in Subsection 5.4.7.

We used LIBLINEAR [16] (L2-regularized L2-loss support vector classification) for training the SVM classifiers. The regularization parameter c was optimized by cross-validation on the training data.

5.3.3 Results of the Target-dependent Sentiment Analysis

In this experiment, the tweets were classified into positive, neutral, or negative about the target. Two approaches were evaluated. First, the tweets were classified as positive, neutral or negative in a single step. Second, a two-step classification was performed: the

[5] http://help.sentiment140.com/api/

tweets were classified as subjective or neutral to the target in step 1, then the subjective tweets were classified to positive or negative in step 2. Tables 5.2 and 5.3 show the results of the one-step and two-step classifications. The row 'Average(3)' and 'Average (9)' indicate the average of three targets where the user sentiment feature is used and all nine targets, respectively.

Table 5.2: Results of one-step classification

	Sentiment140		**SentiStrength**		**SVM-SS**		**SVM-Our**		**SVM-Our_Sen**		**SVM-Our_Usr**		**SVM-Our_All**	
Target	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**
iPhone	0.449	0.457	0.602	0.540	0.598	0.560	0.622	0.587	**0.635**	0.587	0.631	**0.593**	0.633	0.587
Xbox	0.570	0.523	0.718	0.603	0.724	0.637	0.709	0.633	**0.729**	**0.663**	-	-	-	-
Nike	0.489	0.433	0.682	0.593	0.696	0.620	0.684	0.610	**0.704**	**0.650**	-	-	-	-
Google	0.488	0.483	0.606	0.517	0.648	0.583	0.646	0.590	0.665	0.607	0.647	0.587	**0.676**	**0.617**
Verizon	0.562	0.510	0.679	0.600	0.708	0.627	0.719	0.660	**0.723**	**0.660**	-	-	-	-
Sony	0.551	0.530	0.622	0.557	0.676	0.607	0.659	0.603	**0.682**	**0.623**	-	-	-	-
Obama	0.301	0.383	0.526	0.447	0.550	0.493	0.526	0.493	**0.559**	0.503	0.544	0.500	0.556	**0.510**
Beyonce	0.412	0.390	0.670	0.563	0.686	0.583	0.696	0.600	**0.703**	**0.607**	-	-	-	-
Messi	0.469	0.453	0.667	0.567	0.660	0.573	0.672	0.603	**0.682**	**0.607**	-	-	-	-
Average (3)	0.412	0.441	0.578	0.501	0.599	0.546	0.598	0.557	0.620	0.566	0.607	0.560	**0.622**	**0.571**
Average (9)	0.477	0.463	0.641	0.554	0.661	0.587	0.659	0.598	**0.676**	**0.612**	-	-	-	-

Table 5.3: Results of two-step classification

	Sentiment140		**SentiStrength**		**SVM-SS**		**SVM-Our**		**SVM-Our_Sen**		**SVM-Our_Usr**		**SVM-Our_All**	
Target	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**
iPhone	0.449	0.457	0.602	0.540	0.603	0.553	0.619	0.563	**0.638**	**0.577**	0.623	0.560	0.636	0.573
Xbox	0.570	0.523	0.718	0.603	**0.721**	0.633	0.718	0.643	**0.721**	**0.650**	-	-	-	-
Nike	0.489	0.433	0.682	0.593	**0.696**	**0.617**	0.675	0.590	0.683	0.610	-	-	-	-
Google	0.488	0.483	0.606	0.517	0.631	0.560	0.641	0.570	**0.674**	**0.603**	0.633	0.557	0.659	0.580
Verizon	0.562	0.510	0.679	0.600	**0.717**	**0.647**	0.704	0.643	0.704	0.643	-	-	-	-
Sony	0.551	0.530	0.622	0.557	**0.676**	0.590	0.651	0.580	0.670	**0.600**	-	-	-	-
Obama	0.301	0.383	0.526	0.447	0.545	0.483	0.532	0.480	0.532	0.470	0.548	0.477	**0.552**	**0.487**
Beyonce	0.412	0.390	0.670	0.563	0.683	0.577	0.689	0.583	**0.702**	**0.597**	-	-	-	-
Messi	0.469	0.453	0.667	0.567	0.649	0.560	**0.661**	**0.590**	**0.661**	0.583	-	-	-	-
Average (3)	0.412	0.441	0.578	0.501	0.593	0.532	0.597	0.538	0.615	**0.550**	0.601	0.531	**0.616**	0.547
Average (9)	0.477	0.463	0.641	0.554	0.658	0.580	0.655	0.583	**0.665**	**0.593**	-	-	-	-

The results show that, on average, the one-step classification was slightly better than the two-step classification. This was because of the low recall (0.27–0.44) and F1 measure (0.36–0.49) of the neutral class in the first step of the two-step classification. This meant that the classification of the subjectivity was rather difficult. With the one-step classification, we found that our methods (SVM-Our_Sen) outperformed the two baselines (Sentiment140 and SentiStrength) by large margins and improved the accuracy over the two baselines by 5.7%–14.9%, respectively. Our methods also improved the performance compared to SVM-SS (where the training data was labeled by SentiStrength only) about 2.5%. Table 5.4 shows P-values of McNemar's test to evaluate the significance of the differences between SVM-Our_Sen and three baselines in the one-step classification. Comparing the results of individual targets, our method significantly outperformed Sentiment140 and SentiStrength at 99% and 90% confident level for the most targets, respectively. SVM-Our_Sen was better than SVM-SS for all 9 targets by 0.5-3.7% F1 measure and 0.1-3.4% accuracy, although the differences were not so significant. Comparing all targets, our method outperformed each baseline at 99% confident level.

Table 5.4: Statistical test of the difference between SVM-Our_Sen and the baseline

	The two-tailed, P-Value		
Target	**Sentiment140**	**SentiStrength**	**SVM-SS**
iPhone	0.0007	0.0660	0.2299
Xbox	0.0002	0.0125	0.2963
Nike	0.0001	0.0472	0.2531
Google	0.0014	0.0009	0.3239
Verizon	0.0001	0.0207	0.5708
Sony	0.0197	0.0251	0.1003
Obama	0.0038	0.0611	0.7656
Beyonce	0.0001	0.1048	0.2482
Messi	0.0001	0.1344	0.1649
All	**0.0001**	**0.0001**	**0.0003**

The user sentiment feature (SVM-Our_Usr) performed well and the combination of both features (SVM-Our_All) achieved the highest performance in average of 3 representative targets. Moreover, the accuracy for Obama was lower than other targets, because many tweets about Obama contain a lot of sarcasm and irony, which requires special techniques. The detailed analysis of the on-target sentiment and user sentiment features will

Table 5.5: Examples of tweets correctly identified by the proposed method

	Target	Gold	Uni-gram	SVM-Our_Sen	Tweet
1	Google	±	+	±	@pellicott1 @Caraidmocharai1 @carpen_rachel @BonnyPortmore I'll Google it. Thx gf!
2	iPhone	±	+	±	this sucks ass my mom is getting th iphone 5s for free and im probably PROBABLY gonna keep her ratchet iphone 4 ?
3	Xbox	±	+	±	HAHAHHA I hate my life. My dog figured out how to turn my Xbox off while I'm playing it. Great
4	Nike	+	±	+	@Elynakhalid Went to Nike and Adidas, wasn't what I expected :/ choices are somewhat limited, best is Nike
5	Beyonce	+	−	+	Beyonce is the greatest performer of our time and none of your favorites could ever & you can hate but you know it's the truth
6	Sony	±	+	±	@iheartKita yes girl! He worked for Sony and they bought all new stuff just for that! I loooove the steelers.
7	Obama	+	−	+	So its wrong for anyone to question Obama on anything but it's perfectly fine to call Republicans racists.
8	Google	+	−	+	@Miranda_Jeranka it said that was translated from dutch. Stupid bing, theres a reason people like google more.
9	iPhone	+	−	+	i HATE this galaxy phone! i NEED my Iphone back!! #TeamiPhone #Always
10	Verizon	+	−	+	@imc00lest Verizon, A1 service and customer service is amazing. I hate sprint and t mobile ?

(+=positve, −=negative, ±=neutral)

be shown in 5.3.6. In addition, Table 5.5 shows the examples of tweets correctly identified only by the proposed method (SVM-Our_Sen). It shows the target topic, the gold label, the output of SentiStrength and the output of SVM-Our_Sen. The results show that our system can predict the sentiment about a given target more precisely, especially when the users express their sentiment to other things but not truly about the target.

5.3.4 Evaluation of the Subjective and Polarity Classification

We conducted two experiments to evaluate the effectiveness of our proposed method for subjectivity and polarity classification tasks. In the subjectivity classification, we considered positive and negative tweets as a subjective class. Note that the balanced tweet corpus consisting of equal numbers of subjective and neutral tweets was used as the training and test data, unlike step 1 of the two-step classification in Subsection 5.3.3. On the other hand, in the polarity classification, the neutral tweets were discarded, and the tweets were classified as positive or negative. Tables 5.6 and 5.7 show the F1 measure of the subjectivity and polarity classification. The results clearly show that our proposed method (SVM-Our_Sen) outperformed the uni-gram model (SVM-SS) by 3.5% in the subjectivity classification task and 1.4% in the polarity classification task when considering all 9 targets. These results indicate that our method is more effective for the subjectivity classification task than the polarity classification task, as we had expected. Our method was mainly designed to distinguish between the tweets that expressed an opinion but not truly about the target, which should be classified as neutral in the target-dependent sentiment analysis. In addition, the on-target sentiment feature performed better than the user sentiment feature, and the combination of both features achieved the highest performance in both the subjectivity and polarity classification tasks in the average of 3 representative targets.

5.3.5 Contribution of the Add-on Lexicon, Extended Target List, and Competitor List

We evaluated the contribution of three target specific data sets: the add-on lexicon, the extended target list, and the competitor list . Table 5.8 compares the methods without one of these three extra data sets, the method with all of them (SVM-Our_Sen), and the

Table 5.6: F1 measure of subjectivity classification

Target	**SVM-SS**	**SVM-Our_Sen**	**SVM-Our_Usr**	**SVM-Our_All**
iPhone	0.608	0.664	**0.680**	0.670
Xbox	0.570	**0.616**	-	-
Nike	0.616	**0.623**	-	-
Google	0.577	**0.627**	0.607	0.613
Verizon	0.719	**0.732**	-	-
Sony	0.561	**0.617**	-	-
Obama	0.539	0.558	0.538	**0.577**
Beyonce	0.589	**0.606**	-	-
Messi	0.659	**0.709**	-	-
Average (3)	0.575	0.617	0.608	**0.620**
Average (9)	0.604	**0.639**	-	-

Table 5.7: F1 measure of polarity classification

Target	**SVM-SS**	**SVM-Our_Sen**	**SVM-Our_Usr**	**SVM-Our_All**
iPhone	0.776	**0.807**	0.792	0.802
Xbox	**0.885**	**0.885**	-	-
Nike	0.84	**0.845**	-	-
Google	0.775	**0.817**	0.787	0.808
Verizon	**0.835**	0.82	-	-
Sony	0.815	**0.825**	-	-
Obama	0.698	0.691	0.711	**0.712**
Beyonce	0.815	**0.846**	-	-
Messi	0.794	**0.823**	-	-
Average (3)	0.750	0.772	0.763	**0.774**
Average (9)	0.804	**0.818**	-	-

baseline trained with only the uni-gram feature (SVM-SS). It shows the average accuracy of three and all categories. The competitor list seems the most useful for a product like the iPhone, while the extended target list performs the best for a company, such as Google. This may be caused by the fact that people usually compare a product with its competitors, while a company has a lot of features or extended targets compared to a product. In addition, the add-on lexicon, where the polarities of the objective and OOV words were estimated, made the highest contribution on average. As discussed in Subsection 5.2.2, since there are a lot of objective and OOV words in informal text such as tweets, the add-on lexicon can supply the necessary information for a target-dependent sentiment analysis.

Table 5.8: Contribution of 3 target-specific data

Method	**AVG (Product)**	**AVG (Company)**	**AVG (Person)**	**AVG (ALL)**
SVM-SS	0.673	0.677	0.632	0.661
+ ALL (SVM-Our_Sen)	0.689	0.690	0.648	0.676
- Extended Target	0.686	**0.677**	0.643	0.668
- Competitor List	**0.678**	0.687	0.645	0.670
- Add-on Lexicon	0.680	0.686	**0.636**	**0.667**

Examples of an add-on lexicon, extended target list, and competitor list are shown in Tables 5.9, 5.10 and 5.11, respectively. One can see that many hashtags that can be used with the given target were added to the add-on lexicon. Most of the extended targets and competitors are also reasonable. Table 5.12 shows the accuracy of the extended target list and competitor list. The accuracy is defined as the proportion of the appropriate target or competitor to all words in the list. We found that our method could precisely construct the extended target list and competitor list.

Table 5.9: Examples of words in the add-on lexicon

iPhone		**Google**		**Obama**	
Positive	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**
#beautiful	#unhappycustomer	wowza	#nervous	#likes	#wakeupamerica
#lovers	freezing	#awesomesauce	#fuck	bless	#rapist
amo	#fuck	okaay	google-worst	#smiles	#illegal
#greatmusic	#problems	#googleedu	#translate	hahahaa	blind
#teamapple	#iphoneproblems	#greatproduct	#annoyingthings	#heroic	#dictator
#app	#frustrated	#search	#ridiculous	#saved	#terrorism
#wickedawesome	#autocorrect	hihihi	#torture	#fashi	#radicalislam

5.3.6 Contribution of On-target Sentiment and User Sentiment Features

Table 5.13 shows the average F1 measure when the on-target sentiment features and (non-target-specific) sentiment feature were used. The on-target sentiment feature is derived from the weighted sum of the scores of the sentiment words as in Equation (5.3), where the weights are defined as the distance between the sentiment words and the target or competitor, while the sentiment feature is derived from the non-weighted score of the sentiment lexicon. The results reveal that the on-target sentiment feature helps the classifier to improve the performance for the target-dependent sentiment classification in all tasks. Improvements of 2.1% and 1.1% are found in the subjectivity and polarity classification, respectively, which are consistent with the results in Tables 5.6 and 5.7. This is because some polarity words that do not truly express a sentiment about the target are less considered in the model with the on-target sentiment feature.

Table 5.14 shows the average number of on-target tweets per user and the average difference in F1 measure between the model with the user sentiment feature (SVM-Our_Usr) and without (SVM-Our). The user sentiment feature was able to improve the F1 measure for a product (iPhone) and person (Obama) but not for the company (Google). We guess there are two major reasons. First, the sentiments of people might be more consistent for product or people entities than for a company. People who have a positive or negative feeling about some product or person usually express the same sentiment about it in

Table 5.10: Examples of the words in the extended target list

iPhone	**Google**	**Obama**
chargers	fiber	america
ipod	app	action
battery	play	michelle
ios	translator	speech
apple	search	policy
ipad	android	administration
itunes	news	americans
app	store	threat
sprint	chrome	pres
cable	maps	president
charger	nexus	barack

Table 5.11: Examples of the words in the competitor list

iPhone	**Google**	**Obama**
droid	mozilla	bush
android	apple	mitt
samsung	xiaomi	bibi
galaxy	alibaba	congress
htc	duckduckgo	putin
blackberry	firefox	romney
xperia	searchblox	walker
sony	bing	gop
nexus	venmo	netanyahu
google	penguin	guiliani
moto	cyanogen	republicans

Table 5.12: Average accuracy of the extended target list and competitor list over 9 targets

	Accuracy (%)
Extended target list	78.70
Competitor list	82.81

Table 5.13: Evaluation of the on-target sentiment features

Task	**On-target sentiment features**	**Sentiment features**
3-class classification	**0.676**	0.661
Subjective classification	**0.639**	0.618
Polarity classification	**0.818**	0.807

Table 5.14: Evaluation of the user sentiment features

	iPhone	**Google**	**Obama**
Average number of on-target tweet per user	6.12	5.77	24.62
Average F-1 improvement	**0.8%**	0.1%	**1.8%**

their tweet collection. On the other hand, a person might express difference sentiments about a company due to the variety of aspects of that company. For example, the user may express a positive sentiment about Google's search engine but a negative sentiment about Google's translator. Therefore, the sentiment of the user may not be consistent for a company, especially a big company like Google. Second, the performance of the user sentiment feature also depends on the number of on-target tweets of each user. Because of the limitations of Twitter API, we can download only the last 3,200 tweets of each user. Note that the number of tweets containing the target keyword is much smaller than the limitation, as shown in Table 5.1. Intuitively, the user sentiment feature is less reliable when the size of the tweet corpus is small. Actually, the improvement about Obama, where there are 24.62 tweets per user on average, is greater than for iPhone, where there are only 6.12 tweets. Therefore, other information, i.e. the friendship networks of the users, should be considered to overcome the sparseness of the data of the users' tweets and improve the performance of the user sentiment feature.

5.3.7 Contribution of Sarcasm Feature

In this subsection, the effectiveness of the sarcastic feature is investigated. As described before, the accuracy for the target Obama was low since there may be many sarcastic tweets about Obama. Recall that the sarcastic feature is the additional feature indicating if the given tweet is sarcastic. To evaluate the performance of the sarcastic feature, the following two additional methods were performed.

SVM-Our_Sen_wSAR: SVM classifier trained from our target-specific training data with unigram, on-target sentiment and sarcasm features.

SVM-Our_All_wSAR: SVM classifier trained from our target-specific training data with unigram, on-target sentiment, user-sentiment and sarcasm features.

Table.5.15 shows the F1 measure and accuracy of one-step sentiment classification with and without the sarcasm identification feature. It is found that the sarcasm feature boost the performance of the sentiment classifier for three popular targets (iPhone, Google, Obama), but not improve in average for all nine targets. One possible reason we guess is that there might be more sarcastic tweets about popular targets than rare topics. It makes the sarcasm feature more effective when applied to the popular target, at least in

Table 5.15: Results of target-dependent sentiment analysis with sarcasm feature

	SVM-Our_Sen		**SVM-Our_Sen_wSar**		**SVM-Our_All**		**SVM-Our_All_wSar**	
Target	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**
iPhone	0.635	0.587	0.638	0.590	0.633	0.587	**0.643**	**0.597**
Xbox	**0.729**	**0.663**	0.722	0.660	-	-	-	-
Nike	**0.704**	**0.650**	**0.704**	**0.650**	-	-	-	-
Google	0.665	0.607	0.670	0.613	0.676	0.617	**0.678**	**0.620**
Verizon	**0.723**	0.660	0.721	**0.667**	-	-	-	-
Sony	**0.682**	**0.623**	**0.682**	**0.623**	-	-	-	-
Obama	**0.559**	0.503	0.557	0.507	0.556	0.510	**0.559**	**0.513**
Beyonce	**0.703**	**0.607**	0.702	0.607	-	-	-	-
Messi	0.681	0.607	**0.688**	**0.610**	-	-	-	-
Average (9)	**0.676**	0.612	**0.676**	**0.614**	-	-	-	-
Average (3)	0.620	0.566	0.622	0.570	0.622	0.571	**0.627**	**0.577**

our data.

5.3.8 Error Analysis

To better understand the limitations of our system, we carry out an error analysis of our proposed system (TASK-SEN). The common errors are summarized as follows. (1) Sarcastic tweets are still difficult even though the sarcasm feature is introduced. For example, the tweet “Honestly the app made my iPhone restart 2 times Thank you!” is wrongly classified as positive by TASK-SEN. The sarcasm identification system [76] could not correctly classify it as sarcastic. The method of sarcasm recognition should be improved to gain the performance of the target-dependent sentiment analysis. (2) Subjective tweets that do not contain sentiment terms are often misclassified, such as “Come October, I will be back with my iphone.” This tweet requires the semantic knowledge to refer the positive sentiment to iPhone which could not be handled well in our system. (3) The competitor list is useful to infer the correct sentiment of comparative tweets by inverting the sentiment score of words that express the opinion to the competitor. However, the user sometimes expresses the same sentiment to both target and competitor. For example, “Bing is shit. Google translate shitter than bing.” ‘Bing’ is a search engine including translation service and it is considered as the competitor of Google and our system inverts the negative sentiment of the word ‘shit’ to positive. In such cases, the competitor list will provide the incorrect information to the classifier. (4) Long and complex sentences are rather hard to judge their polarity. Recall that our on-target sentiment feature is based on the weighted sentiment scores of the words, where the weight is defined as the reciprocal of the distance between the sentiment word and the target in a dependency tree, to capture the sentiment toward the target more precisely. However, in some long sentences, the sentiment of the words is weakened due to long distance to the target. Let us consider an example tweet “Is it a sad reflection of our society that my iPhone autocorrects ’gave’ to ’have’...? ”. Since the negative word ‘sad’ is far from the target ‘iPhone’ in a dependency tree of this sentence, TASK-SEN failed to classify it as negative.

5.4 Summary

In this chapter, we have presented a new method for incorporating on-target sentiment information and user sentiment information into a machine learning classifier for the target-dependent sentiment analysis of the tweets. Our method requires no human annotation for the development of the classifier. First, three extra resources, an add-on lexicon, an extended target list, and a competitors list, were automatically constructed from the unlabeled tweets. Then, target-specific training data was created based on heuristic rules and the lexicon-based sentiment analysis method. Two features for training the sentiment classifier were introduced. One is the on-target sentiment feature, giving greater weight to the sentiments of the words near the target; the other is the user sentiment feature, that captures the tendency of the sentiment expressed by the same user. The results of the experiment indicate that our proposed method is effective and improves the classification accuracy compared to the baseline methods in both the 3-class classification and the subjectivity/polarity classification. In addition, we found that performance of our classifier is improved when integrated with the external sarcasm identification system, especially when it is applied to the very popular targets.

The contribution of the user sentiment feature is not so marked, because it is difficult to prepare a large amount of the tweets posted by a user. In the future, we plan to incorporate other network information, i.e. the social relations of the users, to overcome this problem and improve the performance of the user sentiment feature. Furthermore, we plan to find a sophisticated method to retrieve the relevant tweets and filter the spam and advertising tweets before the polarity classification instead of simply using the URLs as an indicator.

Chapter 6

User-level Sentiment Analysis

This chapter presents an approach to improve the performance of the sentiment analysis at user-level by incorporating an implicit and explicit user similarity network. In Twitter, the sentiments of individual tweets are difficult to classify, but the overall opinion of a user can be determined by considering their related tweets and their social relations. It would be better to consider not only the textual information in the tweets, but also the relationships between the users. In this thesis, we propose a framework that takes into consideration not only the "explicit connections" such as follow, mention and retweet but also the "implicit connections" between users. An implicit connection refers to the relations of users who share similar topics of interest, as extracted from their historical tweet corpus, which contains much data for analysis. The results of experiments show that our method is effective and improves the performance compared to the baselines.

6.1 Background and Motivation

Existing approaches to sentiment analysis mainly focus on classification at the message level, and ignore information from network relations. The individual tweets of the users are difficult to classify, but their overall opinion can be determined by considering their related tweets and their social relations, which can be of benefit for many opinion mining systems. Unlike traditional previous approaches, not only the textual information in the tweets but also the relationship between the users should be taken into account. Two social science theories [2] indicate important phenomena that can apply to social networks:

Homophily : When a link between individuals (such as friendship or other social connection) is correlated with those individuals being similar in nature. For example, friends often tend to be similar in characteristics like age, social background, and educational level.

Co-citation regularity : A related concept, which holds when similar individuals tend to refer or connect to the same things. For example, when two people tweet messages with similar topics, they probably have similar tastes in other things or have other common interests.

Previous approaches that have incorporated network information into a classifier have mainly focussed on the first phenomenon, "Homophily", and define "a link" by the explicitly connected network. Tan et al. used a friendship network such as that from 'follow' and from the 'mention' graph to perform a user-level sentiment analysis [69]. Pozzi et al. used the approval relation based on the retweet graph to solve the same problem, and got satisfying results [57]. In some social networks, however, the presence of explicit link structures is limited. The statistics for Twitter in 2009[1] indicate that 55.50% of the users were not following anyone, 52.71% had no follower, and only 1.44% of the tweets are retweets. Therefore, in real-life situations, a large part of the social network does not contain explicit links, and so the current opinion mining systems do not derive any benefit from the network information. In order to overcome this limitation, we propose a framework that incorporates the "implicit connections", based on similarities between users. Following the "co-citation regularity theorem", we will take implicit connections to refer to the relations between users who share similar interests in topics, as extracted from their historical tweet corpus. This will enable us to use more data for sentiment classification. The hypothesis behind this research follows.

Users who have similar interests and often post messages on microblogging containing similar topics tend to have similar opinions in some areas.

In sum, the goal of this research is to develop a method of classifying the overall sentiments (positive or negative) of users about a certain topic by using textual information as well as both explicit and implicit relationships between users in the social network. We also propose an improved method to discover latent topics in the tweets via an en-

[1] http://www.webpronews.com/wonder-what-percentage-of-tweets-are-retweets-2009-06/

hanced pooling scheme with the conventional Latent Dirichlet Allocation (LDA), called the Hashtag-PMI pooling scheme. In addition, the whole process does not require any human intervention, such as annotation of labeled data. This enables us to apply our method to the sentiment analysis of various targets. Figure 6.1 shows the example of explicit and implicit relations and the final goal of this research.

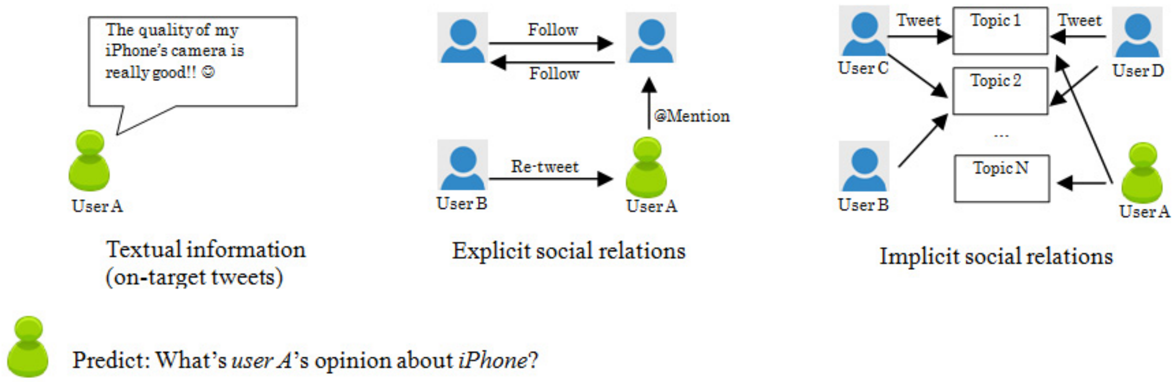

The quality of my iPhone's camera is really good!! ☺
User A
Textual information (on-target tweets)
Follow
Follow
@Mention
Re-tweet
User B
User A
Explicit social relations
Tweet
Topic 1
Tweet
User C
User D
Topic 2
...
Topic N
User B
User A
Implicit social relations
Predict: What's user A's opinion about iPhone?

Figure 6.1: An example of opinion prediction about topic "iPhone" incorporating textual information, explicit and implicit relations

6.2 Proposed Method

This section presents the proposed approach. The system accepts a set of users and a certain topic as input, and classifies the sentiment (positive or negative) of the users for the given topic. An overview of the framework of the system is shown in Figure 6.2. This system is divided into three main parts. First, the implicit relationship between the users is extracted using the LDA with the proposed enhanced pooling scheme. Second, the sentiment of the on-target tweets is classified by a target-dependent sentiment analysis, incorporating target specific knowledge. After that, the information about the implicit relationship, the explicit relationship based on the retweet network and the textual information are incorporated into a heterogeneous factor-graph model. Finally, a loopy belief propagation is applied to predict the sentiment of the users.

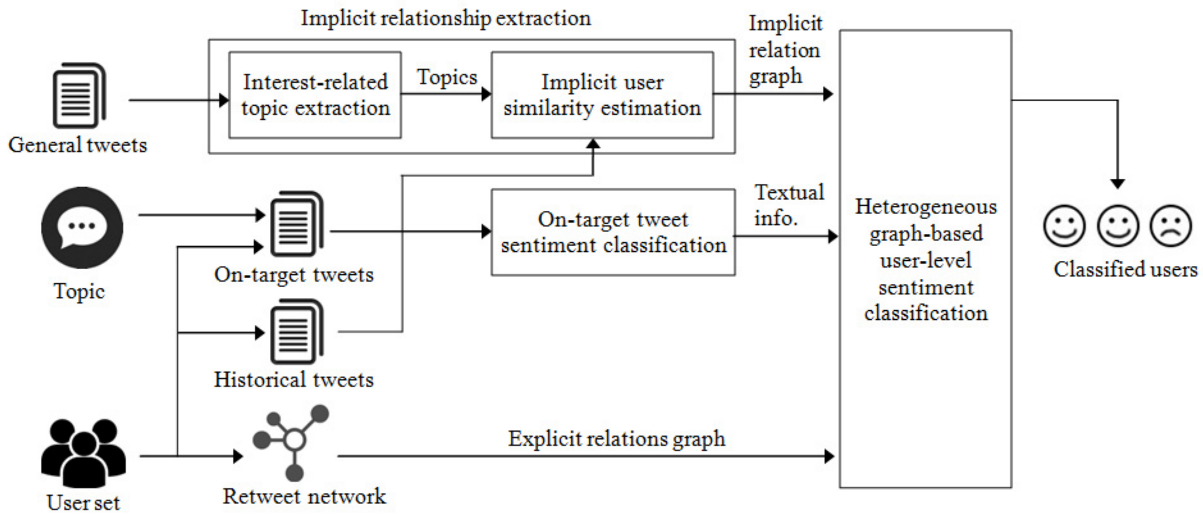

Implicit relationship extraction
Interest-related topic extraction
Topics
Implicit user similarity estimation
Implicit relation graph
General tweets
Topic
On-target tweets
Historical tweets
On-target tweet sentiment classification
Textual info.
Heterogeneous graph-based user-level sentiment classification
Classified users
User set
Retweet network
Explicit relations graph

Figure 6.2: System framework of user-level sentiment analysis

6.2.1 Implicit Relationship Extraction

In this module, we would like to extract the implicit relationship between the users in the social network. As discussed above, the implicit connection refers to the relations of the users who share the similar interested topics extracted from their historical tweet corpus. This module carries out two sub tasks. In the first task, the interest-related topics have been identified by LDA with the enhanced pooling schema. In the second task, the similarity between the users have been estimated based on the cosine similarity in TF-IDF-like vector space.

Interest-related Topic Extraction

We present an alternative way to discover the latent topics in a general tweet corpus using the conventional LDA, called “Hashtag-PMI” pooling scheme, which constructs a document set by aggregating the tweets that likely to express the same topics into the same documents to create better training data for LDA as shown in Figure 6.3. First, the tweets including the same hashtag are merged as a single document. The pooled tweets very likely represent the same topic, since the hashtag can be considered as the topic labeled by the user. The tweets with multiple hashtags are assigned to the multiple document and the tweets without hashtag are left unchanged and unmerged. Finally, the LDA is applied on the set of the aggregated documents to infer the latent topics on the tweet corpus.

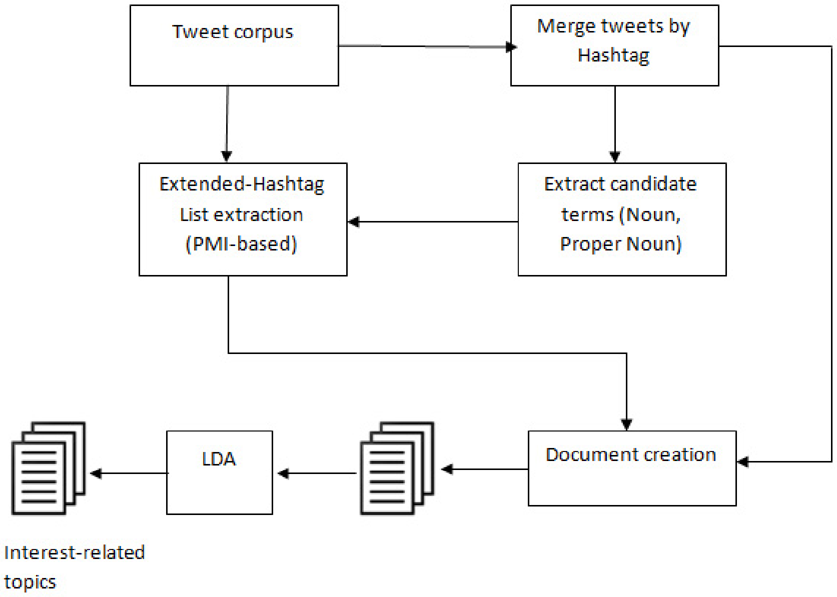

Tweet corpus
Merge tweets by Hashtag
Extended-Hashtag List extraction (PMI-based)
Extract candidate terms (Noun, Proper Noun)
LDA
Document creation
Interest-related topics

Figure 6.3: Interest-related topic extraction process

An additional procedure is performed for pooling the tweets of the same topic. First, correlation between the hashtag H and the candidate terms C (noun and proper noun) is calculated by PMI on the general tweet corpus as shown in Equation (6.1).

$$PMI(H,C) = \log \frac{p(H,C)}{p(H)p(C)} = \log \frac{p(H|C)}{p(H)} \tag{6.1}$$

Then, the candidate terms are selected as an extended-hashtag if their PMI value is greater than a threshold T_PMI. The extended-hashtag is the list of synonyms or terms that usually appear together with a given hashtag. Finally, the hashtag is added to the tweets containing one of the terms in the extended-hashtag list. We believe that the potential relations between the tweets can be captured by the terms (the extended-hashtag) that are highly correlated with a hashtag, even when the terms in those tweets are totally different. In the experimental results presented in Section 6.4, the number of LDA topics is set at 100, a value which was decided on after some preliminary experiments.

The difference between our approach and previous approaches [41, 64] is that instead of assigning the un-hashtagged tweets to the document with the highest textual similarity, we consider the co-occurrence between a hashtag and a term, based on Point-wise Mutual Information (PMI), which explicitly captures the relation between them; even the terms used in those tweets are totally different. Figure 6.4 illustrates our basic idea. In this example, the term ‘iPad’ highly correlated to the hashtag ‘#iPhone’. By adding ‘iPad’ as the extended-hashtag of ‘#iPhone’, the identity of two tweets T1 and T2 can be recognized, although there is no overlapped content word between them.

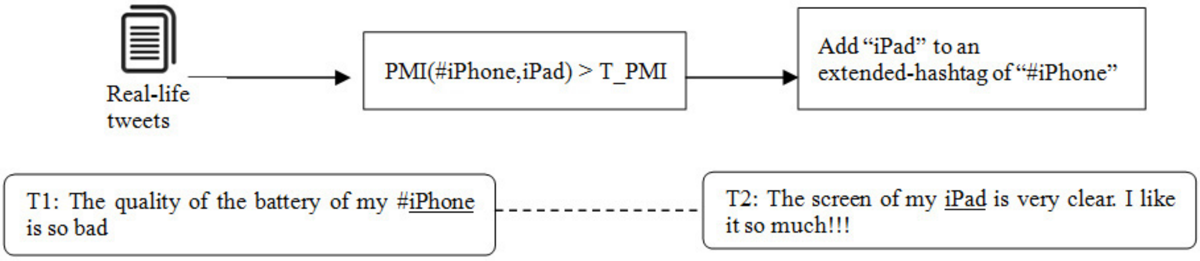

Real-life tweets
PMI(#iPhone,iPad) > T_PMI
Add “iPad” to an extended-hashtag of “#iPhone”
T1: The quality of the battery of my #iPhone is so bad
T2: The screen of my iPad is very clear. I like it so much!!!

Figure 6.4: Potential relationship between tweet T1 and T2 through “#iPhone” and “iPad”

Implicit User Similarity Estimation

The list of the topics inferred from the previous step are used for estimating the implicit similarity between the users as shown in Figure 6.5. First, the topic with the highest probability estimated by LDA is assigned to each historical tweet of the user. Only the tweets that have a probability greater than a threshold, T_IMP1, are selected. This screening is applied for filtering out the interest-unrelated or daily chat tweets. After that, the implicit similarities between each pair of users are estimated by cosine similarity in the topic vector space with modified TF-IDF weighting as shown in Equation (6.2).

$$\begin{aligned} SIM(user_i, user_j) &= \frac{\vec{u_i} \cdot \vec{u_j}}{|\vec{u_i}||\vec{u_j}|} && (6.2) \\ \vec{u_i} &= (\ldots, tf\text{-}idf(topic_k, user_i), \ \ldots)^T && (6.3) \\ tf\text{-}idf(topic_k, user_i) &= tf(topic_k, user_i) \cdot idf(topic_k) && (6.4) \\ &= tf(topic_k, user_i) \cdot \log \frac{N}{df(topic_k)} \end{aligned}$$

where $tf(topic_k, user_i)$ is the number of times that user i tweets about topic k, $df(topic_k)$ is the number of the users who tweet about topic k, N is the total number of users, and $idf(topic_k)$ is the inverse frequency of topic k. Note that $\vec{u_i}$ is T-dimensional vector where T represents the number of the latent topics.

Only the connections between the users whose implicit similarity is higher than a certain threshold, T_IMP2, are preserved. In the experiments, we empirically set T_IMP1 to 0.01 after some preliminary experimentation, and we varied T_IMP2 from 0 to 1.

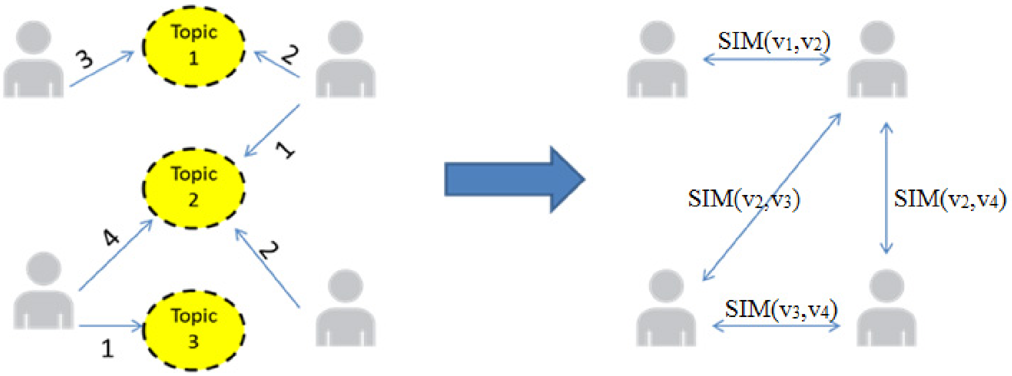


Figure 6.5: Implicit relationship extraction process

6.2.2 On-Target Tweet Sentiment Classification

This module classifies the sentiments (positive, negative or neutral) of the on-target tweets; they will be used as the textual information in the next step (explained in Subsection 6.2.3). We apply the method presented in Chapter 5, called **Ta**rget **S**pecific **K**nowledge **Sen**timent Classification (TASK-SEN), where several techniques are used to improve the performance of target dependent sentiment classification. Let us review the overview of TAKS-SEN. First, a target-specific add-on lexicon is automatically constructed. Second, an extended target list and competitor list are constructed. The former is the list of synonyms of the target. The latter is a list of the competitors of a given target (e.g. a product). Third, not general but target-specific training data is constructed for learning the sentiment classifier. It is automatically created from unlabeled tweets by a lexicon-based method and several heuristics using the extended target list and competitor list. Finally, the classifier is trained from the target-specific training data and add-on lexicon for the target-dependent sentiment analysis.

6.2.3 Heterogeneous Graph-based User-level Sentiment Classification

Starting with the definition of the user-level sentiment analysis task, the proposed heterogeneous factor-graph model will be described. Then, the inference and prediction algorithm on the graph will be explained.

Social Similarity Factor Graph Model

Given a topic q and a set of users $V_q = \{v_1, v_2, \cdots, v_n\}$ who have tweeted about q, the goal is to infer the sentiment polarities $y = \{y_1, y_2, \cdots, y_n\}$ of the users in V_q, where $y_i \in \{pos, neg\}$. For each user $v_i \in V_q$, we have the set of the tweets of v_i about q, $TW_{v_i,q}$, and the explicit relations of the user v_i's retweeting a message of another user v_j. We also have the users' implicit relationship representing that they tend to tweet about similar topics. We incorporate both textual information and the social similarity network (explicit and implicit relationship) into a single heterogeneous factor graph.

For a given topic q, a *Social Similarity Factor Graph*, denoted by $SG_q = \{V_q, TW_{v_i,q}, E_{tw}, E_{ex}, E_{im}\}$ is constructed as in Figure 6.6.

In SG_q, a node is a user in $v_i \in V_q$ or a set of tweets $tw_{v_i} \in TW_{v_i,q}$. There are three types of edges: a user–tweet edge E_{tw} that connects v_i with tw_{v_i}, explicit similarity edges E_{ex}, and implicit similarity edges E_{im} that connect users. E_{tw}, E_{ex} and E_{im} are weighted by $f(v_i)$, $g(v_i, v_j)$ and $h(v_i, v_j)$, respectively, which will be defined later.

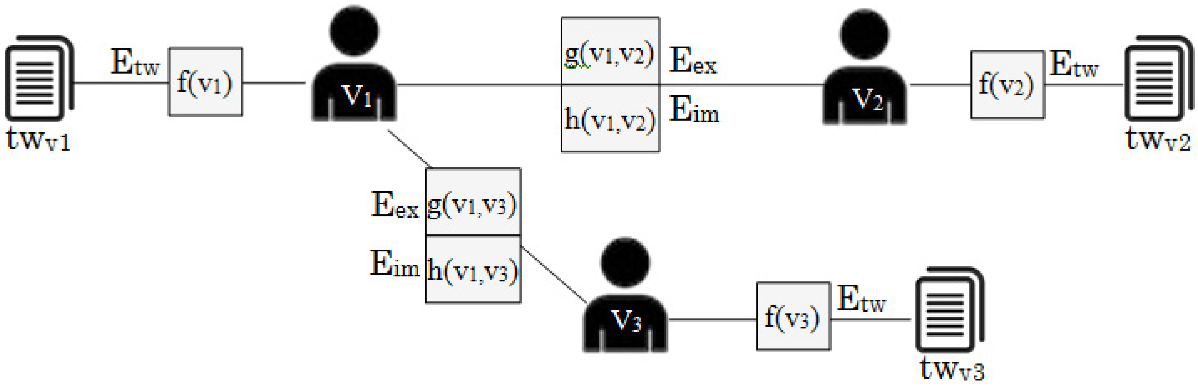


Figure 6.6: An example of a social similarity factor graph

Given the social similarity factor graph SG_q, we would like to classify those users v_i with a given sentiment y_i. Based on the Markov assumption, the sentiment $y_i \in \{pos, neg\}$

of the user v_i is influenced by the sentiment labels of the on-target tweets tw_{v_i} and the sentiment labels of the neighboring users $N(v_i)$. This assumption leads us to adapt the concept of factor graph model defined in Tan et al. [69] and Pozzi et al. [57] to combine the user's tweet with the explicit and implicit user–user relationships, as shown in Eq. (6.5).

$$\begin{aligned}\log(P(y|SG)) = &\Bigg(\sum_{v_i \in V}\Bigg[\log\big(f(y_i|tw_{v_i})\big)\\ &+ \sum_{v_j \in N(v_i)}\Big[\log\big(g(y_i, y_j|v_i, v_j)\big) + \log\big(h(y_i, y_j|v_i, v_j)\big)\Big]/2\Bigg]\Bigg)\\ &- \log Z\end{aligned} \tag{6.5}$$

The first line corresponds to the user–tweet factor and the second line refers to the inclusion of explicit and implicit user–user factors. Z is a normalization factor. We define the feature functions as follows.

The user–tweet factor This function takes into account the sentiment of the user v_i by analysing his/her on-target tweets. The polarity of a tweet $tw \in tw_{v_i}$ is classified by the on-target tweet-level classifier described in Subsection 6.2.2. Note that the neutral tweets are discarded since they represent no sentiment. The user–tweet function is defined as follows:

$$f(y_i|tw_{v_i}) = P_{y_i}(tw_{v_i}) \quad = \frac{fr_{y_i}(tw_{v_i})}{fr_{v_i}} \tag{6.6}$$

where $fr_{y_i}(tw_{v_i})$ is the number of v_i's on-target tweets that are classified as y_i sentiment and fr_{v_i} is total number of on-target tweets that belong to v_i.

The user–user explicit factor This function takes into account the sentiment of the neighboring users connected by retweet relations. The user–user explicit function is defined as follows:

$$g(y_i, y_j|v_i, v_j) = \frac{\#retweet_{i\to j}}{\sum\limits_{v_k \in N(v_i)} \#retweet_{i\to k}}.\delta_{y_i,y_j} \tag{6.7}$$

where δ_{y_i,y_j} is the Kronecker's delta (1 when $y_i = y_j$ and 0 otherwise), and $\#retweet_{i\to j}$ denotes the number of times that v_i retweets v_j's posts.

The user–user implicit factor This function takes into account the sentiment of the neighboring users by analysing the implicit similarity between v_i and their neighbors. The user–user implicit function is defined as follows:

$$h(y_i, y_j | v_i, v_j) = \frac{SIM(v_i, v_j)}{\sum\limits_{v_k \in N(v_i)} SIM(v_i, v_k)} . \delta_{y_i, y_j} \tag{6.8}$$

where $SIM(v_i, v_j)$ denotes the implicit similarity between user v_i and v_j described in Subsection 6.2.1.

Finally, our objective is to maximize the following function with respect to the appropriate sentiment labels.

$$\hat{y} = \arg\max_{\mathbf{y}} \ \log(P(\mathbf{y}|\mathbf{SG})) \tag{6.9}$$

The Inference and Prediction Algorithm

We adapt the loopy belief propagation (LBP) defined in [78] to perform the inference and prediction for a given model, i.e., to approximately maximize the function given in Equation (6.9). LBP is a message passing algorithm for performing inference on graphical models and it has been shown to be a useful approximate algorithm on general graphs. Algorithm 3 shows the pseudo code of LBP. First, the initial labels of the users are assigned through the user–tweet factor function defined in Equation (6.6). Messages $m_{i \to j}(y)$, which represent the degree of influence on the sentiment class y from the node i to j, are inferred by an iterative process. In each iteration, the user–user explicit and implicit factor functions defined in Equation (6.7) and (6.8) are applied to the sentiment messages from the user v_i to v_j. These messages are continuously updated until they are convergent. Lastly, the final sentiment labels of the users are computed based on the value of their neighbors' converged messages, as shown in the last loop in Algorithm 3.

6.3 Evaluation

6.3.1 Dataset

In order to evaluate our proposed system, we used the "Obama Retweet" dataset published by Pozzi et al. [57], which contains 1) a set of users and their sentiment labels about the

Algorithm 3: Loopy belief propagation

```
Input: Social Similarity Factor Graph $SG$
Output: Sentiment label of users $V$
for $(v_i, v_j) \in E_{ex}, E_{im}$ do
    for $y \in \{pos, neg\}$ do
        $m_{i \to j}(y) = 1$
        $m_{j \to i}(y) = 1$
    end
end
do
    for $v_i \in V$ do
        for $v_j \in N(v_i)$ do
            for $y_j \in \{pos, neg\}$ do
                $m_{i \to j}(y_j) = \sum_{y_i \in \{pos,neg\}} \big((g(y_i, y_j) + h(y_i, y_j))/2\big).f(y_i). \sum_{v_k \in N(v_i) \setminus v_j} m_{k \to i}(y_i)$
            end
        end
    end
while all $m_{i \to j}(y_i)$ stop changing || reach maximum iteration;
for $v_i \in V$ do
    $\hat{y}_i = \arg\max_{y \in \{pos,neg\}} f(y). \sum_{v_j \in N(v_i)} m_{j \to i}(y)$
end
```

topic "Obama", 2) a collection of the tweets posted by users about the topic "Obama" and their sentiment labels (called on-target tweets), and 3) the users' retweet network information, consisting of 252 retweet connections. In order to extract the users' implicit relationship, we further downloaded the last 3,200 (as a maximum) tweets of the users through TwitterAPI. Note that all users and posts in the "Obama Retweet" dataset have been manually labeled with their polarity (positive or negative), but we did not use this for either training or classification. These gold sentiment labels were used only for evaluation.

6.3.2 Evaluation of the Graph-based User-level Sentiment Classification

We conducted several experiments to evaluate the effectiveness of our proposed method. Accuracy is used as the evaluation criterion. The performance of the following methods were measured.

Text-only approach (Text-only): The sentiments of the users is computed by a simple majority voting strategy among the labels of their on-target tweets. The on-target tweet-level sentiment classifier described in Subsection 6.2.2 is used as the classification tool.

Social similarity factor graph with explicit relations (SG-Exp): The sentiments of the users are inferred by loopy belief propagation on the factor-graph model with the textual information and explicit user relations.

Social similarity factor graph with implicit relations (SG-Imp): The sentiments of the users are inferred by LBP on the factor-graph model using the textual information and implicit user relations.

Social similarity factor graph with explicit and implicit relations (SG-ALL): The sentiments of the users are inferred by LBP on the full factor-graph model described in this chapter.

Results on the Full Retweet Dataset

In this experiment, we did the experiment on the full "Obama Retweet" dataset, described in Subsection 6.3.1, which contains 62 users and 252 retweet connections. All users had at least one retweet connection. We varied the threshold T_IMP2, which controlled the number of implicit edges in the graph, from 0 to 1. Note that $T\_IMP2 = 1$ means no implicit relation was incorporated in the model. Figure 6.7 shows the results of the graph-based user-level sentiment classification on the full "Obama Retweet" dataset. It shows that the social similarity factor graph with explicit relations (SG-Exp) achieved the best performance, 69.35% accuracy. The social similarity factor graph with implicit relations (SG-Imp) was effective and improved the performance compared to the baseline (Text-only), especially when the implicit similarity threshold (T_IMP2) was greater than 0.4. It reached a highest accuracy of 64.52% (a 3.23% improvement over the text-only method). The combination of explicit and implicit relations (SG-ALL) did not improve the performance compared to SG-Exp. Regardless of T_IMP2, the accuracy of SG-ALL was always same as SG-EXP. This may be because Obama dataset was designed for retweet network experiments. That is, since the number of the explicit links is much higher than the implicit links, the implicit relations could not contribute to improve the performance much.

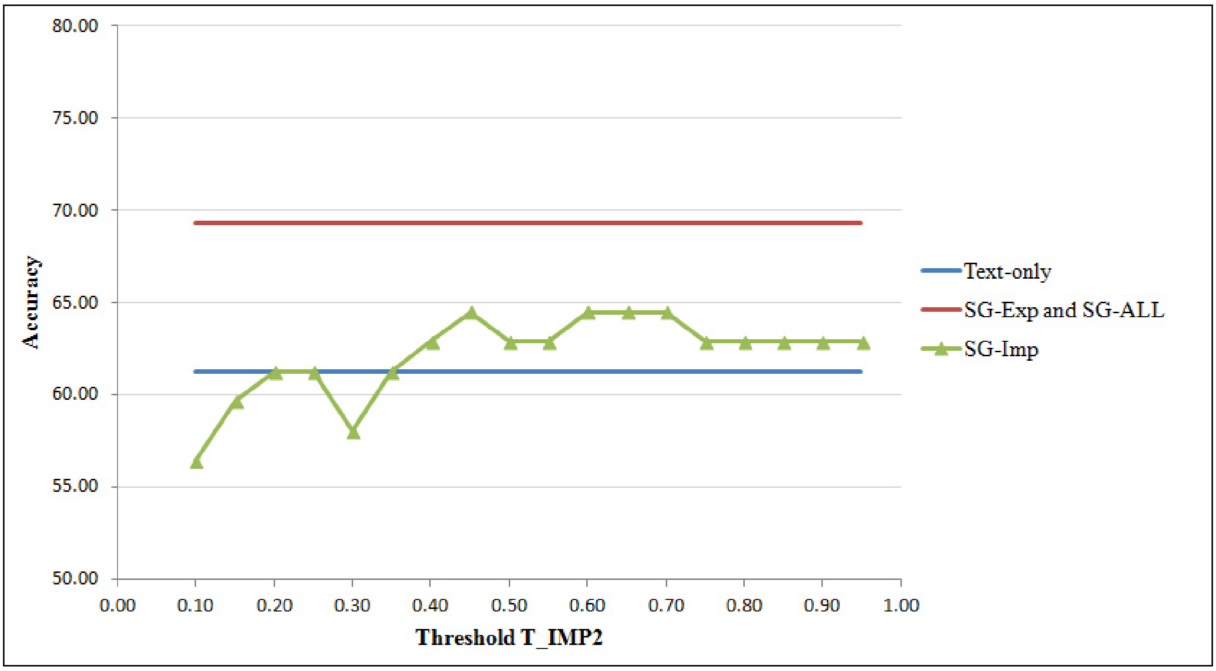


Figure 6.7: Result of graph-based user-level sentiment classification on the full retweet dataset

In sum, we found that the information from the explicit links, like retweet, was more effective than the implicit links. This is not surprising because the explicit links were intentionally created by the users while the implicit links might contain some noise because they were estimated from each user's tweet corpus. However, as we discussed earlier, the presence of such explicit links in real-life situations is limited. Therefore, we conducted another experiment under a more realistic situation.

Results on the Reduced Retweet Dataset

In this experiment, we randomly divided the retweet links in the "Obama Retweet" into 9 parts and constructed 9 datasets, including the retweet connection in only one part. Each dataset contained about 30 retweet links. These datasets are more consistent with a real-life situation in Twitter. Figure 6.8 shows the average of the accuracy of the graph-based user-level sentiment classification over the 9 reduced retweet datasets. It indicates that both SG-Exp and SG-Imp were effective and improved the accuracy compared to the baseline (Text-only) for various T_IMP2 values between 0.5 and 0.95. The combination of explicit and implicit relations (SG-ALL) further improved the accuracy and achieved the best performance compared to other methods when T_IMP2 was set between 0.7 to 0.95. Table 6.1 shows P-values of McNemar's test to evaluate the significance of the differences between our proposed method (SG-ALL) and two baselines. Note that T_IMP2 in SG-ALL is set to 0.7 for the statistical test. It shows that our method significantly outperformed the baseline text-only and SG-Exp at 99% and 85% confident level respectively. These results indicate that our hypothesis in Section 6.1 is correct. The implicit relations, extracted from users' tweet corpora, are useful, especially when there are few explicit relations.

Table 6.1: Statistical test of the difference between SG-ALL and the baselines on the reduce dataset

	The two-tailed, P-Value
Text-only	0.0001
SG-Exp	0.1356

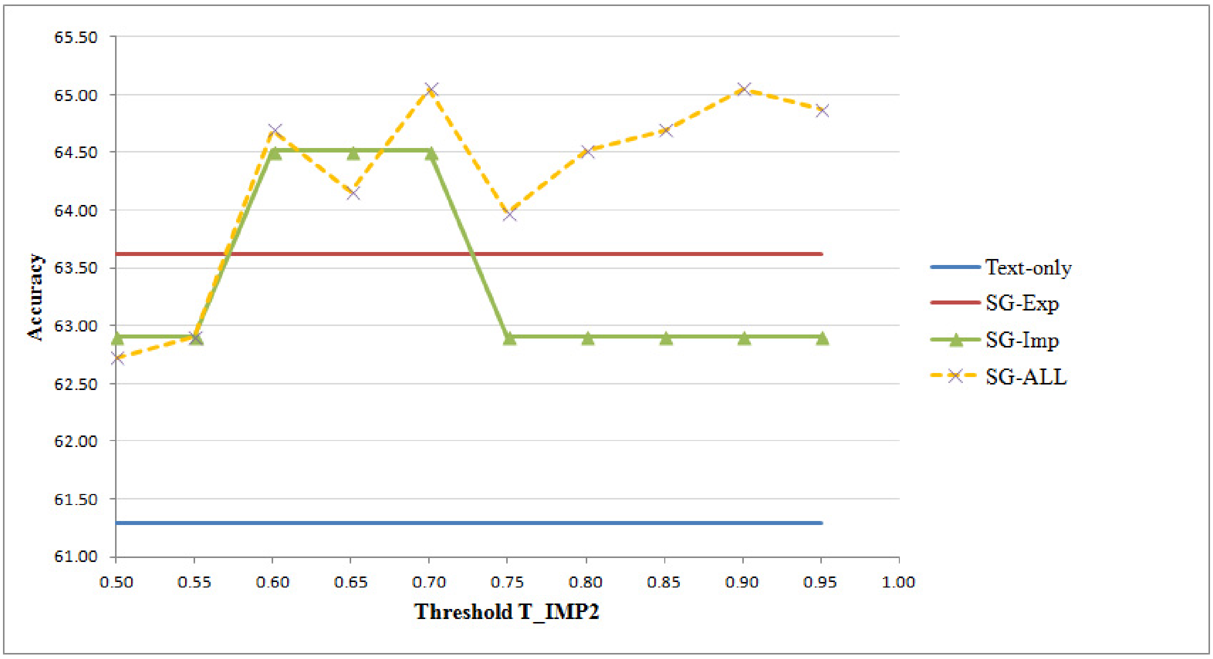


Figure 6.8: Result of graph-based user-level sentiment classification on the reduced retweet dataset

6.3.3 Error Analysis

To better understand the benefits and limitations of our system, we manually inspected the classification result. Table 6.2 and 6.3 show several examples of the users with (1) their gold labels in parentheses, (2) the polarity identified by Text-only, SG-Exp and SG-All, (3) other users connected by the explicit and implicit links with their gold labels and (4) their on-target tweets. T_IMP2 is set to 0.7 to obtain the implicit links.

Table 6.2 shows examples of the users correctly identified only when the implicit relationship is incorporated. We found that some individual tweets are difficult for the SVM classifier to classify. For example, the tweet “ObamaCare encourages folks not to grow up until they’re 26 with healthcare coverage.” of user ‘14818207’ needs some topic specific knowledge to classify it as negative tweet. Moreover, no information is obtained from the explicit links for user ‘6471972’ and ‘14818207’. For the user ‘10879802’, SG-Exp can not judge it as negative because the information obtained from the explicit links is insufficient. In such cases, the implicit relations can help to classify the correct label of the users.

On the other hand, Table 6.3 shows examples of misclassification of our system. The

Table 6.2: Examples of users correctly identified by the proposed method

User ID	T	E	A	Explicit Link	Implicit Link	Tweet
6471972(+)	−	−	+	No	9174252(+)	Obama does something right. http://bit.ly/91bazy Now to get rid of Beltway Bandit Pentagon consultant double dippers.
						Good thing Obama”s not getting a littly girly dog http://bit.ly/dYTE

10879802(−)	+	+	−	11090052(−)
				14847940(+)
				14828712(−)
					6035262(−)
					16103584(+)
					14936857(−)
						Obama proposes raising the minimum wage by printing money on our 3D printers #SOTU

						@jrcornthwait I want Obama and Biden to sing the call and response of, ”You remind me of the babe...” to each other

14818207(−)	+	+	−	No	6742412(−)
					15207713(−)
					6035262(−)
					13880632(−)
					16479015(−)
					15638869(−)
						ObamaCare encourages folks not to grow up until they”re 26 with healthcare coverage. http://bit.ly/c24VjO

						Hard hitting money bomb by @marcorubio equating Christ w/ Obama. http://bit.ly/bDCMTB
						I wonder how much money Obama campaign is spending on this FB ad campaign #birthdayfail http://twitpic.com/2b167h

(T=Text-only, E=SG-Exp, A=SG-All)

Table 6.3: Examples of users incorrectly identified by the proposed method

User ID	T	E	A	Explicit Link	Implicit Link	Tweet
15975726(+)	+	+	−	6035262(+)	14709419(−)	Obama did very well #debate08
						How about follow Obama @qwertygod
14828712(−)	−	+	+	15887542(+)	No	A quick way to prove your ignorance is to discover a ”contradiction” between Perry (Amend 14) & Obamacare (Art I). http://is.gd/e61NT #tlot
						Of all the Bush Administration atrocities Obama could dismantle, he picks No Child Left Behind.

15627816(+)	−	+	−	14099695(+)
				14182457(+)
				14427857(+)
				15487858(−)
					15251890(−)	I wonder if Obama will just sweep up everything #VoteObama

						thats right let him know the facts obama #current
						TELL IT OBAMA #current

(T=Text-only, E=SG-Exp, A=SG-All)

user ‘15975726’ is positive, and can be correctly classified by the textual information only. The positive user ‘15627816’ can also be correctly classified with the help of the explicit links. However, the implicit links of these users are inaccurate, since they connect the positive and negative users. Thus incorporation of the implicit links causes the classification errors. The polarity of the negative user ‘14828712’ can be guessed by textual information only, but the explicit link to the positive user ‘15887542’ makes the system SG-Exp classify this user incorrectly. Since there is no implicit link for this user, SG-ALL also fails to classify.

6.3.4 Performance of Target-dependent Sentiment Classifier

We now present the performance evaluation of our target-specific tweet-level sentiment classifier (TASK-SEN), defined in Subsection 6.2.2, comparing to the previous work as the baseline. We used the collection of 187 tweets about the topic Obama and their manual sentiment labels, described in Subsection 6.3.1. We compared our method with two strong baselines, proposed by Pozzi et al. [57], that are Bayesian Model Average (BMA) and Conditional Random Fields (CRF). Please note that the BMA and CRF are the supervised method and their model has been trained using positive and negative tweets of the Obama-McCain Debate (OMD) dataset [13], while our proposed method is the unsupervised-way method which trained from automatically created training data and requires no human annotation. Table 6.4 shows the result of the tweet-level sentiment classifications. The results showed that accuracy of our method (TASK-SEN) was better than CRF and comparable to BMA. Anyway, the BMA model requires manual labeled training data which needs much human labor, but our method is not.

Next, the contribution of TASK-SEN in the user-level sentiment analysis was investigated. Recall that our graph model incorporates three factors: user-tweet factor, user-user explicit factor and user-user implicit factor. TASK-SEN is used to obtain the user-tweet factor. The question here is how much TASK-SEN can contribute to improve the performance of the user-level sentiment analysis. We compared two systems: one is SG-ALL using TASK-SEN as the target-dependent tweet-level classifier, the other is a system where TASK-SEN is replaced with SentiStrength, which is a state-of-the-art public sentiment classification tool. Table 6.5 shows the accuracy of the user-level sentiment classification

Table 6.4: Performance of the tweet-level sentiment classification

Method	**Accuracy**	**Training data**
CRF	58.49	Obama-McCain Debate dataset [13]
BMA	60.37	Obama-McCain Debate dataset [13]
TASK-SEN	60.38	A corpus of unlabeled tweets

Table 6.5: Accuracy of SG-ALL with TASK-SEN and SentiStrength

	Full Dataset		**Reduce Dataset**	
Method	**with SentiStrength**	**with TASK-SEN**	**with SentiStrength**	**with TASK-SEN**
Text-only	56.45	**61.29**	56.45	**61.29**
SG-Exp	59.68	**69.35**	55.73	**63.62**
SG-Imp ($T\_IMP2 = 0.5$)	56.45	**62.90**	56.45	**62.90**
SG-ALL ($T\_IMP2 = 0.5$)	59.68	**69.35**	56.27	**62.72**

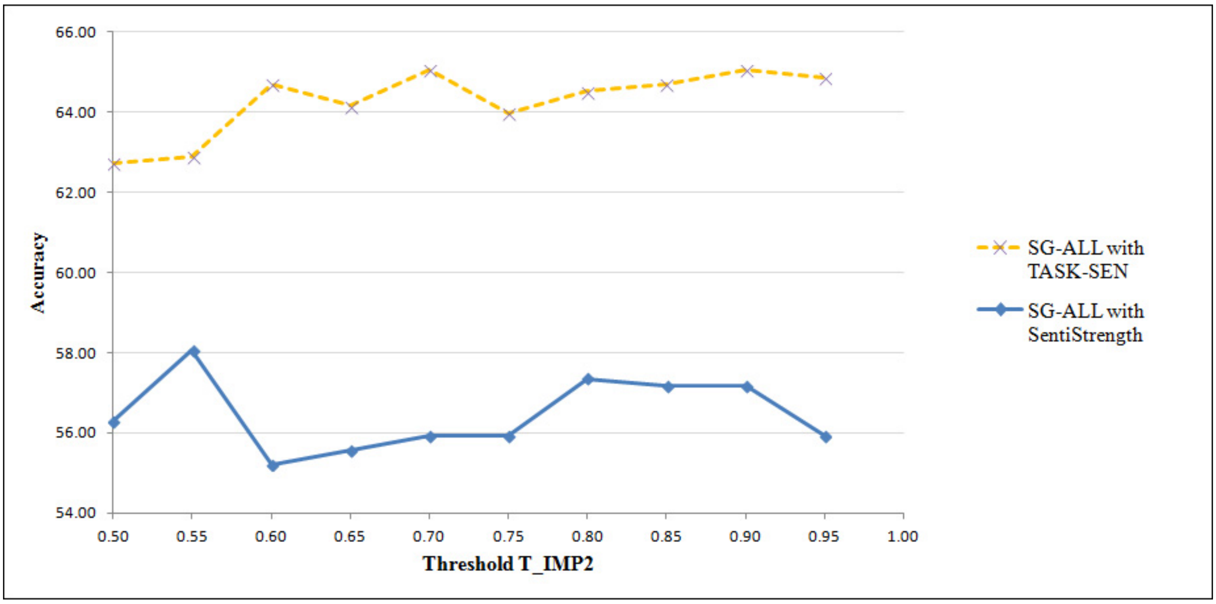


Figure 6.9: Performance of SG-ALL with TASK-SEN and SentiStrength for different values of T_IMP2

with TASK-SEN and SentiStrength on both full and reduced Obama dataset. T_IMP2 in SG-Imp and SG-ALL is set as 0.5 in ad hoc manner. The result shows that the performance of the system with TASK-SEN outperformed the one with SentiStrength by almost 10% in the full dataset and 6% in the reduced dataset. Figure 6.9 shows the accuracy of SG-ALL with TASK-SEN and SentiStrength on the reduced dataset with different values of the threshold T_IMP2. The result indicates that TASK-SEN outperformed SentiStrength for all T_IMP2 values with large margin. From these results, we can conclude the followings. (1) The tweet-level classification is very important. If the user-tweet factor is not good enough, the network relationship could not improve the performance of the graph-based classifier because the wrong information from the neighbor node will be passed to. (2) TASK-SEN gives great contribution to the graph-based user-level sentiment analysis system compared to the state-of-the-art public sentiment classification tool.

6.3.5 Performance of Pooling Methods

We now present the performance of the different pooling schemas for LDA topic extraction. Our proposed hashtag-PMI method was compared to the various baselines:

Unpooled: Represent each tweet as a single document.

Author-based: Merge tweets from the same user into one document.

Hashtag-based: Merge tweets that contain the same hashtag into one document. Tweets containing several hashtags are assigned to several documents, and tweets without a hashtag are left unchanged and unmerged.

Auto Hashtag Labeling [41]: First, aggregate tweets by using a hashtag-based method. Then, assign the tweets without a hashtag to the document that has the highest textual similarity. In this method, the cosine similarity of the word vector weighted by the term frequency (TF) is used as the similarity measure.

Hashtag-PMI: Our proposed method presented in Subsection 6.2.1. T_PMI was set to 1 based on empirical observations.

We constructed the dataset from StanfordTwitter7[2], a tweet collection posted in June 2009. We chose 14 keywords from "Twitter Suggestion Categories" [22], such as 'politics', 'technology', and 'music'. A subset of StandfordTwitter7 was obtained by searching for tweets with these keywords, one by one. In this experiment, the performance of clustering will be measured to evaluate enhanced pooling schemas, but there was no category or topic label in this dataset. We used the hashtags of the keywords, i.e. #politics, as the gold label of the topic cluster. We divided the dataset into two parts. The tweets that contained the keyword hashtag were used as the test data and the remaining tweets were used as the training data. The tweets in the training dataset were merged into a single document according to the different pooling schemas, then the topics were identified by LDA. For evaluation, the topic with the highest probability estimated by LDA was assigned to each tweet, then the tweets with the same topic were merged into one cluster. The detail of the training and test dataset are shown in Table 6.6. Note that we removed the hashtag used as the gold label from the test dataset before clustering. Purity and Normalize Mutual Information (NMI) were used as the evaluation metrics [40]. Purity is the number of correctly assigned documents divided by the total number of documents. NMI is the mutual information between the set of output clusters and the labeled classes of the documents. Table 6.7 shows the number of documents obtained by pooling, the purity score obtained by each method, as well as its NMI. It indicates that our proposed method was effective and improved the purity and NMI from the strongest baseline (Auto Hashtag Labeling), by 3.2% and 4.4%, respectively. Another interesting characteristics of

[2] http://snap.stanford.edu/data/twitter7.html

the proposed Hashtag-PMI pooling method is that the number of the document is much smaller than the others.

Table 6.6: Dataset used for evaluation of pooling methods

	Training dataset			**Test dataset**		
No.	**Query term**	**Number of tweets**	**Proportion (%)**	**Query term**	**Number of tweets**	**Proportion (%)**
1	Art	52,800	9.05	#Art	2,933	14.5
2	Book	52,625	9.02	#Book	746	3.69
3	Business	64,915	11.13	#Business	1,333	6.59
4	Family	43,596	7.47	#Family	119	0.59
5	Fashion	11,762	2.02	#Fashion	941	4.65
6	Cuisine	42,267	7.25	#Cuisine	914	4.52
7	Health	33,995	5.83	#Health	1,431	7.08
8	Politics	44,795	7.68	#Politics	2,695	13.33
9	Science	8,747	1.5	#Science	707	3.5
10	Technology	34,720	5.95	#Technology	624	3.09
11	travel	17,031	2.92	#travel	2,130	10.53
12	Sport	41,592	7.13	#Sport	1,342	6.64
13	Music	77,587	13.3	#Music	3,050	15.08
14	Movie	56,926	9.76	#Movie	1,258	6.22
	Total	**583,358**	**100**	**Total**	**20,223**	**100**

6.4 Summary

This chapter presented a novel graph-based method that incorporates the information of both textual information, as well as the explicit and implicit relationships between the users, into a heterogeneous factor graph for the sentiment analysis of the tweets at the

Table 6.7: Evaluation of the topic extraction methods

No.	Pooling Method	No. of docs	Purity	NMI
1	Unpooled	581,105	0.3705	0.1608
2	Author-based	325,027	0.3781	0.1679
3	Hashtag-based	533,482	0.3721	0.1615
4	Auto Hashtag Labeling [41]	490,299	0.3885	0.1810
5	Hashtag-PMI	55,318	**0.4205**	**0.2252**

user level. First, the implicit relationship between users is extracted by the LDA with the proposed enhanced pooling scheme. Second, the sentiments of the on-target tweets are classified by a target-dependent sentiment analysis incorporating target-specific knowledge. Third, the information about the implicit relationship, explicit relationship based on the retweet network, and the textual information, are incorporated into a heterogeneous factor-graph model. Lastly, loopy belief propagation is applied to predict the final sentiment of the users. The results of the experiments indicate that our proposed method is effective and improves the classification accuracy compared to the baseline methods that consider only textual information or explicit links. Moreover, we have proposed a new enhanced pooling method, "Hashtag-PMI", to more precisely infer the latent topics by the conventional LDA from the tweet corpus. It outperformed the other state-of-the-art pooling schemas. In addition, the process require no human annotation. It enables us to apply our method to the sentiment analysis of various targets.

One drawback of LDA is that the number of the topics must be defined beforehand. Therefore, we plan to find an effective method that determines automatically the optimal number of topics. The implicit user similarity threshold (T_IMP2) is another important parameter for which we plan to explore a sophisticated method to find an optimal value. In addition, we plan to conduct more experiments with a larger dataset and various topics.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis focused on the task of detecting the sentiment of the users about a specific topic. To address this task, we did not only focus on classifying the sentiment of each tweet by considering the textual information, which was usually short and hard to interpret. We aimed to seek other characteristics in microblogging to extract the extra knowledge for boosting the performance of the sentiment analysis classifier.

While certain studies on the sentiment analysis on microblogging were proposed in recent years, our approach differed from the existing research in several ways. The major differences between our proposed system and the previous user-level sentiment analysis approach can be summarized as follows.

(1) We developed the target-dependent sentiment classification system, called "TASK-SEN", which could more precisely classify the tweets according to the sentiment toward the target. We used TASK-SEN as a tool for classifying the users' on-target tweets. Then, the estimated polarity of the tweet was incorporated to the factor graph-based classifier as the user-tweet factor information. The experimental result, presented in Chapter 6, shown that TASK-SEN could improve the overall performance of user-level classification by around 7% accuracy compared to the system using the public state-of-the-art sentiment classification tool.

(2) It was the first work that incorporates the implicit relation between the users to the factor graph-based classifier for user-level sentiment analysis on microblogging. We

estimated the implicit users' relation from their common interested topics extracted by LDA with our proposed pooling schema. Then, the implicit users' relation was incorporated to the factor graph-based classifier as the user-user implicit factor information. The experimental result, presented in Chapter 6, shown that our proposed method with the implicit users' relation outperformed the baseline system which using only the textual information or the textual information and the explicit users' relation with 99% or 85% confident level. In addition, we have shown that the proposed pooling scheme was more precisely infer the latent topics in Twitter and outperformed the state-of-the-art pooling methods with 3.2% purity and 4.4% NMI for clustering tweets based on the latent topics.

More detailed contribution introduced by this thesis is listed below:

- We proposed the method to construct the *Add-on lexicon* that compiled the polarity scores of objective words and out-of-vocabulary words from tweet corpora and studied the contribution of it at multiple sentiment analysis tasks.

- We introduced the novel feature weighting, called *Sentiment Lexicon Interpolation*, by interpolating sentiment lexicon score into uni-gram vectors in the Support Vector Machine (SVM) and studied the effect of this feature on the tweet-level sentiment analysis.

- We proposed the novel system that incorporated target specific sentiment information, including the *Extended target list* and *Competitor list*, and user-sentiment information into a machine learning for the target-dependent sentiment analysis, called ***Ta****rget* ***S****pecific* ***K****nowledge* ***Sen****timent Classification (TASK-SEN)*. We also proposed the method for automatically constructing the *Target-specific training data* based on heuristic rules and lexicon-based sentiment analysis method.

- We proposed the novel graph-based approach of classifying the overall sentiments of users about a certain topic by using textual information as well as both explicit and implicit relationships between the users in the social network.

- We proposed the improved method to discover latent topics in the tweets via an enhanced pooling scheme with the conventional Latent Dirichlet Allocation (LDA), called the *Hashtag-PMI pooling scheme*.

Note that the whole processes of our user-level sentiment analysis framework did not require any human intervention, such as annotation of labeled data. This enabled us to apply our method to the sentiment analysis of various targets.

To this end, we investigated our four research questions and proposed the solution as follows:

Q1: How to overcome the data sparseness problem due to the informal language usage. In Chapter 4, to address this problem, we have proposed an alternative hybrid method that incorporated sentiment lexicon information into the machine learning method to improve the performance of Twitter sentiment classification at tweet-level. The data sparseness problem could be reduced by two methods. First, we estimated the potential polarity of objective and OOV words and used these words as additional information of the public sentiment lexicon. We described how we revised the polarity of objective and OOV words based on the assumption that the polarities of words were coincident with the polarity of their associated sentences, which seemed reasonable due to the short length of tweets. Second, we proposed an alternative way to incorporate sentiment lexicon knowledge into a machine learning algorithm. We proposed the sentiment interpolation weighting method that interpolated lexicon score into uni-gram score in the feature vectors of SVM.

Q2: How to develop effective methods to predict the sentiment toward a certain target. In Chapter 5, to address this problem, we have proposed a new method for incorporating on-target sentiment information and user-sentiment information into a machine learning classifier for the sentiment analysis of the target. First, three extra resources, the add-on lexicon, the extended target list, and the competitors list, were automatically constructed from the unlabeled tweets. Then, target-specific training data was created based on heuristic rules and the lexicon-based sentiment analysis method. Two new features for training the sentiment classifier were introduced. One was the on-target sentiment feature, giving greater weight to the sentiments of the words near the target; the other was the user sentiment feature that captured the tendency of the sentiment expressed by the same user.

Q3: How to extract the preference similarity of users from the historical tweet corpus. In Chapter 6, to address this problem, first we have proposed a new enhanced pooling method, “Hashtag-PMI”, to more precisely infer the latent interested-related topics by the conventional LDA from the tweet corpus. Then, the list of the topics inferred from the previous step was used for estimating the implicit similarity between the users based on cosine similarity in the topic vector space with modified TF-IDF weighting, where the TF referred to the frequency of the ‘topic’ and IDF referred to the inverse frequency of the ‘user’.

Q4: How to incorporate explicit and implicit user relationship into the sentiment classification algorithm and predict the users opinion about a target. In Chapter 6, to address this problem, we have proposed a novel graph-based method that incorporated the textual information as well as the explicit and implicit relationship between the users into a heterogeneous factor graph for the sentiment analysis of the tweets at the user level. First, the implicit relationship between users was extracted by the LDA with the proposed enhanced pooling scheme. Second, the sentiments of the on-target tweets were classified by a target-dependent sentiment analysis with target-specific knowledge. Third, the information about the implicit relationship, explicit relationship based on the retweet network, and the textual information, was incorporated into the heterogeneous factor-graph model. Lastly, loopy belief propagation was applied to predict the final sentiment of the users.

To this end, we investigated and evaluated the effectiveness of our proposed methods in multiple sentiment analysis tasks, including tweet-level, target-dependent and user-level sentiment analysis, which enabled us to deeply understand the problem of sentiment analysis in different points of view. Both public and real-life tweet corpus were used in our experiments. The results of experiments showed that our methods were effective and improved the performance compared to the several baselines.

7.2 Future work

The future research directions for sentiment analysis on microblogging include the use of more sophisticated machine learning algorithms such as deep learning. Recently, deep learning becomes a powerful method for discovering semantic representation of the text without feature engineering and has been applied in some previous sentiment analysis approaches [70, 71, 32, 77, 66]. Therefore, we plan to apply deep learning method to improve the performance of our approaches in many angles. First, the word embedding features could be integrated into our proposed framework. It enables the classifier to learn more about semantic of a word and the relationship among words. Second, in the implicit user similarity extraction process, described in Subsection 6.3.1, we can apply deep learning-based techniques, such as Word2Vec[1] [43] and Doc2Vec [31], to extract the similarity between users instead of LDA-based method that is required to fix the number of the latent topics beforehand.

Another direction is addressing other sentiment analysis tasks such as aspect-level sentiment analysis. Since the terms in the extended target, described in Chapter 5, are referred to the list of terms that usually cooccur with the target, we can adopt the similar technique of the extended target list extraction in order to extract the latent aspects of a given target. Then, it is useful to identify the sentiment of the aspects of the target.

Open-domain target sentiment analysis is another task that we plan to further investigate. In this thesis, it is supposed that the target is given as an input. However, it would be better if the system can detect both the target and the sentiment expressed toward it automatically. For example, the technique of named entity recognition (NER) could be applied to extract the target in the tweet.

In addition, most of methods of the target-dependent sentiment analysis rely on the result of linguistic tools such as syntactic parser and POS tagger. However, in some languages, e.g. Thai, the performance of the syntactic analysis and parsing accuracy is significantly lower than those in English. Therefore, we would like to investigate the method for analyzing the opinion on the low resource language, i.e. lack of corpus data, sentiment lexicon and NLP tools, as well.

Finally, although several evaluations were performed in this thesis, there is a room for improving evaluation methodology of our approaches, including a use of bigger dataset, comparison against more previous approaches and assessment on other type of media such as Facebook.

[1] https://code.google.com/archive/p/word2vec/

Bibliography

[1] Robert P Abelson. Whatever became of consistency theory? *Personality and Social Psychology Bulletin*, 1983.

[2] Charu C Aggarwal and Tarek Abdelzaher. Integrating sensors and social networks. In *Social Network Data Analytics*, pages 379–412. Springer, 2011.

[3] Anthony Aue and Michael Gamon. Customizing sentiment classifiers to new domains: A case study. In *Proceedings of recent advances in natural language processing (RANLP)*, volume 1, pages 2–1, 2005.

[4] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[5] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics, 2010.

[6] Adam Bermingham and Alan F Smeaton. Classifying sentiment in microblogs: is brevity an advantage? In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1833–1836. ACM, 2010.

[7] Albert Bifet and Eibe Frank. Sentiment knowledge discovery in twitter streaming data. In *Discovery Science*, pages 1–15. Springer, 2010.

[8] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[9] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[10] Francine Chen and Seyed Hamid Mirisaee. Do topic-dependent models improve microblog sentiment estimation? In *ICWSM*, 2014.

[11] Lu Chen, Wenbo Wang, Meenakshi Nagarajan, Shaojun Wang, and Amit P Sheth. Extracting diverse sentiment expressions with target-dependent polarity from twitter. 2012.

[12] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd international conference on computational linguistics: posters*, pages 241–249. Association for Computational Linguistics, 2010.

[13] Nicholas A Diakopoulos and David A Shamma. Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1195–1198. ACM, 2010.

[14] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240. ACM, 2008.

[15] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL (2)*, pages 49–54, 2014.

[16] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[17] Ji Fang and Bi Chen. Incorporating lexicon knowledge into svm learning to improve sentiment classification. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP)*, pages 94–100, 2011.

[18] Laurene Fausett. Fundamentals of neural networks: architectures, algorithms, and applications. 1994.

[19] Michel Genereux and Roger Evans. Distinguishing affective states in weblog posts. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 40–42, 2006.

[20] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12, 2009.

[21] Vasileios Hatzivassiloglou and Kathleen R McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*, pages 174–181. Association for Computational Linguistics, 1997.

[22] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88. ACM, 2010.

[23] Chihli Hung and Hao-Kai Lin. Using objective words in sentiwordnet to improve word-of-mouth sentiment classification. *IEEE Intelligent Systems*, 28(2):0047–54, 2013.

[24] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.

[25] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics, 2011.

[26] Yongyos KAEWPITAKKUN and Kiyoaki SHIRAI. Incorporation of target specific knowledge for sentiment analysis on microblogging. *IEICE TRANSACTIONS on Information and Systems*, E99-D No.4:959–968, 2016.

[27] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367. Association for Computational Linguistics, 2004.

[28] Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11:538–541, 2011.

[29] Akshi Kumar and Teeja Mary Sebastian. Sentiment analysis on twitter. *IJCSI International Journal of Computer Science Issues*, 9(4):372–373, 2012.

[30] M. Dekhil M. Hsu L. Zhang, R. Ghosh and B. Liu. Combining lexiconbased and learning-based methods for twitter sentiment analysis. 2012.

[31] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[32] Huizhi Liang, Richard Fothergill, and Timothy Baldwin. Rosemerry: A baseline message-level sentiment classification system. *SemEval-2015*, page 551, 2015.

[33] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[34] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.

[35] Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. Emoticon smoothed language models for twitter sentiment analysis. In *AAAI*, 2012.

[36] Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. Automated rule selection for aspect extraction in opinion mining. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[37] Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210. ACM, 2009.

[38] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[39] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[40] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[41] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892. ACM, 2013.

[42] Rada Mihalcea and Hugo Liu. A corpus-based approach to finding happiness. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 139–144, 2006.

[43] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[44] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*, 2013.

[45] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[46] Andrius Mudinas, Dell Zhang, and Mark Levene. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, page 5. ACM, 2012.

[47] Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Theresa Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. 2013.

[48] Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. Affect analysis model: novel rule-based approach to affect sensing from text. *Natural Language Engineering*, 17(01):95–135, 2011.

[49] Debora Nozza, Daniele Maccagnola, Vincent Guigue, Enza Messina, and Patrick Gallinari. A latent representation model for sentiment analysis in heterogeneous social networks. In *Software Engineering and Formal Methods*, pages 201–213. Springer, 2014.

[50] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11(122-129):1–2, 2010.

[51] Tim OKeefe and Irena Koprinska. Feature selection and weighting methods in sentiment analysis. In *Proceedings of the 14th Australasian document computing symposium, Sydney*, pages 67–74. Citeseer, 2009.

[52] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[53] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[54] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[55] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[56] Witold Pedrycz and Shyi-Ming Chen. *Sentiment Analysis and Ontology Engineering: An Environment of Computational Intelligence*, volume 639. Springer, 2016.

[57] Federico Alberto Pozzi, Daniele Maccagnola, Elisabetta Fersini, and Enza Messina. Enhance user-level sentiment analysis on microblogs with approval relations. In *AI* IA 2013: Advances in Artificial Intelligence*, pages 133–144. Springer, 2013.

[58] Quintilien and E. Butler, H. *The Institutio Oratoria Of Quintilian. With an English Translation by H. E. Butler.* V. Heinemann, 1953, 1953.

[59] Daniel Ramage, Susan T Dumais, and Daniel J Liebling. Characterizing microblogs with topic models. *ICWSM*, 10:1–1, 2010.

[60] Fengyuan Ren and Ye Wu. Predicting user-topic opinions in twitter with social and topical context. *Affective Computing, IEEE Transactions on*, 4(4):412–424, 2013.

[61] Filipe Nunes Ribeiro, Matheus Araujo, Pollyanna Goncalves, Fabicio Benevenuto, and Marcos Andre Goncalves. A benchmark comparison of state-of-the-practice sentiment analysis methods. *arXiv preprint arXiv:1512.01818*, 2015.

[62] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112. Association for Computational Linguistics, 2003.

[63] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In *The Semantic Web–ISWC 2012*, pages 508–524. Springer, 2012.

[64] Manos Schinas, Symeon Papadopoulos, Yiannis Kompatsiaris, and Pericles A Mitkas. Streamgrid: Summarization of large scale events using topic modelling and temporal analysis. In *SoMuS@ ICMR*, 2014.

[65] Laura M Smith, Linhong Zhu, Kristina Lerman, and Zornitsa Kozareva. The role of social media in the discussion of controversial topics. In *Social Computing (SocialCom), 2013 International Conference on*, pages 236–243. IEEE, 2013.

[66] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

[67] Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 53–63. Association for Computational Linguistics, 2011.

[68] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[69] Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1397–1405. ACM, 2011.

[70] Duyu Tang, Bing Qin, and Ting Liu. Deep learning for sentiment analysis: successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(6):292–303, 2015.

[71] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565, 2014.

[72] Jiliang Tang, Huiji Gao, and Huan Liu. mtrust: discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 93–102. ACM, 2012.

[73] Harsh Thakkar and Dhiren Patel. Approaches for sentiment analysis on twitter: A state-of-art study. *arXiv preprint arXiv:1512.01043*, 2015.

[74] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

[75] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[76] Piyoros Tungthamthiti, Shirai Kiyoaki, and Masnizah Mohd. Recognition of sarcasm in tweets based on concept level sentiment analysis and supervised learning approaches. In *Proceedings of Pacific Asia Conference on Language, Information and Computing, Phuket, Thailand*, 2014.

[77] Duy-Tin Vo and Yue Zhang. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1347–1353, 2015.

[78] Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1031–1040. ACM, 2011.

[79] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010.

[80] Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136. Association for Computational Linguistics, 2003.

[81] Le Yu, Rong Pan, and Zhangfeng Li. Adaptive social similarities for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 257–260. ACM, 2011.

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

[83] Tian Tian Zhu, Fang Xi Zhang, and Man Lan. Ecnucs: A surface information based system description of sentiment analysis in twitter in the semeval-2013 (task 2). *Atlanta, Georgia, USA*, page 408, 2013.

Publications

International journal

[1] Yongyos Kaewpitakkun, Kiyoaki Shirai. Incorporation of Target Specific Knowledge for Sentiment Analysis on Microblogging. IEICE Transactions on Information and Systems. Vol. E99-D, No. 4, pp.959-968, 2016, April.

International conference

[2] Yongyos Kaewpitakkun, Kiyoaki Shirai, Masnizah Mohd. Sentiment Lexicon Interpolation and Polarity Estimation of Objective and Out-Of-Vocabulary Words to Improve Sentiment Classification on Microblogging. The 28th Pacific Asia Conference on Language, Information and Computation (PACLIC 28). pp.204-213, 2014, December.

[3] Yongyos Kaewpitakkun, Kiyoaki Shirai. Incorporating an Implicit and Explicit Similarity Network for User-level Sentiment Classification of Microblogging. The 14th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2016). pp.180-192, 2016, August.