Sentiment Analysis of Sarcasm on Microblogging

Piyoros Tungthamthiti

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

Sentiment Analysis of Sarcasm on Microblogging

Piyoros Tungthamthiti

Supervisor: Professor Kiyoaki, SHIRAI

School of Information Science
Japan Advanced Institute of Science and Technology

September, 2016

Abstract

Sentiment analysis of sarcasm in microblogging is important in a range of natural language processing (NLP) applications such as text mining and opinion mining. However, this is a challenging task, as the real meaning of a sarcastic sentence is the opposite of the literal meaning. Furthermore, microblogging messages are short and usually written in a free style that may include misspellings, grammatical errors, and complex sentence structures. This thesis proposes a novel method of sentiment analysis on microblogging that enables us to identify orientation and intensity of the sentiment expressed in the tweets, especially in the sarcastic tweets.

First, we introduce a novel method to identify sarcasm in tweets. It is an ensemble of two supervised classifiers: one is Support Vector Machine (SVM) with N-gram features, the other is SVM with our proposed features. Our features represent intensity of sentiment and contradiction of sentiment derived by a naive sentiment analysis of the tweet. In the sentiment contradiction feature, coherence among multiple sentences in the tweet is also considered, which is automatically identified by our proposed method based on unsupervised clustering algorithm. Furthermore, a way to expand concepts of unknown sentiment words is presented to compensate for insufficiency of a sentiment lexicon. Our method also considers punctuation and special symbols, which are frequently used in Twitter. Results of experiments using two datasets show that our proposed system outperforms baseline systems. The accuracy of sarcasm identification on two datasets is 83% or 76%.

Next, we propose a sentiment analysis system designed for handling sarcastic tweets. To train the model to guess the polarity and intensity of the sentiment in the sarcastic tweets, we used a rich set of features, that are our proposed features used for sarcasm recognition as well as the features grounded on several linguistic levels proposed by the previous work. A decision tree with these features is trained to classify the tweets into an 11-scale score in range of -5 to 5. The system is evaluated on the dataset released by the organizers of the SemEval 2015 task 11. The results show that our method largely

outperforms the systems proposed by the participants of the task on sarcastic and ironic tweets.

Finally, we propose a method for developing a sentiment analysis tool that can guess the fine-grained sentiment score for various types of the tweets. The system consists of two steps. At the first step, the given tweets are classified if they are sarcastic by our sophisticated sarcasm recognition method. At the second step, our sentiment analysis system designed for the sarcastic tweets is used to guess the sentiment scores of the tweets that are judged as sarcasm in the first step. On the other hand, for the tweets judged as non-sarcasm, the three existing sentiment analyzers are applied to guess the sentiment score. The results of the experiments show that our proposed two-steps sentiment analysis system outperforms any single sentiment analyzers on a data set consisting of both sarcastic and non-sarcastic tweets.

In addition, as for the application of the proposed method, our technique to recognize the sarcasm is integrated to an existing target-dependent sentiment analysis system. We also show that the integration can improve the performance via the experiments using a relatively small data set consisting of three targets.

Keywords: Sarcasm, Microblogging, Sentiment analysis, Coherence, Concept Knowledge, Machine learning, Clustering

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Assoc. Prof. Shirai Kiyoaki for the continuous support of my Ph.D. study and research, for his patience, encouragement, motivation, enthusiasm, and immense knowledge. In 2013, Assoc. Prof. Shirai Kiyoaki offered me one of the greatest opportunities of my life to join his natural language processing laboratory, Japan Advanced Institute of Science and Technology (JAIST). Since then, he has been giving me a lot of advises and knowledges regarding the area of my studies, researches, and also giving me a lot of help in my daily life. Without his encouragement and support, I would have given up on the completion of this research. I could not have imagined having a better advisor for my Ph.D. study.

I would like to thank the other members of my committee, Assoc. Prof. Minh Le Nguyen, Assoc. Prof. Hieu Chi Dam, Assoc. Prof. Shinobu Hasegawa and Prof. Hiroyuki Iida for their time, insightful comments on my thesis and their excitement and interest in all my professional and personal research routines.

In addition, I acknowledge my external committee member, Assoc. Prof. Hiroya Takamura for sharing his tremendous experience in the field of natural language processing and data mining. I also greatly appreciate his enthusiasm on giving me a lot of suggestions and comments in details to improve the quality of my dissertation.

I am also grateful to my friends at Hong Kong Polytechnic University, Enrico Santus, Hongzhi Xu and their advisor, Prof. Chu-ren Huang for a great collaboration in the 29th Pacific Asia Conference on Language, Information and Computation (PACLIC 29). I want to thank their excellent cooperation and all of the opportunities I was given to conduct my research.

The days would have passed far more slowly without the support of my Thai friends at JAIST. I had an enjoyable moment and memorable time during these 3 years. Thank you for putting up with me and being so supportive of me and my work.

My sincere thanks also go to Dr. Yongyos Kaewpitakkun and Tomotaka Fukuoka for their generosity, friendship and kindness during these three years. They always offered to

help and gave good advice on anything.

Also, I would also like to thank all Shirai laboratory members for their questions and useful comments on my research. The learning curve would have been much steeper without them.

Special thanks to my family. Words can not express how grateful I am to my parents for all of the sacrifices they made on my behalf. I would like to thank for the love, support, and constant encouragement I have gotten over the years. I undoubtedly could not have done this without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With a rapid growth of microblogging such as Twitter, the importance of sentiment analysis on the microblogs is increasing in a filed of natural language processing (NLP). Meanwhile, it is well known that the automatic interpretation of sarcasm is difficult due to its special linguistic features. Understanding the sarcasm in the microblogging is indispensable for a practical use of sentiment analysis, but it is still challenging. In this chapter, we first explain the background of this research, then clarify the goal of this thesis.

1.1 Background

1.1.1 Natural Language Processing

NLP is a technique to handle with the interactions between computers and human natural languages. The process of understanding the information in natural language can be divided into multiple layers, including lexical analysis, morphological analysis, syntactic analysis and semantic analysis. The machine should proceed through each layer (i.e., character, word, phrase, sentence, paragraph, meaning and so on) in order to perceive the information from given texts or speeches efficiently. Each layer is related to many different major tasks in NLP as follows.

- **Lexical analysis**

  Generally, lexical analysis is considered as the first stage of text processing. The task

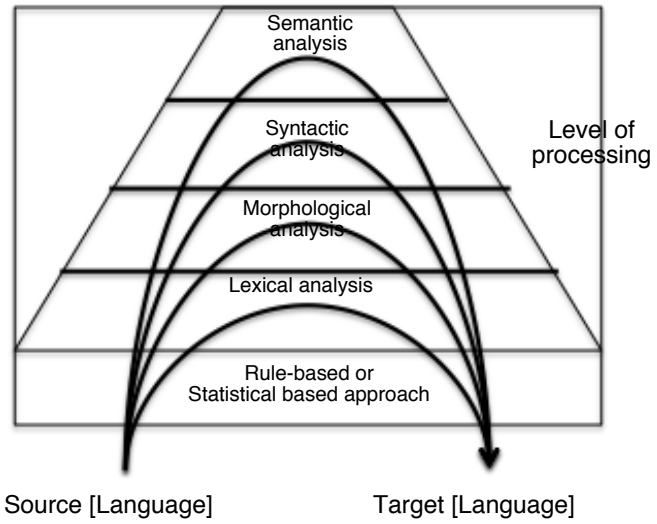

Figure 1.1: Levels of natural language processing

of the lexical analysis is to segment a given sequence of string characters into word tokens or lexemes. This level is capable of determining of word or token and sentence boundary. Then, the tokens will be taken into the next level of the processing such as part-of-speech tagging (POS), parsing and so on. The task is usually performed together with a parser (syntactic anlysis) to analyze the grammatical errors in linguistic or syntax errors in programming languages.

- **Morphological analysis**

  Morphological analysis is a process to identify the structure of words or morphemes. Morpheme is considered as the smallest meaningful unit in linguistic, such as prefix, suffix and root of the word. The task is capable for determining a form of the word and other additional information. For example, "dogs" is divided into the root "dog" and the suffix of the plural form "s". "investment" is divided into the root "invest" and the suffix "ment". It means that the noun "investment" is derived from the verb "invest". Another important task in morphological analysis is POS tagging, where the POS of each word in the sentence is identified.

- **Syntactic analysis**

  Syntactic analysis also known as "parsing" is a process of determining the structure of an input sentence. This task is to ensure that the input sequence of words or symbols

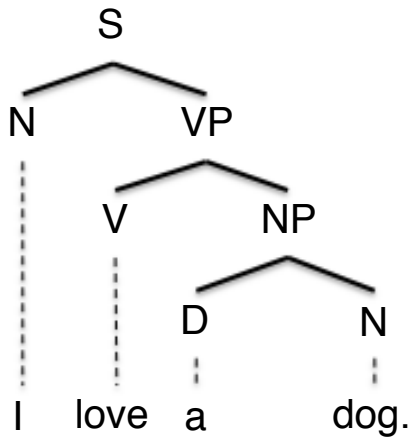

Figure 1.2: Example of syntactic analysis parsing sentence "I love a dog."

conforms to the rules of the grammar in linguistics. Generally, the process describes the structure of an input sentence by a tree where each node represents a word, phrase or clause and the root of the tree represents the overall sentence. Figure 1.2 shows an illustrative example of syntactic analysis of the sentence "I love a dog." The parse tree represents the entire structure, starting from the root S (Sentence), branching NP (Noun phrase) and VP (Verb phrase), and ending in the leaf nodes N (Noun), V (Verb) and D (Determiner).

- **Semantic analysis**

  Semantic analysis refers to the analysis of a meaningful sequence of words or tokens. It can be performed on various levels of textual units, including phrases, sentences, paragraphs or the whole document. The task examines the grammatical and word patterns to determine the actual intended meaning of written texts. In linguistics, it is not always necessary that the correctly written phrases or sentences can have only one intended meaning as their meaning. Let us consider the example of sentence "I love being sick." This sentence is correctly written according to the English grammatical rules. However, in pragmatic meaning, it is unnatural for a human to have such a positive feeling for sickness. Thus, it illustrates an example of sarcastic sentence where the intended meaning is opposite to its literal meaning. In this case, semantic analysis is capable for determining the actual meaning for such sentences. The task is also capable of handling with other types of ambiguous phrases or sentences, such as the usage of idioms or figurative language.

1.1.2 Sentiment Analysis and Opinion Mining on Microblogging

Microblogging is an easy access online information sharing platform with a significant increase in number of usage of social media. It is usually created in a form of blog where people can write and post messages of text or media (pictures, video, or sounds) in limited length. Twitter is one of the most well-known microblogging service, where users can post a short text messages less than 140 characters, called "tweets", to their followers. In Twitter, users can also unidirectionally follow other users and subscribe to their tweets. Since it was launched in 2006, Twitter has been on an explosive growth to a global service with over 200 millions active users [1] generating messages at a peak rate over 230,000 tweets per a minute [2]. The amount of the information sharing and spreading in today's microblogging services is unprecedentedly large.

According to the statistics provided by Twitter, there are more than hundreds millions of individuals who have registered on Twitter and more than billions of new status are updated everyday. Tweets carry the users' views, opinions and sentiments on various kinds of topics, including both personal and businesslike ones. They can be used to keep in touch with friends and family and also to express opinions or broadcast messages on some specific topics. Due to this reason, tweets can become a useful source of information to investigate people's opinions and attitudes on some particular topics.

Sentiment analysis or opinion mining is a task of understanding the subjective information, such as attitudes, emotions and opinions of a speaker or a writer. It is applied in a wide variety of media, including customer reviews, social media, news, chat dialogues, etc. One of the most simple applications is classifying the subjectivity or polarity (neutral, positive, negative) of a given text or speech. For example, sentiment analysis can be applied to online product reviews to determine the polarity of customers' comments on a particular product or service[3]; or infer 5-star rating of the users from written reviews in terms of wide range of categories, such as "product quality", "price", "service", etc.[4, 5]. The sentiment analysis can be applied for business companies to trace their customers' opinions in order to improve the quality of their plans, decisions and strategies.

Table 1.1: Examples of figurative language

Categories of figurative language	**Examples of sentences**
Sarcasm	I love being ignored.
Irony	The trip of our dreams. (In fact, the worst nightmare.)
Metaphor	Life is one long scary roller coaster.
Simile	The cloud was fluffy like cotton candy.
Hyperbole	Her smile is a mile wide.

1.1.3 Literal and Figurative Language

Literal and figurative are two distinct terms that are related to each other in the research field of linguistics. Literal language refers to the use of words to convey the exact meaning or definition as they are given in dictionaries. It is easy to understand and often used to deliver important information, such as scientific, technical and legal documents. In contrast, figurative language is known as the use of words or expressions with a meaning that is deviated from their original interpretation. Figurative language is an effective way to express abstract thoughts. It provides an excellent communication for emotional content. It vibrantly visualizes emotion and imagery in the reader's or listener's mind. Figurative language can make the expressed meaning become easier and more understandable to the readers. The definitions of several types of figurative language are shown below, while Table 1.1 shows the examples of sentences in each category.

Sarcasm – is a form of communication that intends to mock or harass someone by using the opposite meaning of words. It is normally represented in a form of ironic speech in which the speakers convey implicit message to criticize a particular person. The basic purpose of using sarcasm is when bitterness is hard to express in a pleasant way, or in other words to say something without hurting somebody directly.

Irony – is the use of words to convey a meaning that is the opposite of its literal meaning, but its purpose is not intended to hurt other people. It is mainly used to emphasize the meaning of messages by the intentional use of language to say the opposite of the truth. Also, the readers' and listeners' role in realizing the difference between what

is said and what is expected is essential to the successful use of irony.

Metaphor – is a figure of speech which makes an implied or implicit comparison between two different things that are unrelated but share something important in common. They can be very helpful when trying to explain something that's very complicated because it provides a visual description of the word or thought. Furthermore, a metaphor avoids the usage of explicit words "like" and "as" by using implicit or hidden comparison.

Simile – is similar to metaphor since both are used for making comparisons between two different things. However, simile usually uses the words "like" or "as" for making the comparison. The purpose of simile is also similar to metaphor. Generally, writers or speakers try to use simile to visualize a picture inside the reader's or listener's head, in order to make a story become more interesting.

Hyperbole – is the use of overemphasized statement to exaggerate a strong feeling or response. Similarly to other figurative terms, hyperboles are used in speaking and writing to make a boring story become more interesting. Normally, it is used to express excitement, distress, and many other emotions or feelings depending on the context in which the speaker or writer uses.

1.2 Goal

In this research, we aim to create a sentiment analysis system with a particular focus on sarcasm on microblogging. This section presents major problems of sentiment analysis, goal of this thesis, some research questions and an overview of the proposed method.

1.2.1 Statement of problems

Over recent years, the sentiment analysis have become very popular in the area of business, especially in the stock market and e-commerce.

In stock market, there are many researches in both financial and computational linguistic domain showing that news articles can influence the stock market price [6, 7, 8, 9]. News is an important source of information about the situations of everywhere around the world, which is updated every second. For this reason, it contains the information which can influence the stock market prices. For example, a cheap gasoline price will

cause an increase in car sales. According to this example, the stock prices of all vehicle companies will increase due to the lower cost of complementary product. However, due to the great amount of information available on the internet and newspapers, the needs of producing summaries have become more considerable. In e-commerce, merchants selling products on the online marketplace often ask their customers to write a review for the purchased product and the associated service. The reviews can be useful for the sellers as they show how the sellers can improve their products to satisfy their customers. Also, the reviews can be helpful for other customers to make a decision for purchasing the reviewed product. However, as e-commerce is becoming more popular, the number of customer reviews for each product also grows rapidly. It makes difficult for both customers and sellers to handle all of the reviews. To solve the problems regarding the stock market and e-commerce, the sentiment analysis and other related NLP techniques are very helpful.

Apart from the news and product reviews, Twitter is also considered as an important source to gain information about people's opinions in various topics. Many previous studies have shown that tweets also contain the information which can influence both stock market and e-commerce [10, 11, 12]. However, tweets are represented in short messages, where opinions, evaluations and judgments often constitute an important part of the message [13]. The users are allowed to write only short messages of 140 characters per tweet. Also, the users usually post tweets in free or non-restricted writing styles including complex sentence structures. Regarding to these issues, it is difficult to understand the actual meaning of the tweet messages.

Another problem is that tweets are sometimes written as a sarcastic message. Recognition of sarcasm is a problem of determining if an actual meaning of a given tweet is not coincident with a literal meaning. Normally, the sarcasm is used in unpredictable ways in communication (either in criticized forms or in creative ways) and it can involve several linguistic and extra-linguistic levels (i.e. from syntax to concepts and pragmatics). Therefore, identification and understanding of sarcasm is often difficult, even for human beings. Another problem is that the tweet is not a speech, where prosody plays an important role in communication. Although humans are able to rely on prosody (e.g. stress or intonation), kinesis (e.g. facial gestures), co-text (i.e. immediate textual environment) and context (i.e. wider environment), as well as cultural background, machines

are usually hard to access the same type of information. These difficulties pose a major challenge in sarcasm identification of the tweets.

1.2.2 Goal and research questions

The ultimate goal of this research is to create a sentiment analysis system designed for handling sarcastic tweets. It can accept a set of the tweets as an input, and guess an 11-scaled sentiment score between -5 to 5 that represents the polarity and intensity of the opinions in each tweet. The system can handle any types of the tweets, but it introduces special procedures to precisely guess the sentiment score of the sarcastic tweet. The system can potentially provide a lot of benefits to many areas of NLP, such as machine translation, text summarization, word sense disambiguation and knowledge acquisition. Understanding sarcasm enables consumers to obtain more accurate information about people's opinions in various topic domains (e.g. commercial products, business, sports and politics). It prevents us from misinterpreting sentences whose meanings are opposite to their literal meaning. It also allows companies or service providers to know precise opinions about their products or services, which are useful to improve their plans, decisions or business strategies. Therefore, the system can help us to overcome the difficulty in understanding of sarcasm which causes misunderstanding in our daily communication.

This dissertation investigates the following research questions that take the above goal in our mind.

Q1 What are effective features to identify sarcasm in microblogging?

As discussed before, recognition of the sarcasm is useful for various NLP applications but a challenging task. This study aims at developing an effective method to identify the sarcasm in the tweets. Especially, we will explore useful features for sarcasm identification and empirically investigate the effectiveness of them via experiments.

Q2 How to handle informal and short sentences in microblogging in sarcasm identification process?

Processing of the text in microblogging is more difficult than the text in other domains such as newspaper, technical paper or web pages. This study also aims at exploring the way how to precisely and robustly handle informal texts in Twitter

in the task of sarcasm identification. We mainly focus on handling unknown words, since one of the major difficulties for processing of microblogging is that a large number of the words are not compiled in sentiment lexicons.

Q3 How to infer polarity and intensity of sentiment in microblogging, especially in sarcastic tweets?

Sentiment analysis is a fundamental technique for opinion mining and text mining. However, sentiment analysis of the sarcastic text is more difficult than the ordinary text because the genuine meaning of the sarcasm is not coincident with its literal meaning. This study develops a sentiment analysis system that especially focuses on guessing the polarity and intensity of the sarcastic tweets.

Q4 How to develop a general method to infer polarity and intensity of sentiment in microblogging?

Obviously, not all tweets are sarcastic. To improve the robustness, the sentiment analyzer should handle both the sarcastic and non-sarcastic tweets. This study also aims at developing a general system that can precisely analyze the sentiment of various types of the tweets.

We will show the solutions of the research questions Q1 and Q2 in Chapter 3, Q3 in Chapter 4 and Q4 in Chapter 5.

1.2.3 Research methodology and originality

Considering the above research questions, our ultimate goal can be divided into three research objectives or sub goals. Below is each objective and a brief summary of our method to achieve it.

- **To develop a method that can identify sarcasm in tweets**

  The first objective of this thesis is to propose a new method to identify sarcasm in tweets. Our solution is based on supervised learning method that focuses on several features: 1) sentiment score, 2) sentiment contradiction, 3) punctuation & special symbol and 4) N-grams. Support vector machine (SVM) will be used to classify sarcastic

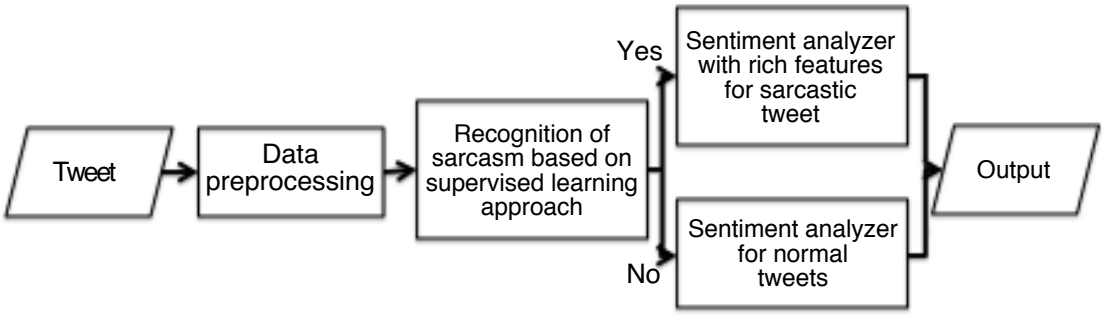

Figure 1.3: Flowchart of overall process of our method

tweet based on our 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.

- **To develop a sentiment analysis system with a particular focus on sarcasm**

  The second objective of this thesis is to introduce a new sentiment analysis system that can guess a polarity score of a given sarcastic tweet. The system is developed by combining and improving two algorithms [14, 15]. In particular, some additional features grounded on several linguistic levels, including token based and polarity dictionary based features, are also used to classify tweets in an 11-scale range.

- **To create a sentiment analysis application for tweets**

  The final objective of this thesis is to create a sentiment analysis application for both normal and sarcastic tweets. The application consists of two main steps: 1) sarcasm identification and 2) sentiment analysis. In the first step, the system checks whether the input tweet is sarcastic or not. Then, the output from the first step will decide which sentiment analyzer will be used to generate the sentiment score for the input tweet. If the tweet is a sarcastic tweet, our proposed sentiment analyzer will be used. Otherwise, the existing tools will be carried out for the task.

Our proposed sentiment analysis system with a particular focus on the identification and proper elaboration of sarcasm in tweets is summarized as shown in Figure 1.3. First, an input tweet is pre-processed by removing stop words, lemmatizing and so on. Next, sarcasm identification system is created to identify whether the input tweet is sarcastic or not. Then, two different sentiment analyzers are build: one is for sarcastic tweets,

the other is for normal tweets. According to the result of the first step, either system is chosen to determine the sentiment score of the given tweet.

One of the important characteristics of our method is that the system considers coherence among multiple sentences in the tweet to derive the sentiment contradiction feature in the task of sarcasm recognition. Although the contradiction of the sentiment is one of the useful clues to identify sarcasm, the contradiction in incoherent sentences might not support that they are sarcastic. We will propose a sophisticated method to identify coherence in the tweets based on unsupervised clustering algorithm. Furthermore, a concept expansion mechanism is introduced to improve the sentiment analysis of the tweets. Since the sentiment analysis often suffers from unknown opinion words that are not compiled in a sentiment lexicon, related concepts of unknown words would be helpful to guess the sentiment polarity of them.

Although there are many types of figurative languages, this thesis only focuses on sarcasm. Furthermore, our method can be applied only for English. However, some knowledge obtained from this study as well as results and discussions presented in this dissertation would be helpful for development of sentiment analysis for other types of figurative language and languages other than English.

1.3 Chapter organization

The remaining chapters in this dissertation is organized as follows.

Chapter 2: Literature review

This chapter discusses various types of machine learning methods. In this part, the explanation mainly focuses on the methods that will be used in this research. The chapter also discusses the related work on sarcasm identification, sentiment analysis and coherence identification task. It is important to examine the validity of existing work and their possible influence on the future development.

Chapter 3: Recognition of sarcasm in tweets based on sentiment analysis and coherence identification

This chapter introduces the method of sarcasm identification. It uses the word N-gram, sentiment score, sentiment contradiction, and punctuation & special symbol as the features for supervised machine learning. In this method, two methods of coherence iden-

tification are proposed to identify the relationship across multiple sentences in the tweets. One is a heuristic-based method and the other is coherence clustering with feature weight optimization (CC-FWO). Concept expansion and concept pruning are also presented to enhance the accuracy of the sentiment analysis feature. Two experiments are conducted to evaluate the performance of our method.

Chapter 4: Sentiment analyzer with rich features for sarcastic tweets

This chapter will introduce a new technique to create a sentiment analyzer with a particular focus on sarcasm in tweets. The idea of the proposed method is to use various kinds of feature based on two modules. Module 1 derives the features used in the sentiment analysis system proposed by Xu et al. [15]. Module 2 derives from our proposed features, including sentiment score, sentiment contradiction and punctuation & special symbols. Experiments are conducted to evaluate the performance of the sentiment analyzer using several data sets.

Chapter 5: General sentiment analyzer for microblogging

This chapter proposes a robust sentiment analyzer that can guess the sentiment score for various texts on microblogging. It is implemented by merging our proposed method into three additional methods: NRC-Canada sentiment analyzer, Stanford sentiment analyzer and SentiStrength. The performance of this sentiment analysis tool is evaluated on the data including four types of the tweets. Furthermore, as the application of our proposed sarcasm recognition method, it is incorporated into the existing tool of a target-dependent sentiment analysis. The contribution of our sarcasm recognition method is empirically evaluated.

Chapter 6: Conclusion

This chapter provides a summary of the entire research in this thesis. The summary includes the answers of the research questions and the contribution of this thesis. Finally, future work of this study is addressed.

Chapter 2

Literature review

In this chapter, we discuss the related work of this research. The chapter is structured into five parts. Section 2.1 introduces the statistical machine learning method that will be used in this research. Section 2.2 will summarize the previous work regarding the aspect of sarcasm in natural language. Section 2.3 will discuss the related work on sarcasm identification task. Section 2.4 will discuss various kinds of techniques for sentiment analysis task. Finally, the chapter will end with Section 2.5 that discusses some related work in the area of coherence and coreference resolution.

2.1 Statistical machine learning methods

Machine learning is a method to automatically acquire or learn a model that can classify entities from a large amount of data. Studies of machine learning are typically classified into four broad categories: 1) supervised learning, 2) unsupervised learning, 3) semi-supervised learning and 4) reinforcement learning. In this section, we introduce common machine learning methods that will be used in this research. Our method relies on only supervised learning and unsupervised learning. Supervised learning is briefly introduced in Subsection 2.1.1. As for unsupervised learning, we use several unsupervised clustering algorithms in our proposed method. Thus, clustering methods are introduced in Subsection 2.1.2.

2.1.1 Supervised learning

In supervised learning, the machine will classify the output into one of the predefined categories based on a set of given examples. The classifiers are trained using labeled examples, where the desired outputs of them are known. The goal of this method is to generate a function that maps inputs to desired outputs. Examples of supervised learning methods are Support Vector Machine, Decision Tree, Random Forest, kNN, Logistic Regression etc.

Support Vector Machine

Support Vector Machine (SVM) is one of the most well-known supervised learning methods [16, 17]. Given a set of training examples containing data in two categories, SVM creates a model that can identify the category of a newly input example. In SVM, a set of training examples L will be considered as data point x_i with D attributes (or D-dimensional vector), and we want to separate such points into two classes $y_i = -1$ or $+1$. The training data can be represented in the following form:

$$\{(x_1, y_1), (x_2, y_2), ..., (x_L, y_L)\} \quad where \quad i = 1, ..., L, y_i \in \{-1, 1\}, x_i \in \Re^D \tag{2.1}$$

A hyperplane will be used to separate the data points. The hyperplane can be described by the Equation (2.2)

$$w \cdot x + b = 0 \tag{2.2}$$

where, w and b are the parameters of the classification model.
The best hyperplane is the one that represents the largest separation, or margin, between the separating hyperplane and the nearest data point of either class. For all training data x_i, Equations (2.3) and (2.4) should be fulfilled.

$$x_i \cdot w + b \geq 1 \quad for\ y_i = +1 \tag{2.3}$$

$$x_i \cdot w + b \leq -1 \quad for\ y_i = -1 \tag{2.4}$$

Figure 2.1 illustrates a hyperplane of two linearly separable classes. Based on the points that lie closest to the separating hyperplane, the planes H_1 and H_2 can be described in the Equation (2.5) and (2.6), respectively.

$$x_i \cdot w + b = 1 \tag{2.5}$$

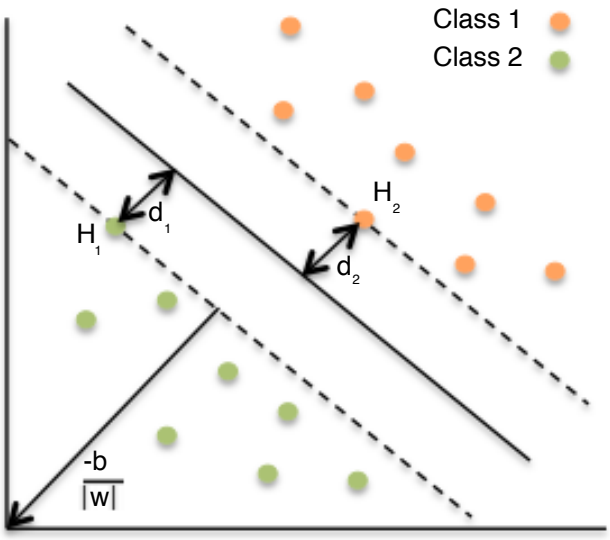

Figure 2.1: Separating hyperplane in Support Vector Machine

$$x_i \cdot w + b = -1 \tag{2.6}$$

In Figure 2.1, d_1 and d_2 are the distance from H_1 and H_2 to the separating hyperplane, respectively. In this case, we consider the variable d_1 and d_2 as margin of SVM. The distance of d_1 and d_2 are always the same ($d_1 = d_2$). Roughly saying, the parameters w and b are determined so that the margin d_1 and d_2 are maximized. Then, the margin can be used to represent the reliable degree of the results generated by the SVM classification.

Decision tree

Decision tree is a machine learning method that creates a classification model in the form of a tree structure. The method breaks down the dataset into smaller subsets while associated decision tree is incrementally developed at the same time. The core algorithm for building decision trees is called ID3 algorithm [18]. The algorithm uses Entropy and Information Gain to construct a decision tree.

A decision tree is built from the root node to leaf nodes. To build a decision tree, we need to calculate two types of entropy as follows.

1. Entropy of the distribution of the target attribute:

$$E(T) = \sum_{i=1}^{c} -p_i log_2 p_i \tag{2.7}$$

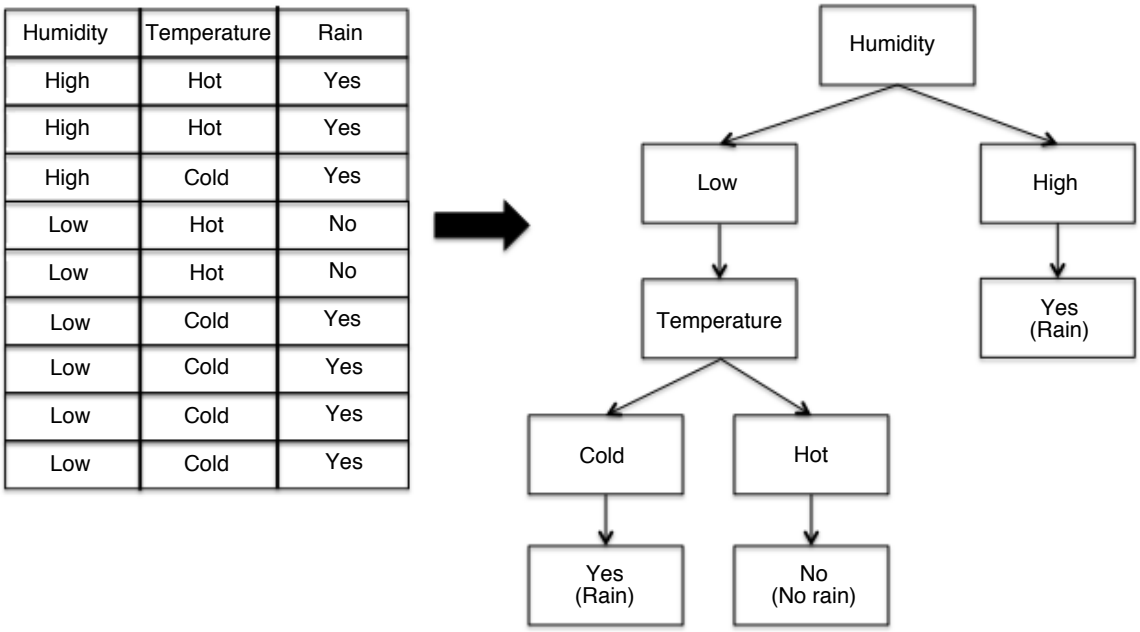

Figure 2.2: Example of Decision Tree training

where p_i stands for the probability of i-th class of the target T. c stands for a number of classes.

2. Entropy of the distribution of the target and branch attribute:

$$E(T, X) = \sum_{c \in X} P(c)E(c) \tag{2.8}$$

where c is an instance of the attribute X. $P(c)$ and $E(c)$ stand for the probability and entropy of c, respectively.

The information gain is analyzed based on a decrease in entropy value after a dataset is split on an attribute. A decision tree is constructed by repeatedly finding attribute that returns the highest information gain. The training procedure consists of five main steps as follows.

1. Calculate the entropy of the target.

In an example of Figure 2.2, the entropy of the target (Rain) can be calculated as:

$$\begin{aligned} E(Rain) &= Entropy(7, 2) \\ &= -(0.78log_2 0.78) - (0.22log_2 0.22) \\ &= 0.76 \end{aligned}$$

where $Entropy(a, b)$ stands for the entropy of binary class distribution as in (2.9).

$$Entropy(a, b) = -\frac{a}{N}\log_2\frac{a}{N} - \frac{b}{N}\log_2\frac{b}{N} \quad \text{where } N = a + b \tag{2.9}$$

2. Calculate the entropy and information gain of the target and branch attribute.
In Figure 2.2, the entropy of the target (Rain) and branch (Humidity) can be calculated as:

$$\begin{aligned} Information\_gain(Rain, Humidity) &= E(Rain) - E(Rain, Humidity) \\ &= 0.76 - ((\tfrac{3}{9} \times E(3, 0)) + (\tfrac{6}{9} \times E(4, 2))) \\ &= 0.76 - (0 + 0.61) \\ &= 0.15 \end{aligned}$$

3. The attribute with highest information gain will be chosen as a decision node. As seen in Figure 2.2, "humidity" is chosen as a decision node at the root.
4. If there is a branch where the target entropy value is equal to 0, it will be considered as a leaf node. As seen in Figure 2.2, the entropy of target attribute is equal to 0 when the humidity is high. Thus, the high humidity is considered as a leaf node. Otherwise, the branch needs to continue splitting.
5. The ID3 algorithm is run recursively on the non-leaf branches, until all training data is classified.

A decision tree can be extended to an regression model that estimates not a discrete class but a continuous numerical value. In this thesis, the decision tree regression model is used to guess the sentiment score of the given tweet.

2.1.2 Clustering method

Clustering is a technique to split or divide a set of instances (or data points) into several groups, called clusters, that share similar characteristics. Clustering is performed in unsupervised manner; no label is given in the data set. Popular algorithms of clustering are K-means, Hierarchical clustering, EM algorithm, Hidden Markov models etc. In general, each data point is represented by a vector, and similarity between two data points is measured by similarity of two vectors.

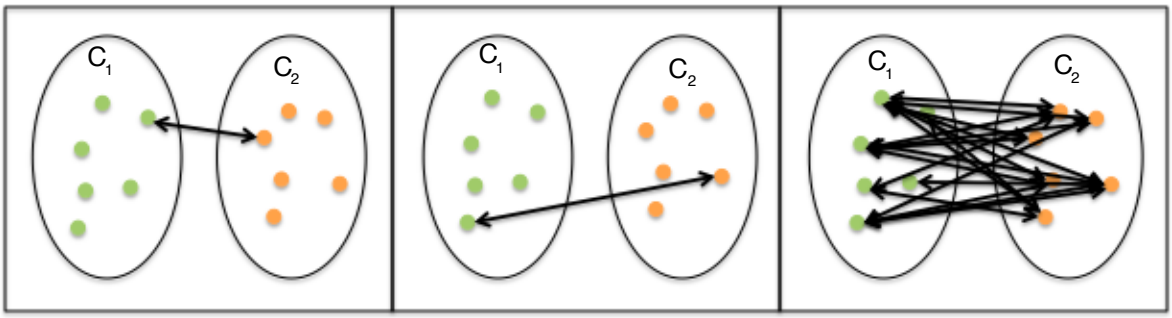

Figure 2.3: Hierarchical clustering: Single Linkage (left), Complete Linkage (center) and Average Linkage (right)

Hierarchical clustering

Hierarchical clustering is a method of cluster analysis which constructs a hierarchy of clusters [19, 20]. The basic procedures of the algorithm consist of four steps. First, each data point is regarded as a single cluster at the beginning. In other words, if there are N data points, there will be N clusters. Each of them contains just one data point. Second, the number of the clusters are reduced by merging the most similar pair of clusters into a single cluster. Third, after the merging, the distance between the newly merged cluster and each of the old clusters is updated. Finally, the procedure in the second and third step will be repeated until the number of clusters is reduced to the specified size of N.

The distance (similarity) between clusters can be measured in three different ways as shown in Figure 2.3.

Single Linkage In single linkage, the distance between two clusters is determined by the shortest distance between two points in each cluster. It is represented as Equation (2.10). L and D are the distances between the clusters and data points, respectively. C_i refers to a cluster, while x_{C_ij} stands for a j-th point in the cluster C_i.

$$L(C_1, C_2) = \min_{i,j}(D(x_{C_1 i}, x_{C_2 j})) \tag{2.10}$$

Complete Linkage In complete linkage, the distance between two clusters is determined by the longest distance between two points in each cluster. It is represented as Equation (2.11).

$$L(C_1, C_2) = \max_{i,j}(D(x_{C_1 i}, x_{C_2 j})) \tag{2.11}$$

Average Linkage In average linkage, the distance between two clusters is determined by the average distance between all pairs of data points in two clusters. It is represented as Equation (2.12), where n_{C_i} stands for the number of data points in the cluster C_i.

$$L(C_1, C_2) = \frac{1}{n_{C_1} n_{C_2}} \sum_{i=1}^{n_{C_1}} \sum_{j=1}^{n_{C_2}} D(x_{C_1 i}, x_{C_2 j}) \tag{2.12}$$

K-means algorithm

K-means is one of the simplest unsupervised clustering algorithms [21]. At the beginning, the number of clusters K is determined and the centres of these clusters are randomly created. The next step is to calculate the distance between each data point and cluster centres. The data point will be assigned to the cluster centre whose distance is the minimum among all cluster centres. Then, the new point of cluster centre will be calculated using the following formula:

$$V_i = (\frac{1}{c_i}) \sum_{j=1}^{c_i} x_j \tag{2.13}$$

where x_j is the vector of the data point, V_i is the vector of the cluster center and c_i is the number of data points in i^{th} cluster. The method continues the same procedure until assignment of all data points is unchanged.

EM clustering algorithm

Expectation-maximization algorithm or EM algorithm is an unsupervised machine learning method. It infers a set of parameters θ from a training (or observation) data x so that $P_\theta(x)$ is maximized, where $P_\theta(x)$ is the probability of the observation data x under the estimated parameter θ. Since it is unsupervised learning, there is no annotation to the training data x. EM algorithm is also frequently used for data clustering. The technique of this algorithm is similar to the K-means. However, the EM algorithm extends this basic approach of clustering by computing the probabilities of cluster membership based on one or more probability distributions. The goal of the clustering algorithm then is to maximize the overall probability or likelihood of the data. The EM algorithm attempts to approximate the observed distributions of values based on mixtures of different distributions in different clusters. The EM algorithm does not compute actual assignments of

data points to clusters, but classification probabilities. In other words, each point belongs to each cluster with a certain probability.

2.2 Linguistic aspect of sarcasm

Sarcasm has been studied since the ancient Greece and Rome. It was, in fact, a part of the basic rhetorical background that all politicians, lawyers and military officers should have had, in order to be able to persuade and convince their audiences. Already in the first century CE, Quintilian defined sarcasm as "saying the opposite of what you mean" [22]. This rhetorical figure violates the expectations of the listener, flouting the maxim of quality [23, 24]. In a similar way, sarcasm is also understood as the use of ironic statements to express disdain in the guise of approval [25]. In sarcasm, ridicule or mockery is used harshly, often crudely and contemptuously, for destructive purposes [26].

According to Stringfellow [23] and Gibbs et al. [27], the usage of sarcasm was studied to derive a definition and demonstrate some characteristics of sarcasm. Both studies agreed on the similar basis that irony and sarcasm arised from the contradictory intentions represented by the opposed meaning of an ironic or sarcastic statement. These studies also discovered the theories of verbal irony comprehension 1) that verbal irony requires a violation of expectations, and 2) that it requires violation of felicity conditions for speech acts. Thus, if we observe both contradictory intentions and violation of felicity conditions within a context, we can recognize a sarcastic context.

Kreuz and Glucksberg claimed that the purpose of using sarcasm and irony was to express disapproval towards situation [28]. In their experiment, they found that positive statements were more readily interpreted as sarcastic. Also, positive sarcastic utterances do not require explicit antecedents (or related situations), while negative ones do. Therefore, we can presume that sarcasm usually occurs in a positive context.

Sarcasm and irony are well-studied phenomena in linguistics, psychology and cognitive science. They are ubiquitous aspect of human communication from ancient religious to modern text styles. There is no consensus on whether sarcasm and irony are essentially the same thing, with superficial differences, or if they differ significantly. In Haiman [29], the main difference between sarcasm and irony is that sarcasm requires the presence of the intention to mock. Irony, instead, can exist independently (i.e. there are ironic

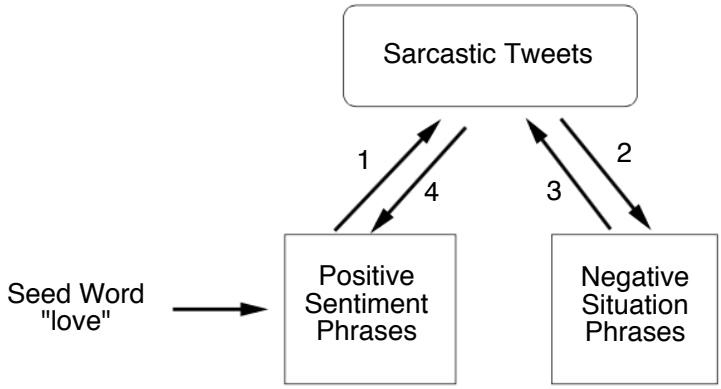

Figure 2.4: Bootstrapping Learning of Positive Sentiment and Negative Situation Phrases

situations, but not sarcastic ones). In Gibbs et al. [27], there is a fine statement to distinguish between sarcasm and irony: "sarcasm is a term commonly used to describe an expression of verbal irony"; whereas "sarcasm, along with jocularity, hyperbole, rhetorical questions, and understatement, are types of irony." Sperber and Wilson distinguished the difference between irony and sarcasm as echoing one's own utterance (irony) and echoing another person's utterance (sarcasm) [30]. Schaffer reported the different verbal clues for irony and sarcasm, such as phonological markers and facial markers [31]. However, there are also numerous studies indicating that there does not appear to be a consensus on how to determine whether an utterance is ironic or sarcastic [32, 33, 34].

2.3 Recognition of sarcasm

In the last several years, many studies related to sarcasm have attracted a lot of attention due to the availability of data [35]. However, algorithms for sarcasm recognition are still far from perfect. Among the several approaches to sarcasm identification, Riloff et al. introduced a novel bootstrapping algorithm that automatically learned lists of positive sentiment phrases and negative situation phrases from sarcastic tweets [36]. The learning process relied on an assumption that a positive sentiment verb phrase usually appeared to the left of a negative situation phrase in a sarcastic tweet. A bootstrapping algorithm continued iteration consisting of the following two steps, which are illustrated in Figure 2.4. The first step was learning negative situation phrases following positive sentiment, where "love" was used as an initial seed of positive sentiment word. Then, the second step

learned positive sentiment phrases that occured near negative situation phrases. After multiple iteration processes, the obtained list of negative situations and positive sentiment phrases were used to identify sarcasm in tweets by checking if the tweet contained a positive sentiment in close proximity (occurring nearby) to a negative situation phrase. This method relied on the assumption that many sarcastic tweets contained the following structure:

$$[+VERB\ PHRASE][-SITUATION\ PHRASE]$$

The result showed that their method yielded some improvement in recall for sarcasm identification. However, the limitation of this method is that it can consider only a number of specific syntactic structures. Also, sarcasm could not be identified accurately when sarcasm appeared in a separate clauses or across multiple sentences.

There are also many different approaches to identify sarcasm. Reyes and Rosso represented irony by six kinds of features, that is n-grams, POS-grams, funny profiling, positive/negative profiling, affective profiling, and pleasantness profiling [37]. Naive Bayesian, Support Vector Machine and Decision Tree were used to train classifiers, achieving an acceptable level of accuracy. Moreover, Reyes et al. proposed a new extended complex model to consider not the surface but deeper semantic level of the text [35]. The method introduced a new set of features in four levels: signatures, degree of unexpectedness, style, and emotional scenarios. They demonstrated that these features did not help the identification of irony and sarcasm when they were independently applied. However, they did when they were combined in a complex framework.

Tsur et al. proposed a semi-supervised method for the automatic recognition of sarcasm in Amazon product reviews [38]. Their method exploited syntactic and pattern-based features and it was compared to a strong heuristic baseline that was built by exploiting the star rating meta-data provided by Amazon (i.e. strongly positive reviews associated with low star rates were considered sarcastic). A similar method was then applied to tweets by Davidov et al. [39], achieving high precision.

Sarcasm in written and spoken interaction may work differently [40]. In spoken utterance, sarcasm can be easily identified through the unsterilized tone of voice [41], a special intonation [42, 43] or an incongruent facial expression [44]. However, in written texts,

there is no clue like a tone of voice, a special intonation or an incongruent facial expression at their disposal [45]. Carvalho et al. investigated the usage of a set of pre-defined surface patterns (i.e. emoticons, onomatopoeic expressions for laughter, heavy punctuation marks, quotation marks and positive interjections) in comments at newspaper articles [46]. They showed that the surface patterns were much more accurate (from 45% to 85% accuracy) than deeper linguistic information. Barbieri and Saggion [47] also proposed a method based on seven sets of lexical and semantic features, including the frequency of the words in reference corpora, their intensity, their written/spoken nature, their length and the number of related synsets in WordNet [48]. Thelwall et al. also aimed at assessing the sentiment lexicon (SentiStrength) in a variety of different online contexts [49]. The results showed that the usage of punctuation, such as a single punctuation, repetitive punctuation marks, question marks and exclamation marks, played a key role to predict the sentiment score. Since punctuation and special symbols such as emoticons are often used to emphasize users' emotion in the tweets, they should be taken into account for sarcasm identification.

Hao and Veale proposed a 9 steps algorithm to automatically distinguish ironic similes from non-ironic ones, without any sentiment dictionaries [50]. Buschmeier et al. assessed the impact of features used in previous studies, evaluated different classifiers and achieved 74% F1-measure using logistic regression [51]. They provided an important baseline for irony detection in English.

2.4 Sentiment analysis

Currently, a large number of researches have been devoted to the area of sentiment analysis. It is an ongoing research in the field of text mining. In this section, we provide an overview of the recent researches in this area. Enhancement and applications of many recently proposed algorithms are briefly investigated.

Sentiment analysis techniques can be roughly divided into three categories: lexicon-based methods [52], machine learning-based methods [53] and hybrid methods [54]. In the lexicon-based approach, the method relies on a dictionary of words with assigned semantic scores to identify a sentiment polarity of a text or sentence. In machine learning-based approaches, the method uses the machine learning algorithms to perform the sentiment

analysis as a regular text classification task. The methods use a large number of training instances that are represented by various kinds of training features. The hybrid approach is a combined method of both lexicon-based and machine learning-based approaches.

Dictionaries for lexicon-based approaches can be created either manually [55, 56] or automatically using seed words to expand the list of words [57, 58, 59]. Regarding the dictionary-based approaches, many researches focused on using adjectives as indicators of the semantic orientation (positive, negative or neutral) of text [60, 61, 62]. First, a dictionary, a list of adjectives and their corresponding sentiment score, is prepared. Then, for any given contexts, the adjectives are extracted and annotated with the sentiment score in the dictionary. Finally, from the sentiment scores annotated to the adjectives, the statistical methods are applied to compute a single score to represent the overall polarity of the given contexts.

Recently, Deepak et al. proposed the machine learning-based method of sentiment classification [63]. In this method, the sentiment polarity was classified based on the aspect term extraction, which was a task to extract aspects or features on which opinions have been expressed [61, 64]. In other words, an aspect term was an attribute or component of the product that had been commented by the user in a review. Let us consider a product review "This camera is good but the weight is heavy." In this example, the aspect terms are "camera" and "weight". The word "good" denotes a positive opinion of the camera and the word "heavy" denotes a negative opinion about the weight. In this research, polarity classification of aspect terms referred to the classification into several sentiment classes such as positive, negative, and neutral. Various kinds of the features were used such as local context, part-of-speech, chunk, root word, stop word, function word, sentence length, etc. The results showed that the system achieved the accuracy of 67.37% and 67.07% for the reviews of restaurants and laptops, respectively.

In Medhat et al. [65] and Pang and Lee [66], we can find a comprehensive study of the different techniques used to identify the polarity of a text. Many efforts have been made to apply such techniques for general text to the text extracted from social media. In the literature, we can find recent attempts to solve this problem using different machine learning approaches such as Support Vector Machine, Maximum Entropy, Naive Bayes, etc. [67, 68, 69]. At best, these studies achieved F1-score close to 70%. Therefore, the

sentiment analysis on social media can still be further improved.

There are still some major problems in applying sentiment analysis to microblogging such as Twitter. Tweets are written text messages, which do not contain much contextual information and also generally require a lot of implicit knowledge to understand. They are also written in a complex grammatical sentence structure and make frequent use of emoticons and hashtags, which can form an important part of the meaning. Typically, they also contain extensive use of irony and sarcasm, which are also difficult for a machine to detect.

Gimenez et al. proposed a new method of sentiment analysis with a particular focus on figurative tweets (irony, sarcasm and metaphor) [70]. Their method used a machine learning, that is Support Vector Machine, for sentiment classification. The method considered various kinds of features, including N-grams, negation context, Twitter features (e.g. hashtags, url, retweets etc.) and character encoding (capitalized words and elongated characters). The results of sentiment classification indicated that the method achieved the best result in sentiment analysis of the tweets including sarcastic ones among the fifteen participated systems of SemEval 2015 Task 11 [71].

Similarly, Raja and Asif proposed a sentiment analysis system to compete in SemEval-2014 Task 9 (Sentiment Analysis in Twitter evaluation challenge) [72]. In this method, Support Vector Machine was also used to perform the classification process. The method utilized a small set of features, including local context, upper case, elongated words, hashtags, repeated characters and negation context. The results showed that their system achieved the F-score in the ranges of 66-76% for contextual polarity disambiguation task and 36-55% for message polarity classification task.

We utilize three existing sentiment analysis systems in this study: 1) NRC-Canada [73], 2) Stanford sentiment analyzer [74] and 3) SentiStrength [49]. Our goal is to create a sentiment analysis system that can handle both sarcasm and normal texts on microblogging. We will propose a method of sentiment analysis for sarcastic tweets, then it will be integrated with these three systems to enable our system to analyze the sentiment of both normal and sarcastic tweets. The explanation of these systems and the integration process will be reported in Chapter 5.

2.5 Coherence identification

As will be reported later, we consider coherence of two or more sentences to identify sarcasm in a tweet consisting of multiple sentences. Coherence among the sentences generally refers to agreement of topics in them. If the topic of the sentence is same or related to that of the previous sentence, we can say that two sentences are coherent. On the other hand, if two sentences mention different topics, they are incoherent. To identify coherence in multiple sentences, coreference resolution can take an important role. Coreference resolution is a task to identify an antecedent of a pronoun such as "he" and "it". If the antecedent is found for the pronoun by coreference resolution, it is very likely that the sentences including that pronoun and antecedent are coherent. Both coherence identification and coreference resolution can be regarded as a kind of context analysis in natural language processing. In past, however, there are much more studies on coreference resolution than coherence identification. This section introduces some previous work of coreference resolution.

In Lehnert et al. [75, 76, 77], a set of manually created rules was proposed to resolve some obvious types of coreference, but they tended to be very conservative. They only considered phrases to be coreferred if there was overwhelmed evidence to support their hypothesis. The method could not figure out which features of the phrases should be looked at when determining coreference. Another problem is how to resolve conflict of positive and negative evidence or how to define a preference order of the rules.

To address these problems, a system called RESOLVE [78] was proposed based on the decision tree. The method used the C4.5 decision tree system [79] to learn how to classify pairs of potential coreference phrases. To train a deccision tree, the method extracted various kinds of features such as name, joint venture child, alias, common noun phrase and same sentence reference. The results showed that the performance of RESOLVE was as good as the manually engineered rule based system in MUC-5. In addition, it was found that some additional features incorporating syntactic knowledge could improve the system to attain a higher level of accuracy.

A more complex method, which is also based on machine learning, was presented by Soon et al. to link coreferring noun phrases both within and across sentences [80]. Twelve features were proposed to create a feature vector. Then, a classifier was trained based on

the feature vectors generated from the training documents. C5 [79, 81] was used as the learning algorithm in this method. The results indicated its performance was comparable to that of state of the art non-machine learning based systems on MUC-6 and MUC-7 standard datasets. Vincent and Cardie [82] also proposed a noun phrase coreference system based on two types of extensions of Soon's method [80]. First, three additional extra-linguistic modifications were introduced to the machine learning framework, which led substantial and statistically significant gains in coreference resolution precision. Second, Soon's feature set was expanded from 12 features to a richer set of 53 features. The additional features were mostly regarded as lexical, semantic and knowledge-based features. The results showed that the method achieved the best results on the MUC-6 and MUC-7 coreference resolution data sets with F-measures of 70.4% and 63.4%, respectively.

Recently, Culotta et al. proposed a machine learning-based method where the features were not simply represented by nouns or noun phrases [83]. The method used arbitrary features using the full expressivity of first-order logic. This enables a more flexible representation of the features. The method was evaluated on the ACE coreference dataset, showing that the first-order logic features could lead to an 45% error reduction.

However, such methods would not be appropriate for our purpose, since they focus specifically on coreference resolution, rather than identifying the coherent relationship. Our coherence identification method will be explained in Section 3.4.

Chapter 3

Recognition of Sarcasm in Tweets Based on Sentiment Analysis and Coherence Identification

In this chapter, we present a novel method to classify if a given tweet is sarcastic or not. Our sarcasm recognition system is based on a supervised machine learning. In addition to conventional N-gram of words, several features for machine learning are derived from the sentiment analysis of the tweet. These features consider intensity of the sentiment and contradiction of the sentiment in the given tweet. Punctuation and special symbols that frequently appear in Twitter are also used as the features. One of the important characteristics of our method is that the system considers coherence among multiple sentences in the tweet to derive the sentiment contradiction feature. Although the contradiction of the sentiment is one of the useful clues to identify sarcasm, the contradiction in incoherent sentences might not support that they are sarcastic. We will propose a sophisticated method to identify coherence in the tweets based on unsupervised clustering algorithm. Furthermore, a concept expansion mechanism is introduced to improve the sentiment analysis of the tweets. Since the sentiment analysis often suffers from unknown opinion words that are not compiled in a sentiment lexicon, related concepts of unknown words would be helpful to guess the sentiment polarity of them.

Figure 3.1 shows the overall process of our system. First, an input tweet is preprocessed by removing stop words, lemmatizing and so on. Next, four kinds of features

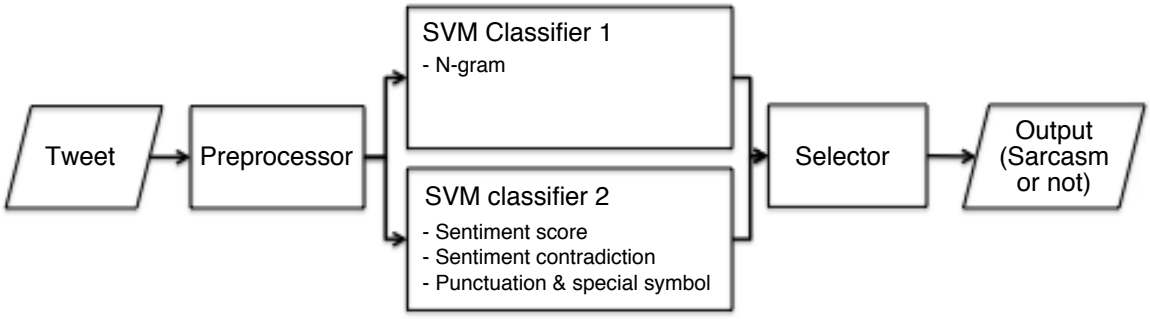

Figure 3.1: Method overview of sarcasm recognition system

are extracted: word N-gram (N = 1, 2, 3), sentiment score, sentiment contradiction and punctuation & special symbol. Next, two classifiers are applied to judge if the given tweet is sarcastic or not. One is Support Vector Machine (SVM) classifier using N-gram features, the other is SVM using the remaining features we propose. These classifiers are trained from labeled data, i.e. a collection of tweets with sarcasm tags. A simple voting method is applied to determine the final judgment. If two classifiers disagree, the result with larger margin, which is distance between a vector of the given tweet and a separating hyperplane, is chosen. In the rest of this chapter, we will explain how to train the classifiers in details.

This chapter is structured into six parts. Section 3.1 begins with the procedures of data preprocessing. Section 3.2 discusses the derivation of our proposed features. Section 3.3 explains the method of concept expansion and pruning. Section 3.4 explains the method of coherence identification. Section 3.5 explains the procedures of sarcasm classification. Section 3.6 describes how the experiment is conducted to evaluate the proposed method and reports the results. Finally, the chapter will finish with a summary in Section 3.7.

3.1 Data preprocessing

In this research, the data in which we try to recognize sarcasm is tweets. However, the tweets are not just the simple plain text data. Sometimes, the tweets contain URL address, twitter user names (mentions) and hashtags. For example, in the tweet "Congrats to @Kelly_clarkson on the birth of her baby GIRL! http://eonli.ne/1vgXVOU #gorgeous",

"@Kelly_clarkson" is a username, "http://eonli.ne/1vgXVOU" is a URL and "#gorgeous" is a hashtag. Users can attach the URL to the tweet when they want provide more information or show an image related to the post. The tweet can also contain a mention feature (@<username>), which allows the notification of other users about the tweet. Hashtags (#<texts>) are used to mark keywords or topics in the tweet. Although the usage of these meta tags are optional, they frequently appear in a lot of tweet messages.

Before the training of SVMs, the tweets in the dataset are preprocessed. The data preprocessing consists of two main steps: 1) lemmatization and 2) user names, URLs and hashtags removal. First, the Stanford Lemmatizer[1] is applied to obtain parts-of-speech (POSs) of the words and also transform the words into their lemmas. Second, user names, URLs and hashtags are removed from the tweets. Since these features are less informative for the sarcasm classification, they shall be removed to reduce noise in the classification process. The same preprocessing is applied when a new tweet is classified by the trained classifier.

3.2 Proposed features

In addition to the ordinary N-gram features, we propose the other three features to characterize the properties of sarcasm. Although the basic ideas of our proposed features are shared with the previous work on the analysis of sarcasm, the ways to extract the features are different. Especially, our sentiment contradiction feature is unique as will be discussed in Subsection 3.2.2. Note that in this study the feature vector is binary: the value of the feature is defined as 1 if it exists in the tweet, 0 otherwise.

3.2.1 Sentiment score features

It is said that sarcasm contains violation of expectations and violation of felicity conditions in the statements [23]. Thus, we attempt to recognize the level of violation and aggressiveness of the words in the tweet. The sentiment score feature represents intensity of positive or negative sentiment in the tweet. It is basically measured by sentiment scores of the words derived from a public sentiment lexicon. Furthermore, our method

[1] http://nlp.stanford.edu/software/corenlp.shtml

also considers concepts of the words to obtain the sentiment score feature. Since many potential sentiment words are not compiled in a sentiment lexicon, our system extract the concepts of the unknown sentiment words to alleviate shortage of a public sentiment lexicon. This procedure is called "concept expansion" in our research. Let us consider an example of sarcastic tweet T1.

T1: I love going to work on holidays.

Suppose that the system can identify only one positive word "love" using a sentiment lexicon, while other words have no polarity. By concept expansion, however, we can recognize that the word "work" refers to "tiring" or "stressful situation" and "holiday" refers to "day where person stay home and relax". Now the system is able to identify two additional sentiment words "holiday" and "work", which illustrate positive and negative sentiment respectively. In this way, concept expansion can compensate for insufficiency of the sentiment lexicon.

In this research, two lexicons are used to obtain the sentiment scores of the words: SentiStrength [49, 84] and SenticNet [85]. SentiStrength is a sentiment analysis tool that estimates the strength of positive and negative sentiment in short texts in English. It has a sentiment lexicon and rules to perform sentiment analysis. The lexicon in SentiStrength provides positive and negative sentiment scores for various types of polarity words such as booster words, question words, emotion words, negation words, slang, idioms and emoticons. The score is represented as an integer from -5 to 5 where the large absolute value stands for the strong sentiment. SenticNet is another sentiment lexicon consisting of the sentiment scores for common sense concepts. The sentiment score is scaled from -1.0 to 1.0 to signify the polarity and intensity of the sentiment. The score in SenticNet is multiplied by 5 and rounded so that the sentiment scores of both SentiStrength and SenticNet are represented by an integer from -5 to 5. Finally, the polarity score of the word w, $po\_score(w)$, is defined as Equation (3.1), where $score_{SS}$ and $score_{SN}$ are the scores given by SentiStrength (SS) and SenticNet (SN). If the word is found in SentiStrength or SenticNet, the sentiment score in the lexicon is used as the $po\_score(w)$. If the word is found in both SentiStrength and SenticNet, the average of the sentiment score of both

lexicons is used as the $po\_score(w)$. Otherwise, the $po\_score(w)$ will be set to 0.

$$po\_score(w) = \begin{cases} \frac{1}{2}\left(score_{SS}(w) + score_{SN}(w)\right) & \text{if } w \in SS \text{ and } w \in SN \\ score_{SS}(w) \text{ or } score_{SN}(w) & \text{if } w \in SS \text{ or } w \in SN \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

The extended polarity score of w, $ex\_po\_score(w)$, is defined as Equation (3.2)

$$ex\_po\_score(w) = \begin{cases} po\_score(w) & \text{if } w \in SS \text{ or } w \in SN \\ \frac{1}{|C(w)|} \sum_{c \in C(w)} po\_score\_c(c) & \text{if } C(w) \text{ is derived by concept expansion} \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where $C(w)$ is a set of concepts expanded from the word w. The expanded concepts of the word are obtained by choosing the number of appropriate concepts from ConceptNet lexicon. Then, the average sentiment scores of the concepts are used as $ex\_po\_score(w)$. The detailed procedures of concept expansion will be explained in Section 3.3.

Finally, the sum of the sentiment scores of all positive or negative words in the tweet, sum_pos_score or sum_neg_score, is calculated as Equation (3.3) or (3.4)

$$sum\_pos\_score = \sum_{pos\_w} ex\_po\_score(pos\_w) \tag{3.3}$$

$$sum\_neg\_score = \sum_{neg\_w} ex\_po\_score(neg\_w) \tag{3.4}$$

where pos_w or neg_w is the word whose extended polarity score is positive or negative. We define six sentiment score features "*po-degree*", where *po* is either "positive" or "negative" and *degree* is one of "low", "medium" or "high". The feature "positive-low", "positive-medium" and "positive-high" is activated if $sum\_pos\_score = 0$, $0 < sum\_pos\_score \leq 2$ and $sum\_pos\_score > 2$, respectively. The range of the sentiment scores for each class is determined based on our intuition. The sentiment score features for negative polarity are defined similarly.

3.2.2 Sentiment contradiction feature

As previously explained, sarcasm normally occurs in a sentence that expresses the meaning opposite to the intended meaning. Therefore, we attempt to apply the sentiment

analysis to find contradiction in sentiment polarity among the words in the tweet. Although conflict of the polarity has been sometimes used in the analysis of sarcasm, an important characteristic of the proposed feature is coherence in the tweet. We assume that coherence identification plays a significant role because the sentiment contradiction found in incoherent sentences may not indicate sarcasm. Let us consider the following two example tweets.

T2: I am coughing and choking. I am feeling great.

T3: I am coughing and choking. Mary is still fine.

Since there are negative words ("coughing" and "choking") and positive words ("great" or "fine"), the sentiment contradiction is found in both T2 and T3. However, T2 is sarcasm, while T3 is not. In T2, two sentences are coherent or related to each other. The coherence is captured by the fact that pronoun "I" is repeated as the subjects of these sentences. T2 clearly shows contradiction in logical meaning by saying the opposite word of the intended word (coughing and choking $\neq$ great). Thus, T2 can be classified as sarcastic. On the other hand, in T3, there is no sign of logical connection between the words within two sentences. It is obvious that negative and positive words refer to different subjects "I" and "Mary", respectively. In such cases, the tweet can be regarded as non-sarcastic even when the sentiment contradiction is found.

The sentiment contradiction feature is represented either *contra* or *contra+coher*. The feature *contra* is activated if the following two conditions are satisfied: 1) the tweet consists of only one sentence and 2) contradiction in sentiment score is found as both *sum_pos_score* (defined in Equation (3.3)) and *sum_neg_score* (Equation (3.4)) are greater than 0. The feature *contra+coher* is activated if the following three conditions are fulfilled: 1) the tweet consists of two or more sentences, 2) contradiction in the sentiment score is found and 3) the tweet is classified as coherent. The procedures to identify coherence in the tweet will be explained in Section 3.4.

3.2.3 Punctuation and special symbol feature

Many studies have shown that punctuation plays an important role in the text communication as a sign of pausing, changing in the tone of voice or even to indicate the strong

feeling or exaggerate something. Punctuation has a lot of influence in text classification tasks, especially in the area of the sentiment analysis. Special symbols frequently used in Twitter may also be effective for the sarcasm recognition. Thus, punctuation and special symbols are considered as one of the main features in our research. The following 7 symbols or words are considered as punctuation and special symbol features:

1. Emoticons
2. Repetitive sequence of punctuation
3. Repetitive sequence of characters
4. Capitalized words
5. Slang or booster words[2]
6. Exclamation marks
7. Idioms[3]

The frequency of these elements in the tweet, denoted as fre, is classified into three classes: "low" ($fre = 0$), "medium" ($1 \leq fre \leq 3$) and "high" ($fre > 3$). The range of the frequency for each class is determined through our preliminary experiment. Our punctuation and special symbol features are represented as the pair of one of 7 elements and 3 frequency classes. That is, 21 features are introduced.

3.3 Concept expansion and pruning

3.3.1 Concept expansion

Concept level and common sense knowledge are indispensable to perceive, understand and acknowledge things, which are shared through the common knowledge or facts that can be reasonably realized. In this research, we focus on the semantic analysis of tweets using the semantic network consisting of concepts of words to obtain more affective information. As explained in Subsection 3.2.1, the concepts of the word are used for calculating the sum of the polarity score. Here we will explain a procedure to derive a set of the concepts, i.e. $C(w)$ in Equation (3.2). A concept lexicon called ConceptNet 5.0[4] is used to expand the concepts of the word whose sentiment score is unknown. The ConceptNet 5.0 is a seman-

[2]SentiStrength is used as a lexicon of slang and booster words.

[3]http://www.englishcurrent.com/idioms/esl-idioms-intermediate-advanced/

[4]http://conceptnet5.media.mit.edu/

tic network consisting of common sense knowledge and concepts, represented in the form of nodes (words or short phrases) and labeled edges (relationships) between them. For example, the sentence "A dog is an animal" is parsed into an assertion as "dog/IsA/animal". The assertion consists of two nodes ("dog" and "animal") and one edge ("IsA"). There are 31 types of relationships, such as "PartOf", "UsedFor", "MadeOf", etc. ConceptNet 5.0 contains over 800,000 assertions. These assertions are ranked based on the number of votes by users to ensure the quality and significance of each assertion. In our method, for each word w, the top five ranked concepts are set as $C(w)$.

Since the concepts in ConceptNet 5.0 can be represented as the phrases (such as "day where person stay home and relax"), the polarity score of the concept c ($po\_score\_c(c)$ in Equation (3.2)) is defined as the average of the polarity score of the words in the concept:

$$po\_score\_c(c) = \frac{1}{|C|} \sum_{w \in C} po\_score(w) \tag{3.5}$$

C stands for a set of sentiment words in c.
The concept-level lexicon improves the robustness of our system in terms of calculation of the sentiment scores of tweets. The lexicon also allows the system to recognize sarcasm of the sentence at the concept level.

3.3.2 Concept pruning

Although the concept expansion is effective to recognize the polarity of the unknown sentiment words, some irrelevant concepts can be obtained. They may cause errors on the sarcasm identification. For example, five concepts can be expanded from "holiday" in the tweet T4.

> **T4**: The typhoon is still blowing hard. What is a nice holiday!
>
> holiday => ["special day", "day where person stay home and relax", "~~special event celebrate by person~~", "special day that celebrate event", "day where person do not have to work"]

In this case, the concept of "special event celebrate by person" should not be expanded, since "holiday" in T4 means not an event but a day. We introduce a procedure called "concept pruning" to prevent from expanding such irrelevant concepts.

The concept pruning consists of three steps: 1) word sense disambiguation (WSD), 2) keyword extraction and 3) similarity measurement. In the first step, WSD is performed to find the actual meaning of the unknown sentiment word within the tweet. SenseLearner 2.0[5] is used to determine the WordNet sense of the word. SenseLearner is a statistical WSD system trained on SemCor corpus[6], which is a corpus annotated with WordNet senses. The method has participated in Senseval-3 English All Words task[7] and achieved an average accuracy of 64.6%, while the "most frequent sense" baseline of this task was 60.9% [86, 87]. In the second step, the disambiguated sense and one of the 5 highly ranked concepts of the word are represented as a set of keywords K_s and K_c, respectively. K_s is a set of the words in the gloss of the WordNet sense s, while K_c is a set of the words in the concept c. Only nouns, verbs, adjectives and adverbs are extracted as the keywords. In the final step, the similarity between K_s and K_c is measured by Equation (3.6).

$$sim(K_s, K_c) = \max_{w_s \in K_s, w_c \in K_c} sim\_word(w_s, w_c) \tag{3.6}$$

In this study, Skip-gram model proposed by Mikolov et al. [88] and Resnik's algorithm [89, 90] are used to compute the word similarity $sim\_word(w_s, w_c)$. If $sim(K_s, K_c)$ is greater than a threshold T_c, the concept c is kept, otherwise pruned.

Skip-gram model is a method to obtain vector representation of words from a large amount of raw texts. Word similarity can be measured by the cosine similarity between the vectors of the words. We use the Word2Vec tool in Python library distributed by Radim Rehurek in gensim[8] to compute the word similarity score. The tool contains 300-demensional vectors for 3 million words and phrases. Google News dataset[9] containing about 100 billion words was used as a text corpus to create vector representation of words [91].

In Resnik's algorithm, the similarity between two words are defined as the information content of the least common subsumer (most specific ancestor node) in WordNet *is-a* taxonomy for nouns. Given two words w_1 and w_2, the most informative subsumer of two words is the concept c that maximizes their semantic similarity. That is, the word

[5]http://lit.csci.unt.edu/~senselearner/

[6]http://www.cse.unt.edu/~rada/downloads.html#semcor

[7]http://www.senseval.org/senseval3

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

[9]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?pref=2&pli=1

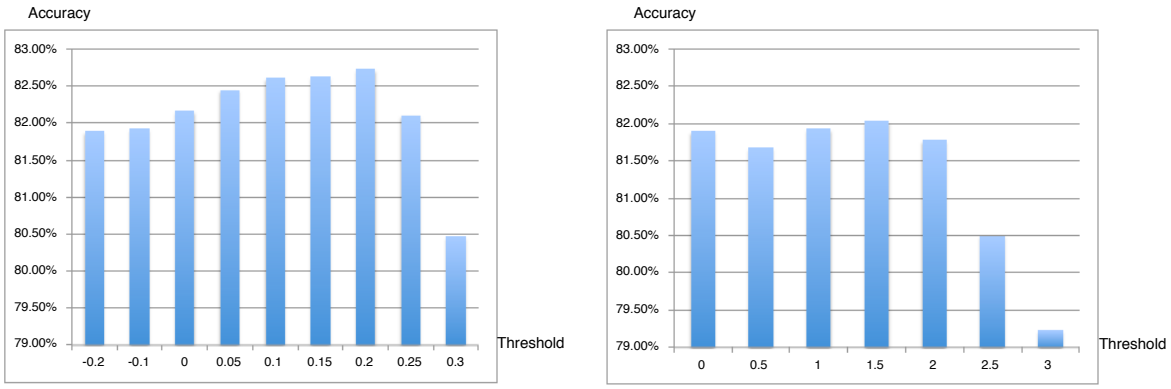

Figure 3.2: Optimization of the parameter T_c: Skip-gram model (left) and Resnik's algorithm (right)

similarity is defined as Equation (3.7).

$$sim(w_1, w_2) = max_{c \in subsumers(w_1, w_2)}[-logP(c)] \tag{3.7}$$

where $subsumers(w_1, w_2)$ is the set of WordNet synsets that are ancestors of both w_1 and w_2. When w_1 or w_2 has two or more senses, ancestors of any pairs of the senses of two words are included in $subsumers(w_1, w_2)$. Probability of the concept, $P(c)$, is the ratio of the number of nouns having a sense subsumed by the concept c to the total number of the observed nouns in a corpus.

The parameter T_c is optimized on a development data. We will explain the details of our experiment to evaluate our proposed methods later, but here we briefly introduce the result of the parameter optimization. Figure 3.2 shows accuracy of sarcasm identification for different threshold T_c. It indicates that Skip-gram model provides better accuracy than Resnik's algorithm, and the accuracy is the best when $T_c = 0.2$. From these results, we choose Skip-gram model as the word similarity measure and set T_c as 0.2.

3.4 Coherence identification

This section presents a new method to identify the coherence among the sentences in the tweet to improve the sentiment contradiction feature. When the sentiment contradiction feature is extracted from a tweet, coherence is another issue that we need to consider. As explained in Subsection 3.2.2, it is not always obvious to say that a tweet consisting of multiple sentences with sentiment contradiction is sarcastic. Therefore, we introduce

two new methods of coherence identification as follows: 1) Heuristic rules-based coherence identification and 2) Coherence Clustering with Feature Weight Optimization (CC-FWO).

3.4.1 Heuristic-based coherence identification

In this method, coherence between two sentences is identified by simply checking coreference between subjects or objects of the sentences. Let us suppose that sentence s_1 precedes s_2, and word w_1 and w_2 are the subject (or object) of s_1 and s_2, respectively. If w_1 and w_2 refer to the same object/thing or they have the same referent, we regard the two sentences as coherent. Note that in fact w_i (subject or object of the sentence) can be not a single word but a phrase. We created the following five rules to check coreference between w_1 and w_2:

1. Pronoun matching - w_1 and w_2 are identical pronouns, including reflexive pronouns, personal pronouns and possessive pronouns.

2. String matching - w_1 and w_2 are identical. Note that stopwords are ignored in string matching.

3. Definite noun phrase - w_2 starts with the word "the".

4. Demonstrative noun phrase - w_2 starts with the "this", "that", "these" and "those".

5. Both proper names - w_1 and w_2 are both named entities. Named entities are recognized by the Stanford Named Entity Recognizer [10].

Two sentences are regarded as coherent if they fulfill one of the above rules. If one pair of w_1 and w_2 satisfies our rules among all combination of w_1 and w_2 in multiple sentences in a tweet, we regard the overall tweet as coherent.

3.4.2 Coherence clustering with feature weight optimization (CC-FWO)

Another proposed method is based on unsupervised clustering. We call this method Coherence Clustering with Feature Weight Optimization (CC-FWO). The coherence identification in CC-FWO is to make clusters of coherent and incoherent tweets for a given set of the tweets. The overall procedure of the coherence identification is shown in Figure 3.3. In the training phase, the set of the input tweets are annotated with coherence

[10] http://nlp.stanford.edu/ner/

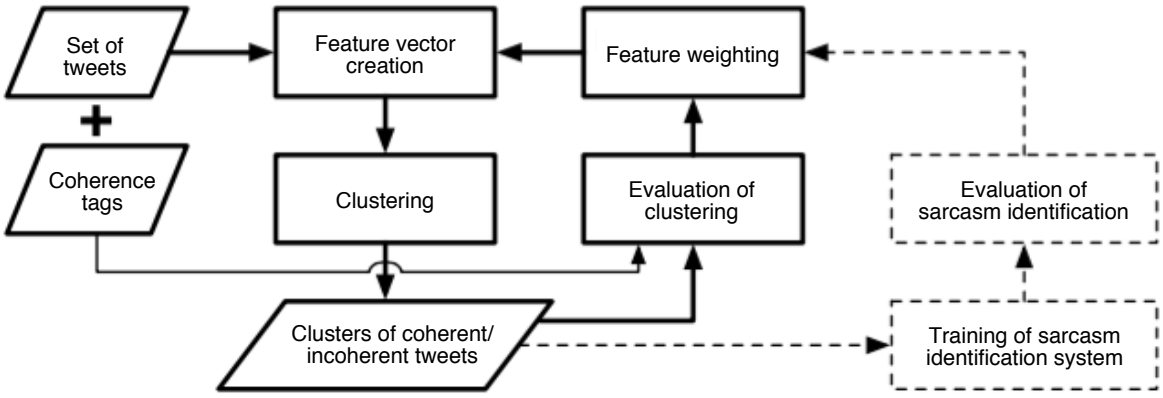

Figure 3.3: Procedures of coherence identification in tweets based on CC-FWO

tags to indicate if the tweet is coherent or not. We construct a manually annotated data consisting of 800 coherent tweets and 800 incoherent tweets.

For the clustering of coherent/incoherent tweets, each tweet is represented as a feature vector. Let us suppose that sentence s_1 precedes s_2 in the tweet, then the word w_1 or w_2 is one of the subject, noun or pronoun in s_1 or s_2. Then, the features used for the clustering are summarized as Table 3.1.

The detail procedure to derive the semantic class agreement feature (9th feature) is as follows.

1. For each i=1 and 2, the sense of w_i, called $sense_i$, is disambiguated by SenseLearner 2.0.

2. A set of hypernyms and hyponyms of $sense_i$ and $sense_i$ itself, called SHH_i, is created.

3. Similarity of all possible pairs of the synsets from SHH_1 and SHH_2 is measured by a method proposed by Resnik [89].

4. The feature is activated when the similarity of one of the synset pairs is greater than a threshold. It is set to 1.37 based on preliminary experiment.

The last two features in Table 3.1 are introduced because we often treat acronym, abbreviation and emoticon as a separate sentence. In other words, the isolated acronym, abbreviation or emoticon are always considered as coherent with other sentences. If there

[11]http://nlp.stanford.edu/projects/coref.shtml

Table 3.1: Features for clustering of coherent/incoherent tweets

Feature	**Description**
Pronoun feature 1	w_1 is reflexive, personal or possessive pronoun.
Pronoun feature 2	w_2 is reflexive, personal or possessive pronoun.
String match	w_1 and w_2 are identical.
Definite noun phrase	w_2 starts with the determiner "the".
Demonstrative noun phrase	w_2 starts with "this", "that", "these" or "those".
Both proper names	Both s_1 and s_2 contain named entities. Proper names are recognized by the Stanford Named Entity Recognizer (NER).
Coreference resolution	s_1 and s_2 are judged as coreferred by Stanford Deterministic Coreference Resolution System[11].
Number agreement	wc_1 and wc_2 agree in number (i.e. they are both singular or plural), where wc_1 and wc_2 are judged as coreferred by the Stanford Deterministic Coreference Resolution System.
Semantic class agreement	w_1 and w_2 are semantically similar.
Acronym or abbreviation	A tweet contains an acronym or abbreviation (i.e., "lol", "ynwa").
Emoticon	A tweet contains an emoticon (i.e. "☺", ":-)").

are three or more sentences, the feature vector is constructed as follows. The feature vectors of all pairs of the sentences, denoted as $\vec{c}_{ij}$, are created. The value of each dimension is defined as 0 if the values at the same dimension of all $\vec{c}_{ij}$ are 0, otherwise 1.

After extraction of the feature vectors, unsupervised clustering is performed. We tried three representative clustering algorithms: K-means, EM (expectation maximization) algorithm [92] and Hierarchical clustering. Note that the number of clusters (N_c) should be predefined in these algorithms. N_c is optimized on the training data.

Table 3.2: Accuracy of coherence identification

N_c	2	4	6	8	10	12	14	16	18	20
(1)	.7711	.7748	.7770	.7795	.7792	.7790	**.7816**	.7807	.7801	.7769
(2)	.8130	.8151	.8172	**.8202**	.8173	.8135	.8144	.8159	.8133	.8102
(3)	.7322	.7326	.7349	.7363	**.7387**	.7372	.7343	.7344	.7329	.7300

Note: (1) = K-mean, (2) = EM algorithm, (3) = Hierarchical clustering

Optimization of the feature weights

The remaining problem is how to define the weights in the feature vectors. Usually, the weights are determined as binary; the weight of the present feature is 1, while the absent feature is 0. In our method, however, the weights of the present and absent features are optimized by brute force search. The weight of each feature is changed from -1 to 1 by a step of 0.2. All combination of the weights are evaluated in terms of the performance of the clustering. The performance of the clustering is evaluated by the accuracy defined as the ratio of the agreement between the gold and predicted coherence tags. The coherence tag of each tweet is predicted as follows: each cluster is judged if it is a cluster of the coherent or incoherent tweets by voting the coherent tags of the tweets in the cluster, then all the tweets within the coherent or incoherent clusters are regarded as coherent or incoherent. An optimal set of the feature weights is chosen so that the accuracy of the clustering becomes the highest.

In the test phase, the tweets in both the training and test data are converted to the feature vectors with the optimized weights, then the unsupervised clustering is performed. The cluster labels are used as the coherent feature for sarcasm identification. That is, *contra+coher* described in Subsection 3.2.2 is a set of N_c features represented as *contra+*cl_i, where cl_i stands for the i-th cluster that the tweet belongs to.

Table 3.2 shows the accuracy of the coherence identification on the training data for three clustering algorithms and different N_c (the number of clusters). As shown in Table 3.2, EM algorithm outperformed the others. Furthermore, the best accuracy was obtained when N_c was set to 8.

The advantage of CC-FWO is that the clustering of the tweets is performed in unsuper-

Table 3.3: Comparison of different coherent identification methods in terms of the accuracy of sarcasm identification

N_c	2	4	6	8	10	12	14	16	18	20
(1)	.8186	.8227	.8264	.8271	.8285	.8286	**.8317**	.8270	.8236	.8208
(2)	.8251	.8283	.8302	**.8378**	.8320	.8293	.8303	.8281	.8279	.8236
(3)	.8177	.8191	.8211	.8227	**.8239**	.8218	.8207	.8198	.8172	.8122

Note: (1) = K-mean, (2) = EM algorithm, (3) = Hierarchical clustering

vised manner. Note that the coherence tags of the tweets are used only for determination of feature weights, optimization of the number of clusters N_c and selection of the clustering algorithm (K-means, EM or Hierarchical). Even when the training data is not annotated with the coherence tags, we can run CC-FWO as follows. For each set of the features, our sarcasm identification system is trained with the obtained coherent/incoherent clusters, then the accuracy of sarcasm identification on a development data is measured. An optimal feature weights will be selected so that the performance of the sarcasm recognition is maximized. This procedure is shown by the dotted lines in Figure 3.3. Similarly, the optimization of N_c and selection of the clustering algorithm are also possible.

Table 3.3 shows the result of a preliminary experiment on sarcasm identification when CC-FWO is run without coherence annotated data. We retrieved 50,000 tweets for our development dataset. 25,000 tweets were randomly selected as normal tweets, whereas the other 25,000 tweets are sarcastic tweets[12]. Similarly to the method of coherence identification, three clustering algorithms and different N_c (the number of the clusters) were used for the evaluation. As shown in Table 3.3, it is also found that EM algorithm outperformed the others and the optimized N_c was also 8.

Therefore, although the computational cost becomes much greater, manual annotation of the coherence to the training data is not mandatory to run CC-FWO. In that sense, CC-FWO can be regarded as a kind of semi-supervised method or distant supervision [93]: CC-FWO requires a collection of the tweets with sarcasm annotation, but not with

[12]This data is exactly the same as ARTK-50K which will be described in Subsection 3.6.1. Note that CC-FWO will be evaluated by Experiment II in Subsection 3.6.2, and ARTK-50K is mutually exclusive with a test data of this experiment.

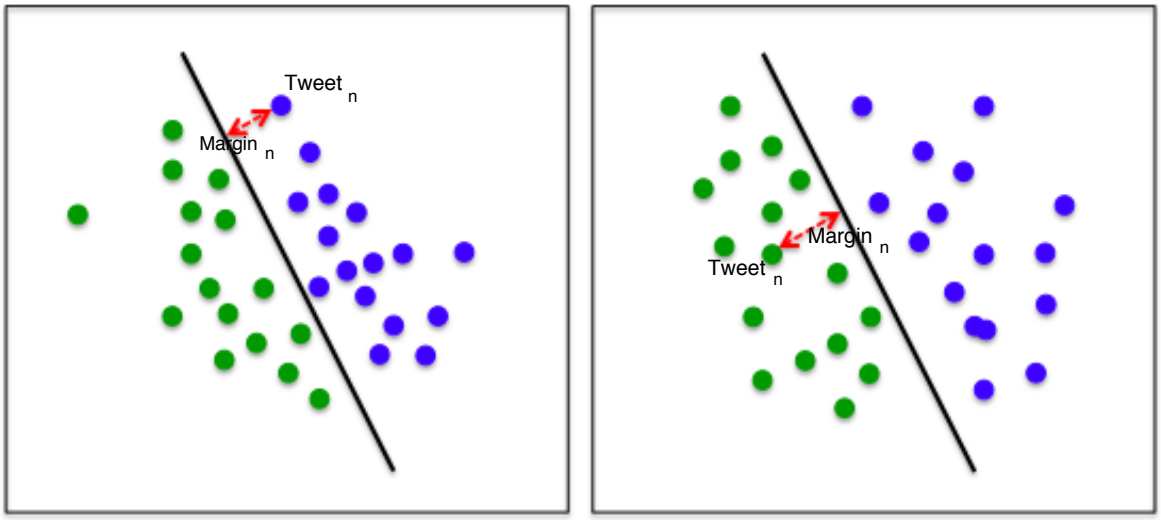

Figure 3.4: Example of conflicts of two SVM classifiers

coherence annotation.

3.5 Classification procedures

The classifiers for sarcasm recognition are trained by supervised learning. In this study, SVM is chosen as learning algorithm due to its simplicity and effectiveness in binary classification.

Table 3.4 shows a summary of our proposed features described from Section 3.2 to 3.4. To combine these features with N-gram feature, we choose an approach in which two feature sets are used separately to train two different SVMs and the final result is chosen from the results of these SVMs.. First, we perform the classification task twice (once with n-grams and once with our features) and obtain two sets of results. Then, we determine the final result by comparing the classification outputs. For each tweet, if the judgments of two SVMs agree, it simply becomes the final result. If they do not agree, we consider the classification margin for each classifier. Figure 3.4 demonstrates a situation where two classifiers obtain different classification results for the same tweet. In this case,

Table 3.4: Summary of features

N-gram feature	
uni-gram	a single word in a tweet.
bi-gram	a sequence of two words in a tweet.
tri-gram	a sequence of three words in a tweet.
Sentiment contradiction feature	
contra	1. the tweet consists of one sentence. 2. the contradiction of the sentiment score is found by the method described in Subsection 3.2.2.
contra + coher	1. the tweet consists of two or more sentences. 2. the contradiction of polarity is detected. 3. the tweet is judged as coherent by the method described in Section 3.4
Sentiment score feature	
pos_low	$sum\_pos\_score \leq -1$
pos_medium	$0 \leq sum\_pos\_score \leq 1$
pos_high	$sum\_pos\_score \geq 2$
neg_low	$sum\_neg\_score \leq -1$
neg_medium	$0 \leq sum\_neg\_score \leq 1$
neg_high	$sum\_neg\_score \geq 2$
Punctuation and special symbols feature	
emoticons *repetitive_sequence_of_punctuations* *repetitive_sequence_of_characters* *capitalized_word* *slang_and_booster_words* *exclamation_marks* *idioms*	*low*: activated if $number = 0$ *medium*: activated if $1 \leq number \leq 3$ *high*: activated if $number \geq 4$

we compare the margin (distance between the data and separating hyperplane) of both classifiers. Usually, the higher the margin is, the more reliable the output is. Therefore,

we take the output from the classifier with higher margin as the final result. In the case of Figure 3.4, the result of right classifier is chosen.

In the experiment that will be described in Section 3.6, we have also trained a single classifier with both N-gram and our proposed features. However, its performance was worse than ensemble of two classifiers.

Figure 3.5 summarizes the overall process of our proposed method described in Sections 3.1 to 3.5. In this research, we propose a new supervised learning method that utilize several major modules, including 1) concept expansion and pruning, 2) polarity identification of words, 3) coherence identification and 4) classification by SVMs. Sentiment scores of words are used to extract the features for the classification. We also use the common sense concept to find the sentiment score for unknown words in the sentiment lexicons. Then, we consider coherence in a tweet to ensure that the tweets with contradiction in the sentiment score have relationships across multiple sentences. Finally, we construct the feature vector to train an SVM classifier based on our proposed features. N-gram and our proposed features are used to train separate classifiers, then a more reliable judgment between them is chosen as the final result.

3.6 Evaluation

In this section, we describe how the experiments were conducted to evaluate the performance of our method. As described in Section 3.4, two methods for coherence identification are proposed. To evaluate the effectiveness of these two methods separately, two experiments were conducted. In the Experiment I, the system with heuristic-based coherence identification described in Subsection 3.4.1 was evaluated. While in the Experiment II, the system with coherence clustering described in Subsection 3.4.2 was evaluated.

3.6.1 Experiment I

Data

In the first experiment, we retrieved 50,000 tweets from Twitter for our datasets. 25,000 tweets were obtained as sarcastic tweets, whereas the other 25,000 tweets were obtained as normal (not sarcastic) tweets. To collect the sarcastic tweets, the hashtag "#sarcasm"

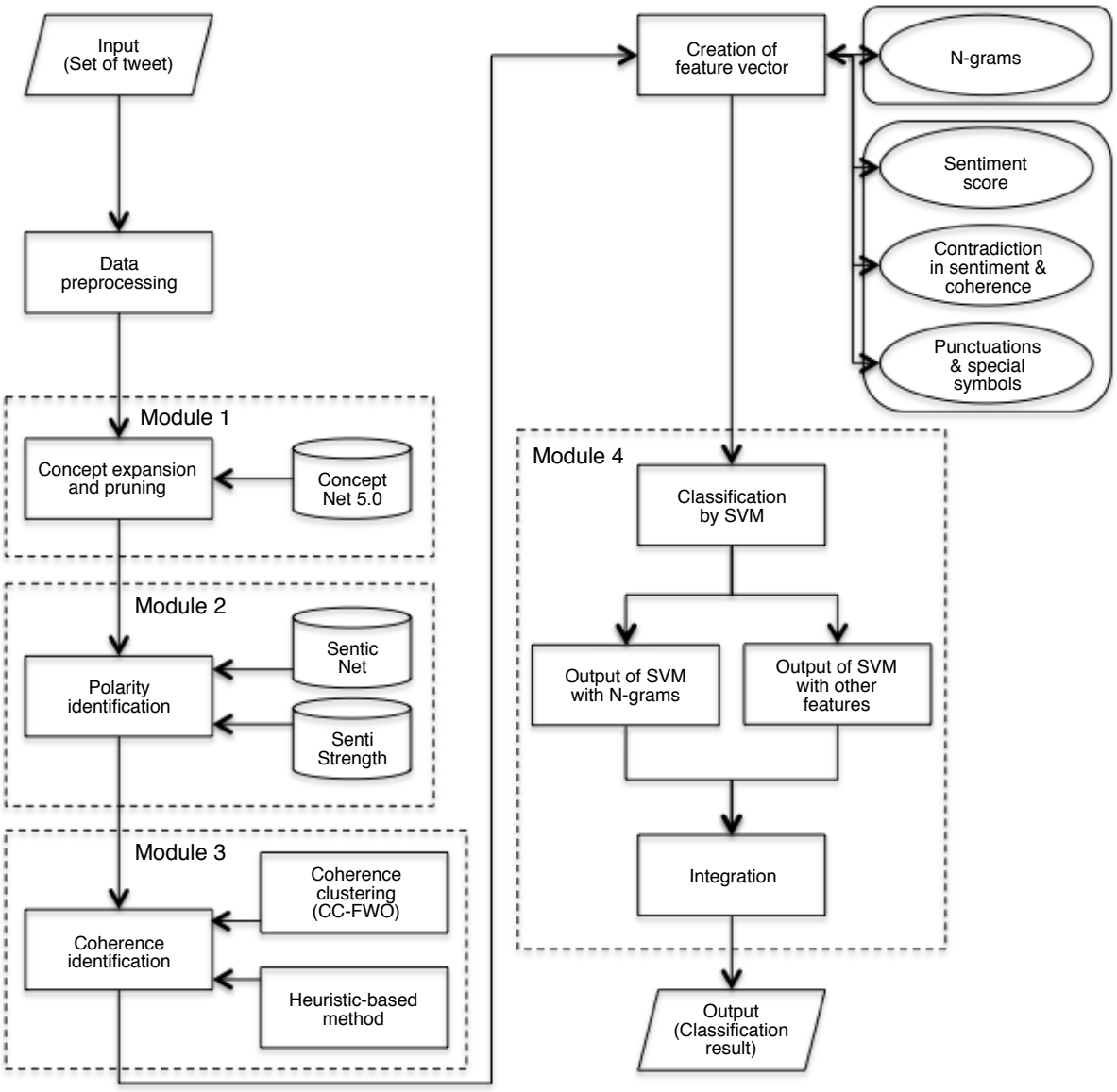

Figure 3.5: Flowchart of overall process of our sarcasm recognition method

was used as a query keyword. While the normal tweets were retrieved by searching with the keywords randomly selected from WordNet. Obviously, the sarcastic tweets can be posted without "#sarcasm" hashtag. We checked randomly sampled 300 normal tweets and found only 3.7% were sarcastic. Therefore, we ignored such noisy tweets in the experiment. Twitter4J[13] was used as a tool to prepare a collection of tweet data. We call this data ARTK-50K. ARTK means Automatically Retrieved Tweets using Keywords, while 50K means the data size.

[13]http://twitter4j.org/en/index.html

Table 3.5: Results of sarcasm identification based on sentiment contradiction

Methods	A	R	P	F
Baseline 1 (sentiment contradiction)	0.5714	0.5537	0.5683	0.5609

Table 3.6: Results of sarcasm identification of single classifier

Methods	A	R	P	F
Our proposed features	0.6417	0.6453	0.6479	0.6466
Baseline 2 (uni-gram, bi-gram and tri-gram)	0.7751	0.7748	0.7792	0.7769

Task

The task of this experiment was to identify a sarcasm class (sarcasm or not) for a given tweet. The tweets were classified based on variety of features, including N-grams and our proposed features. The results of the proposed method were compared against two baseline methods. Baseline 1 was created based on the definition of sarcasm. Sarcasm usually occurs in a sentence that expresses the meaning opposite to the intended meaning. Therefore, the tweets that contained both positive and negative word ($sum\_pos\_score > 0$ in Equation (3.3) and $sum\_neg\_score > 0$ in Equation (3.4)) were regarded as the sarcastic tweets. Baseline 2 was the SVM trained with only N-gram (uni-gram, bi-gram and tri-gram) features. Since N-gram is a common and well-known feature for sarcasm identification task, it is considered as a strong baseline. In this experiment, our proposed methods as well as Baseline 2 were evaluated by 10-fold cross validation on ARTK-50K dataset, while Baseline 1 was simply applied to the overall ARTK-50K. LIBLINEAR[14] was used for training the classifiers in both Baseline 2 and our proposed method. The parameter C in LIBLINEAR was optimized by cross validation of the training data. Recall, precision, F-measure and accuracy are measured to evaluate the performance of sarcasm identification.

Table 3.7: Results of the proposed method and effectiveness of individual features

Methods	A	R	P	F
Uni-gram, bi-gram, tri-gram & all our proposed features	0.8083	0.8173	0.8086	0.8129
− Sentiment contradiction	0.7951	0.7948	0.7997	0.7973
− Sentiment score	0.7842	0.7843	0.7860	0.7852
− Punctuations & special symbols	0.8035	0.8032	0.8041	0.8037
− (Heuristic rules-based coherence identification)	0.7827	0.7833	0.7859	0.7845
− Concept expansion & pruning	0.7809	0.7814	0.7852	0.7833

Results

Table 3.5 shows the results of Baseline 1, where accuracy, recall, precision and F-measure are denoted as A, R, P and F respectively. The performance is relatively high, although Baseline 1 does not rely on supervised machine learning, but on the sentiment lexicon only. Table 3.6 reveals results of single SVM with our proposed features (sentiment score, sentiment contradiction and punctuation & special symbol features) and Baseline 2. SVM with our proposed feature performed better than Baseline 1 but worse than Baseline 2. We found that N-gram features were still powerful for classification of sarcasm. Table 3.7 shows the results of the combination of two SVMs[15]. In this table, the classifiers without one type of the feature were compared with the system with all features. The sixth row in Table 3.7 is the system where coherence in a tweet is not considered[16], while the seventh row indicates the system where ConceptNet is not used for concept expansion. We can find that the combination of N-gram features and all our proposed features improves the accuracy 3% against Baseline 2 with N-grams. It indicates that several sarcastic tweets can be found by our approach but not by N-gram features. Examples of such sarcastic tweets are shown below, where polarity words are in bold:

[14]https://www.csie.ntu.edu.tw/~cjlin/liblinear/

[15]A single SVM classifier using both N-gram and proposed features was also evaluated. The accuracy was 0.7846, which was was worse than the voting of two classifiers.

[16]$contra + coher$ feature is activated even when coherence in a tweet is not confirmed.

Table 3.8: Results of McNemar's test between Baseline 1 or Baseline 2 and our proposed method on ARTK-50K dataset

Pair	**Two-tailed P value**
1. Baseline 1 - Our proposed method	0.0001
2. Baseline 2 - Our proposed method	0.0001

1. I am **thrilling.** The **storm** in my area
2. A **nice** sunny day to go **pay** some **bills**.....
3. It's **brilliant** to realize when your **best** asset **screw** everything up
4. I really **enjoy** running on the treadmill. So **exhausted**!!
5. It has been **freezing** and **snowing** all week. The weather is so **gorgeous**

Although the polarity words in these tweets are effective features, they do not frequently appear in the training data. SVM trained with N-gram features fails to classify them as sarcastic due to data sparseness. Our sentiment score, sentiment contradiction and punctuation & special symbol features are rather abstract and appear many times in the training data. Therefore, our method can classify these sarcastic tweets correctly.

In addition, we also investigated the significance between our proposed method and Baseline 1 or Baseline 2. Table 3.8 shows the results of McNemar's test. It indicates that our method is significantly better than both Baseline 1 and Baseline 2 with 99% confidence interval.

Contribution of our proposed features

In this part, we further discuss the contribution of each proposed feature.

- Punctuations and special symbols

  As seen in Table 3.7, punctuations and special symbols contribute only a slight improvement. The accuracy is decreased by only 0.48% when they are removed from the system. This may be because punctuations and special symbols are also incorporated in uni-gram feature set, that is, our proposed feature is partially duplicated with uni-gram. Nevertheless, the feature provides some improvement to the overall result.

- Concept expansion and pruning

The results show that concept level knowledge expansion can enhance the quality of the sentiment score features from 78.09% to 80.83%. Tweets are unstructured and context free data. There are a lot of unknown words and slang that are very difficult to handle. From this reason, concept level and common sense knowledge can contribute to improve our method.

- Effectiveness of coherent identification

  As explained in Subsection 3.2.2, coherence in the tweet is considered to detect contradiction of polarity more precisely. Next we will discuss the contribution of coherence feature. The accuracy is decreased by 2.56% (from 80.83% to 78.27%) when coherence is ignored as shown in Table 3.7. It is clear that contradiction in the sentiment score with coherence feature has an impact on the improvement of the result. Let us consider a non-sarcastic tweet in our dataset "My gf's mac failed three times and I had to reboot twice. Windows are WAY simpler." Suppose that we ignore coherence when the feature vector is constructed. This tweet would be misclassified as a sarcastic tweet since it contains contradiction in the sentiment score of both positive ("simpler") and negative ("fail") words in two different sentences. However, when coherence in the tweet is checked, our method recognizes that the words "My gf's mac", "I" and "Windows" are not related to each other. In other words, two sentences in this tweet are incoherent. Now it can be correctly classified as a non-sarcastic tweet. As shown in this example, contradiction of polarity in an incoherent tweet does not usually indicate sarcasm.

Contribution of the concept expansion

The contribution of the concept expansion and pruning was evaluated in further detail. Table 3.9 shows the results of three systems: no concept is expanded, the concepts are expanded but not pruned (the 5 most related concepts are always expanded), and only the related concepts are obtained by concept expansion and pruning. Table 3.10 indicates the total number of expanded concepts in ARTK-50K.

It is found that both concept expansion and pruning could improve all criteria. 1.81 concepts per tweet were obtained in the ARTK-50K dataset. Furthermore, the concept pruning reduced the number of expanded concepts by 30% and contributed to an additional improvement. Thus, our pruning method successfully removed the irrelevant

Table 3.9: Effectiveness of concept expansion and pruning

Method	ARTK-50K			
	A	R	P	F
(1)	0.7809	0.7814	0.7852	0.7833
(2)	0.8015	0.8002	0.8071	0.8035
(3)	0.8083	0.8173	0.8086	0.8129

Note: (1) = no concept expansion, (2) = concept expansion only, (3) = concept expansion and pruning

Table 3.10: Number of expanded concepts

Method	ARTK-50K
no concept expansion	-
concept expansion only	90,629
concept expansion and pruning	63,014

concepts.

Limitation of our approaches

There are some limitations in this method. The most important problem is that our coherence identification method is too simple. We have provided some heuristic rules to determine coherent relationship among multiple sentences. Coherence may have a lot of influence in the classification, however, the improvement by coherence identification was not so great in our experiment. We should investigate a better way to identify and incorporate the coherence feature in our method. That is the reason why we propose more sophisticated clustering based method, CC-FWO.

3.6.2 Experiment II

Data

In the second experiment, two datasets were used: 1) ARTK-300K (Automatically retrieved tweets using keywords) dataset and 2) SemEval-2015 Task 11 dataset. The ARTK-

300K dataset consists of 300,000 tweets. 150,000 tweets were prepared as sarcastic tweets, whereas the other 150,000 tweets were prepared as normal (not sarcastic) tweets. The procedure of data collection for both sarcastic and normal tweets is the same as ARTK-50K explained in Subsection 3.6.1. Note that ARTK-300K is same as ARTK-50K with respect to the way of construction, but its size is six times greater. The development data consisting of 30,000 tweets, which was used for the parameter optimization of T_c, was constructed in the same way. In this experiment, we also used the SemEval-2015 Task 11 dataset that was constructed for the evaluation of sentiment analysis of figurative language in Twitter. It is a collection of the tweets annotated with their sentiment score between -5 to 5. The dataset contained hashtags indicating the figurative language such as #sarcasm, #irony, #metaphor and so on. The tweets with #sarcasm and #irony were regarded as the sarcastic tweets, otherwise non-sarcastic. Note that #irony tweets were categorized as sarcastic, since we found that the difference between sarcasm and irony was very subtle and it was rather hard even for human to distinguish them. The training set contained 8,000 tweets, while the test set contained 4,000 tweets. 35% of the tweets were sarcastic in this dataset.

Task

The task of this experiment was also to identify a sarcasm class (sarcasm or not) for a given tweet. The tweets were classified based on variety of features, including N-grams and our proposed features with coherence clustering. The proposed systems as well as baselines were trained and tested by 5-fold cross validation on the ARTK-300K dataset. On the SemEval-2015 task 11 dataset, the classifiers were trained from the training data and evaluated on the test data.

Three baselines were compared with our proposed methods. Baseline 1 was created based on the definition of sarcasm. Baseline 2 used only N-gram (uni-gram, bi-gram and tri-gram) features to train an SVM classifier for sarcasm identification. Both Baseline 1 and Baseline 2 were created using the same method as described in Subsection 3.6.1. In addition, we also created another Baseline 3 based on the method proposed by Riloff et al. [36][17]. LIBLINEAR was used for training the classifiers in Baseline 2 and our proposed method. The parameter C in LIBLINEAR was optimized by cross validation

[17]We implemented their system by ourselves. It was almost same as the original method, since we

Table 3.11: Results of sarcasm identification

Method	ARTK-300K				SemEval			
	A	R	P	F	A	R	P	F
Baseline 1	.5847	.5466	.6276	.5843	.5672	.5454	.5947	.5690
Baseline 2	.7628	.7443	.7639	.7540	.7413	.7150	.7363	.7255
Baseline 3 [36]	.7901	.7706	.8322	.8002	**.7665**	**.7622**	.8032	**.7821**
Proposed features	.6377	.7132	.6147	.6603	.6122	.6709	.6188	.6438
N-gram & proposed features	**.8320**	**.7816**	**.8594**	**.8187**	.7648	.7296	**.8172**	.7709

of the training data. Evaluation criteria were accuracy of sarcasm classification as well as recall, precision and F-measure in terms of retrieval of the sarcastic tweets. In the following discussion, the accuracy was mainly considered to compare the methods.

Results

Table 3.11 reveals the accuracy (A), recall (R), precision (P) and F-measure (F) of several methods on the ARTK-300K and SemEval dataset. Bold font indicates the best result among the compared systems. Baseline 1 achieved 0.58 and 0.57 accuracy on two datasets. Interestingly, the performance of Baseline 1 was acceptable, although the method did not rely on machine learning, but only on the contradiction of the sentiment polarity identified by the sentiment lexicon. The accuracy of Baseline 2 and Baseline 3 were better than Baseline 1. It indicates that the machine learning approach is appropriate for the identification of sarcasm. Baseline 3 was the best among three baselines in terms of all criteria.

The last two rows in Table 3.11 show the results of our proposed methods. The accuracy of the SVM trained with only our proposed features was 0.64 and 0.61 on the ARTK-300K and SemEval dataset respectively, which were approximately 13% lower than Baseline 2. This fact indicates that N-gram features are informative in the sarcasm identification task. The method of voting the SVM classifiers with N-gram and our proposed

deliberately followed the detail algorithm presented in Riloff et al. [36]

Table 3.12: The average length and percentage of single and multiple sentences of tweets in ARTK and SemEval dataset

Types of tweets	ARTK		SemEval	
	Avg. length	Proportion	Avg. length	Proportion
Single sentence	12.24 words	66%	14.21 words	54%
Multiple sentences	16.18 words	34%	16.72 words	46%

Table 3.13: Accuracy of sarcasm identification on different length of tweet data

	ARTK dataset				SemEval dataset			
	A	R	P	F	A	R	P	F
Single sentence								
Baseline 3 [36]	.8206	.8189	.8416	.8301	.7938	.7913	.8010	.7961
N-gram & our proposed features	.8096	.8082	.8203	.8142	.7463	.7254	.7726	.7482
Multiple sentences								
Baseline 3 [36]	.7357	.7291	.7656	.7469	.7101	.7066	.7379	.7219
N-gram & our proposed features	.8447	.8485	.8533	.8509	.7920	.7905	.8106	.8003

features achieved the best performance in ARTK-300K dataset[18]. It outperformed Baseline 3 by 4% accuracy and 1.9% F-measure. On the other hand, in SemEval dataset, our method achieved higher precision but lower recall than Baseline 3. F-measure and accuracy of Baseline 3 and our method were comparable.

To compare our method and Baseline 3 in further detail, we divided each dataset into two subsets: a set of the tweets consisting of a single sentence and multiple sentences. Table 3.12 shows the average length and the proportion of these subsets, while Table 3.13 compares the performance of two methods in each subset. It is found that our method works well for the multiple sentence tweets but Riloff's method (Baseline 3) does not. Our

[18]A single SVM classifier using both N-gram and proposed features was also evaluated. Its performance was worse than the voting of two classifiers. The accuracy of it was 0.7856 and 0.7193 on ARTK-300K and SemEval dataset, respectively.

proposed method achieved 0.84 and 0.79 accuracy for the multiple sentence tweets, which were approximately 11% and 8% higher than Baseline 3 on ARTK and SemEval dataset respectively. Let us consider the sarcastic tweet "I had a fever last night, still coughing as if Im choking. Soon I wont be able to eat either. Things are going well." Note that a positive word "well" and three negative words "fever", "coughing" and "choking" appear far from each other. Since Riloff's method checks the existence of a positive sentiment phrase and a negative situation phrase within five-words window, it fails to find sentiment contradiction in this example. However, our method can classify it as sarcastic correctly. On the other hand, our method performed worse than Riloff's method for the single sentence tweets. One of the reasons is that our method sometimes wrongly identifies the sentiment contradiction in a long single sentence. Let us consider the non-sarcastic tweet "I'm feeling so irritable right now & I just want to go home & not speak to anyone & take a rest." It contains one positive word "rest" and one negative word "irritable". Since our method simply check the existence of the positive and negative words to identify the sentiment contradiction, it misclassified this tweet as sarcastic. Note that coherence is not considered for the single sentence tweets in our method. On the other hand, since Riloff's method strictly checks the positive sentiment phrases and negative situation phrases, it can successfully judge it as non-sarcastic. When the sentence is long, our system causes such misinterpretation of the sentiment contradiction more. In fact, our method works better for the single sentence tweets on ARTK dataset than SemEval dataset, since the average length of the single sentence tweets on ARTK dataset is shorter than SemEval dataset. Furthermore, the reason why our method is better than Riloff's method on ARTK dataset but comparable on SemEval dataset is that the single sentence tweets are longer in SemEval dataset.

Comparing the features used in Riloff's and our methods, N-gram and sentiment contradiction features are commonly used, although the way to drive these features is different. On the other hand, sentiment score and punctuation & special symbol features are only used in our method. They are also widely used in many previous studies [38, 39, 37, 35].

Table 3.14: Effectiveness of individual features

Method	ARTK-300K				SemEval			
	A	R	P	F	A	R	P	F
N-gram & proposed features	.8320	.7816	.8594	.8187	.7648	.7296	.8172	.7709
− Sentiment score	.8119	.7673	.8366	.8004	.7431	.7231	.7960	.7578
− Sentiment contra.	.8266	.7618	.8521	8044	.7519	.7196	.7855	.7511
(− Coherence identification based on CC-FWO)	.8037	.7493	.8244	.7851	.7394	.7070	.7888	.7456
− Punctuation & special symbol	.8303	.7727	.8561	.8123	.7588	.7249	.8124	.7662

Contribution of the features

To evaluate the effectiveness of our proposed features, the classifiers without one type of the features were compared with the system with all features. Table 3.14 shows the results of this experiment. The fifth row (− Coherence identification based on CC-FWO) means that the system does not take care of the coherence identification in tweets based on CC-FWO. That is, the sentiment contradiction feature is always activated if there are positive and negative words.

Among the proposed features, the contribution of the sentiment contradiction feature considering coherence of the tweet was the best. This feature may capture linguistic aspects of sarcasm. Note that the system using the sentiment contradiction feature without considering the coherence (fifth row in Table 3.14) was worse than the system not using the sentiment contradiction feature (fourth row). It strongly suggests that the identification of the coherence is important for sarcasm identification.

The contribution of the sentiment score feature was also remarkable. Normally, sarcasm contains some special elements, which can create violation and aggressiveness in the communication, especially for the negation terms. Therefore, the strength of the sentiment polarity can be used to indicate the level of violation and aggressiveness in order to identify sarcasm in the tweet.

Table 3.15: Effectiveness of concept expansion and pruning

Method	ARTK-300K				SemEval			
	A	R	P	F	A	R	P	F
(1)	.8049	.7538	.8457	.7971	.7491	.7146	.7813	.7465
(2)	.8237	.7732	.8409	.8056	.7502	.7356	.7926	.7631
(3)	.8320	.7816	.8594	.8187	.7648	.7296	.8172	.7709

Note: (1) = no concept expansion, (2) = concept expansion only,
(3) = concept expansion and pruning

Table 3.16: Number of expanded concepts

Method	ARTK-300K	SemEval
no concept expansion	-	-
concept expansion only	576,881	11,374
concept expansion and pruning	385,602	7,722

On the other hand, the punctuation & special symbol seem not so effective, since only 0.17% and 0.60% drop of the accuracy were found by removing this feature on two datasets. Many previous studies have shown that sarcasm often contains strong emotional expressions in language. Emoticons and heavy punctuation can be used as the indicator of strong emotional expressions. Let us consider an example of angry expression "GO!!!!!". This example shows that repetitive punctuations can be used as a sign of yelling, which represents a violent emotional expression. Although the punctuation & special symbol feature can capture the strong emotion of the user, the emotional tweets do not always express sarcasm. Nevertheless, the feature can contribute to gain small improvement on the performance.

Contribution of the concept expansion

The contribution of the concept expansion and pruning was evaluated. Table 3.15 shows the results of three systems: no concept is expanded, the concepts are expanded but not pruned (the 5 most related concepts are always expanded), and only the related concepts

Table 3.17: Effectiveness of feature weights by CC-FWO

Method	ARTK-300K				SemEval			
	A	R	P	F	A	R	P	F
(1)	0.8161	0.7459	0.8431	0.7916	0.7519	0.7198	0.7900	0.7533
(2)	0.8320	0.7816	0.8594	0.8187	0.7648	0.7296	0.8172	0.7709

Note: (1) = without feature weighting by CC-FWO, (2) = with feature weighting by CC-FWO

are obtained by concept expansion and pruning. Table 3.16 indicates the total number of expanded concepts in the test set.

The concept expansion has contributed to improve almost all evaluation criteria on two datasets by maximum of 2.10%. 1.9 and 0.75 concepts per tweet were obtained in the ARTK-300K and SemEval dataset, respectively. Furthermore, the concept pruning reduced the number of expanded concepts by 33 or 32% and contributed to an additional improvement. Thus, our pruning method successfully removed the irrelevant concepts.

Contribution of coherence clustering with feature weight optimization (CC-FWO)

The optimization of the feature weights in CC-FWO has also taken part in the method to enhance the accuracy. Table 3.17 shows the results of our system when the weights of the feature vector is determined by CC-FWO or not. When CC-FWO is not applied, all feature weights were represented as binary. By CC-FWO, the accuracy was increased by 1.59% and 1.29% for the ARTK-300K and SemEval dataset, respectively. This result shows that CC-FWO plays a significant role in predicting the optimized weight for each feature in the clustering of the coherent/incoherent tweets. In our experiment, it was found that less important features were *proper names* and *demonstrative noun phrase* features, while significant features were *semantic class agreement* and *definite noun phrase* features.

Table 3.18: Results of McNemar's test between Baseline 1, Baseline 2 or Baseline 3 and our proposed method on ARTK-300K and SemEval dataset

Pair	**Two-tailed P value**	
	ARTK-300K	SemEval
1. Baseline 1 - Our proposed method	0.0001	0.0001
2. Baseline 2 - Our proposed method	0.0001	0.0018
3. Baseline 3 - Our proposed method	0.0001	0.0452

Statistical test

To investigate the significance of the our proposed system, we verify the difference between Baseline 1, Baseline 2 or Baseline 3 and our proposed method on both ARTK-300K and SemEval 2015 Task 11 dataset by McNemar's test. Table 3.18 shows two-tailed P value for comparison of the proposed method and each of three baselines. In ARTK-300K dataset, the results clearly show that our method significantly outperformed all the baselines with 99% confidence interval. In SemEval dataset, since the two-tailed P values of Baseline 1 and 2 are less than 0.01, our method significantly outperformed them with 99% confidence interval. On the other hand, Baseline 3, which was one of the state-of-the-art system, was better than our method with 95% confidence interval.

3.6.3 Limitations

Through error analysis on the results of the experiment, some limitations in our method were found. First, there are a lot of misinterpreted words in sentiment identification and concept expansion, which leads to misclassification of sarcastic tweets. Let us consider an example tweet "I had a terrible fight with my friend in the garden. I beat him around the bush." Our concept expansion module obtained one positive concept "come out better in a competition race" and one negative concept "give a spanking to" from the word "beat". Since the intensity of the positive concept is stronger than the negative concept in this example, our system will recognize "beat" as a positive word. Thus sentiment contradiction is wrongly identified, causing misclassification of this tweet as sarcastic. Sometimes, inappropriate concepts still remain even though we apply concept pruning.

Second, implicit sentiment in the tweet sometimes causes the error. In the tweet "I just love how you tweet all these other girls.", there is only a positive word "love" and no contradiction of the sentiment. Therefore, the system misclassifies it as a normal tweet. However, the phrase "how you tweet all these other girls" shows jealous emotion of the user and implies negative situation. Since neither the sentiment lexicon nor our concept expansion module can detect the sentiment in this phrase, it is rather difficult for the system to identify this tweet as sarcastic. Finally, the method of coherence identification should also be refined. Let us consider an example tweet "cold, sad & sleepy. that's a perfect combo. goodnight." This tweet contains both a positive word ("perfect") and negative words ("cold" and "sad"). However, our coherence identification method failed to identify coherence in the tweet, causing false-negative error. Recall that the demonstrative noun phrase feature is used in our method. The demonstrative "that" in this tweet may indicate coherence between the first and second sentences. However, the weight of this feature trained by CC-FWO is too small to identify coherence.

Other general causes of errors are also found. #sarcasm hashtag is sometimes used to indicate sarcasm of other previously posted tweets. Let us consider the tweet "@john4768 That was #sarcasm". In this example, not this tweet but a previous tweet of the user "@John" was written in sarcastic manner. In such cases, #sarcasm is not an indicator of the sarcastic tweets. In addition, in our datasets, there are a lot of sarcastic tweets that provide absolutely no clue. For example, it is impossible to recognize sarcasm within the tweet "I feel great #sarcasm" if #sarcasm is not attached.

3.7 Summary

This chapter proposed the novel method for identification of sarcasm in the tweets. It used word N-gram, sentiment score, sentiment contradiction, and punctuation & special symbol as the features for supervised machine learning. Coherence in the tweet was considered to derive the sentiment contradiction feature, which was identified by both the heuristic-based coherence identification and Coherence Clustering with Feature Weight Optimization (CC-FWO). To enhance the features that rely on the polarity score of the sentiment words, the methods of concept expansion and pruning were also presented.

Beyond recognition of sarcasm, we will focus on more practical sentiment analysis.

That is, we will explore a method to guess the polarity and/or the intensity of the sentiment of both sarcastic and non-sarcastic sentences in not only tweets but also product reviews or news articles. Our work towards this direction will be reported in Chapter 4 and 5.

Chapter 4

Sentiment Analyzer with Rich Features for Sarcastic Tweets

In this chapter, we introduce a sentiment analysis system created with a particular focus on the identification and proper elaboration of sarcasm in tweets. We take account of a task to guess a sentiment score of the sarcastic tweets. Here the sentiment score refers to a value on an 11 points scale ranging from -5 to $+5$. The positive and negative value indicate the positive and negative polarity, respectively. While the absolute value of the score indicates the intensity of the polarity. We have already proposed several features for sarcasm identification in the previous chapter. This chapter investigates if these features for sarcasm identification are also useful for the task of sentiment analysis, i.e. estimation of the sentiment score. Basically, our solution is to combine two kinds of features, one is our proposed features for sarcasm identification, the other is the feature proposed by previous work for sentiment analysis of sarcastic tweet.

Our method is divided into two main modules as shown in Figure 4.1. Each module generates various kinds of features, which will be used to classify the sarcastic tweets on an 11 points scale score. The module 1 derives the features used in the sentiment analysis system proposed by Xu et al. [15], while the module 2 derives our proposed features, including sentiment score, sentiment contradiction and punctuation & special symbols, as described in Section 3.2. Also in the module 1, we propose some additional features indicating the strong emotion of the Twitter user that contributes to estimation of the final sentiment score. Then, a classifier is trained with all the features extracted by both

the modules 1 and 2. The decision tree regression algorithm RepTree [94] implemented in Weka [95] is used for training and estimating the sentiment intensity of figurative language. Hereafter, we call this sentiment analysis SA-SAR (**S**entiment **A**nalyzer for **Sar**castic tweets). The method is evaluated on the dataset released by the organizers of the SemEval 2015 task 11. The results show that our method largely outperforms the systems proposed by the participants of the task on ironic and sarcastic tweets.

This chapter is structured into six parts. Section 4.1 begins with the procedures of data preprocessing. Section 4.2 discusses the implementation of the module 1 of our system. Section 4.3 discusses the implementation of the module 2. Section 4.4 describes how the experiment is conducted to evaluate the performance of the module 1, module 2 and integrated system of them. Section 4.5 discusses the results and effectiveness of our proposed method. Finally, the chapter will finish with a summary in Section 4.6.

4.1 Data preprocessing

Before extracting the features, the tweets were preprocessed using the Stanford Lemmatizer in order to transform the words in the tweets into lemmas. Then, a set of heuristic rules was created to handle irregularity of the texts that cannot be recognized by the Stanford Lemmatizer. Words in tweets may contain repeated vowels (e.g. "looooove") or unexpected capitalization (e.g. "LOVE") to emphasize certain sentiments or emotions. Thus, the repeated vowels are removed (e.g. from "looooove" to "love") and the capital letters are converted to lower case (e.g. from "LOVE" to "love") to improve the lemmatization and parsing accuracy. The heavy punctuation is also handled. The use of combination of exclamation and question marks (e.g. "?!?!!") will be replaced with only a single mark (e.g. "?!"). Although the repeated vowels, capitalized words and heavy punctuation are normalized, the appearance of them is saved and used as one of the features in the classification process. Another step of the preprocessing is the segmentation of the words. The segmentation is, in fact, often lost in tweets (e.g. "yeahright"). Therefore, the maximal matching algorithm is applied to segment the words (e.g. "yeah right"). In addition, all usernames, URLs and hashtags are removed from tweets as they do not provide any information about the sentiments and they might become noise for the clas-

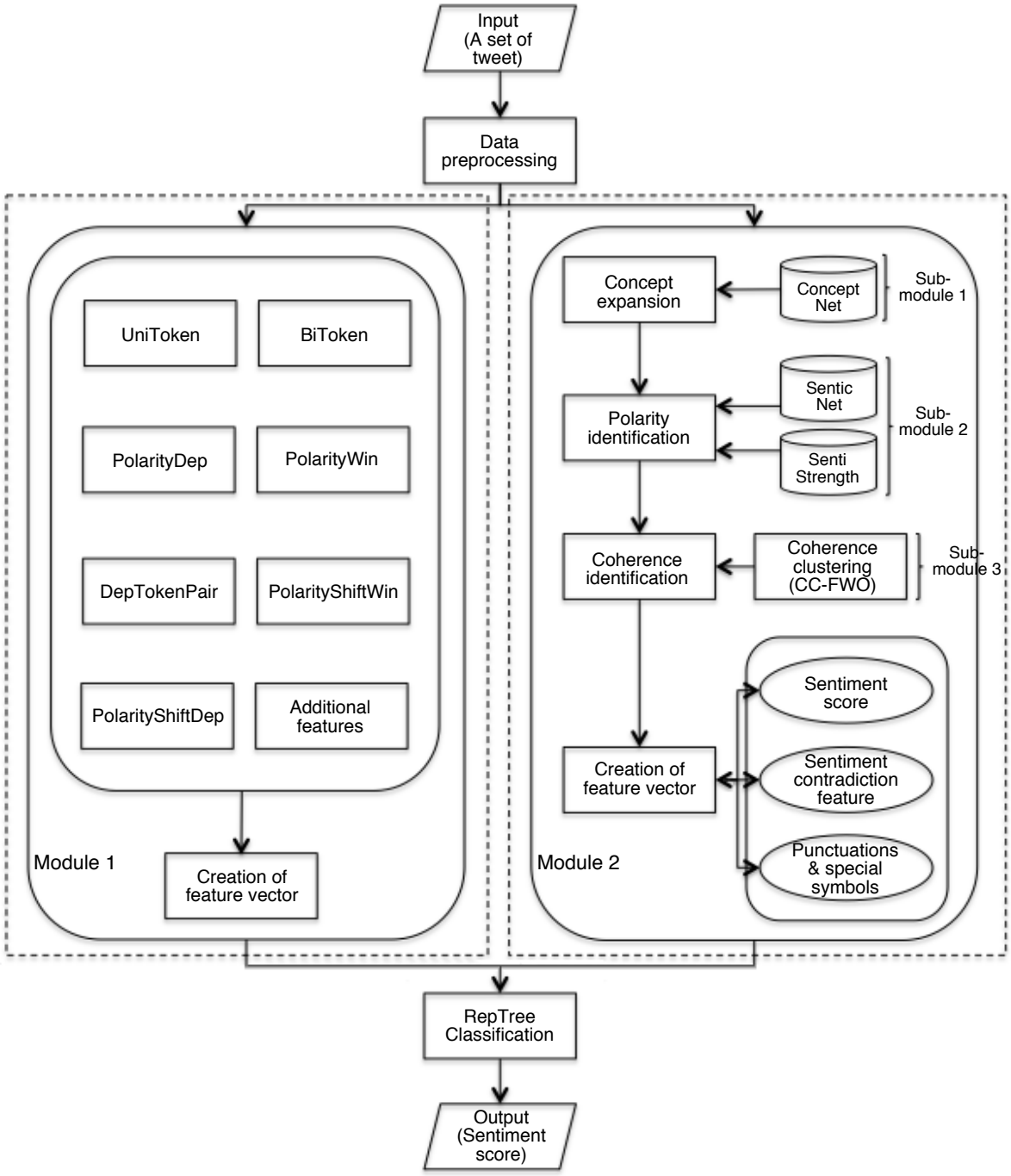

Figure 4.1: Flowchart of overall process of sentiment analyzer for sarcastic tweets

sification process. Finally, the Stanford parser[1] was used to generate the POS tags and dependency structures of the normalized tweets.

[1]http://nlp.stanford.edu/software/lex-parser.shtml

4.2 Module 1

The module 1 is based on the sentiment analysis system that participated in SemEval 2015 task 11 [15]. As shown in Figure 4.1, the feature extraction sub-module derives eight kinds of features for a given tweet. They are categorized into two groups: token based features and polarity dictionary based features.

- Token based features:
  - The "UniToken" refers to uni-grams of tokens.
  - The "BiToken" refers to bi-grams of tokens.
  - The "DepTokenPair" refers to "parent-child" pairs in the dependency structures of the tweets.
  - The "additional features" refers to the emphatic features capturing four ways twitter users express their emotions: *duplicate_vowel* ("loooove"), *capitalized* ("LOVE"), *heavy_punctuation* ("?!?!?"), and *emoticon* (":-D").
- Polarity dictionary based features:
  - The "PolarityWin" stores the sum of the polarity values of all the tokens in a tweet. A window size of five is used to verify whether negations are present. If a negation is present, the resulting value is set to zero. Besides, the sum of the polarity values of the tokens of the same POS tags are also stored in a different dimension. This is to measure the contributions on polarity values by different POS tags.
  - The "PolarityDep" is similar to "PolarityWin", but it differs in that the negation is checked based on the dependency structure.
  - The "PolarShiftWin" measures the difference between the most positive item and the most negative item in a window of size 5.
  - The "PolarShiftDep" measures the polarity difference of "parent-child" pairs in the dependency structures of the tweets.

To extract the polarity dictionary based features, four sentiment dictionaries were used: Opinion Lexicon [61], Afinn [96], MPQA [97], and SentiWordnet [98]. Two additional

dictionaries, which are the union and intersection of four sentiment lexicons, are also used. Formally, the polarity feature can be represented as a (*key, val*) pair, where the key is <*pos, dict*>. For example, (<*adj, mpqa*>, 1.0) means that according to the dictionary MPQA, adjectives contribute to the polarity value for 1.0.

In order to avoid noise and sparseness, only features that occur less than three times are excluded. All the feature values are normalized into the range [-1, 1] according to the formula shown in Equation (4.1), where $f_{i,j}$ is the value of feature j in the ith example, and N is the sample size.

$$norm(f_{i,j}) = \frac{f_{i,j}}{\max\limits_{1 \leq k \leq N} |f_{k,j}|} \tag{4.1}$$

$$r(X,Y) = \frac{\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y}))}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{N}(Y_i - \bar{Y})^2}} \tag{4.2}$$

We perform feature selection through the correlative coefficient measure (Pearson's r score). A threshold value of r is used to rule out less important features. The calculation of r is described in Equation (4.2), where X and Y are the two variables that are evaluated, X_i is the ith sample value of X, Y_i is the ith sample value of Y and N is the sample size.

To find the optimal threshold value of r with all the features listed in this module, two different models; Decision Tree Regression model (RepTree) and Support Vector Regression model (SVR); are used to trained classifiers on observation dataset. From the experiment, RepTree is better than SVR and the optimal correlative coefficient threshold is found when $r = 0.035$.

The module 1 is not exactly the same as the system presented in [15] but slightly revised in this thesis. To be more precise, "additional feature" is newly introduced in the module 1. We strongly believe that the emotional expressions are useful for sentiment analysis.

4.3 Module 2

The second module relies on features that were proven to be effective in the sarcasm identification of the tweets. That is, in this module, we use the same features as explained in Section 3.2. These features include sentiment score feature, sentiment contradiction

feature and punctuation & special symbol feature. Note that weights of all features in the module 2 are binary. To derive the proposed features, three sub-modules are used: concept expansion, polarity identification and coherence identification. In this section, we provide a brief summary of each sub-module.

4.3.1 Concept expansion sub-module

In this sub-module, ConceptNet is used to expand the concepts for the words whose sentiment scores are unknown in SentiStrength lexicon [49, 84]. The expanded concepts are used in the sub-module of polarity identification presented in the next subsection. They provide effective information that would benefit the task of sentiment analysis. Note that the concept pruning is not applied in this sub-module.

4.3.2 Polarity identification sub-module

In the second sub-module, the sentiment polarity scores are calculated for each word and its expanded concepts within a tweet. Then, we create seven features. Six of them are the sentiment score features, which are indicators of positive and negative phrases according to three possible classes (*low*, *medium* and *high*). In addition, sarcasm can be recognized as a contrast between a positive sentiment referring to a negative situation [36]. Thus, another feature is created as the sentiment contradiction feature. This feature is basically activated when there exists both a positive and a negative polarity word within a tweet.

4.3.3 Coherence identification sub-module

As explained earlier in Section 3.2.2, the contradiction of the polarity in a tweet is a useful clue. However, if positive and negative sentences mention different topics (i.e. they are incoherent), conflict of the polarity may not indicate the sarcasm. Therefore, this sub-module identifies coherence in a tweet.

The proposed Coherence Clustering with Feature Weight Optimization (CC-FWO) is based on unsupervised learning approach as described in Subsection 3.4.2. To divide the tweets into coherence and incoherence class, the following eleven features are created: Pronoun feature 1, Pronoun feature 2, String match feature, Definite noun phrase feature,

Demonstrative noun phrase feature, Both proper names feature, Coreference resolution feature, Semantic class agreement feature, Number agreement feature, Acronyms or abbreviation feature and Emoticon feature. Our proposed features are summarized in Table 3.1. After conducting a preliminary experiment, we found that the EM (expectation maximization) algorithm outperforms other clustering methods, that is hierarchical, k-mean and DBScan [99], in the identification of coherence in tweets. Note that DBScan is performed and compared with others for the development of the module 2, while only other three clustering algorithms are investigated in Subsection 3.4.2. Therefore, EM algorithm is used to cluster the tweets into two groups, one for coherent and one for incoherent tweets. Then, cluster labels are used in the sentiment contradiction feature.

4.3.4 Punctuation and special symbols

In addition to the sentiment score feature and sentiment contradiction feature, the punctuations & special symbol feature is also extracted by the module 2. The following 7 indicators are considered to determine the weights for punctuation features: the number of emoticons, the number of repetitive sequence of punctuations, the number of repetitive sequence of characters, the number of capitalized words, the number of slang and booster words, the number of exclamation marks and the number of idioms. The frequency of each type of punctuation and special symbol in a tweet is classified into *low*, *medium* and *high*. Therefore, the punctuation and special symbol features amount to $7 \times 3 = 21$.

4.4 Experiment

In this section, we describe how the experiments were conducted to evaluate the performance of our method.

4.4.1 Data

In our experiment, we used the training and test data distributed for SemEval-2015 Task 11 "Sentiment Analysis of Figurative Language in Twitter"[2]. The data set consists of tweets containing sarcasm, irony, metaphor and non-figurative tweets. The training set

[2]http://alt.qcri.org/semeval2015/task11/

contains 7,952 tweets, while the test set contains 4,000 tweets. All tweets are manually annotated with a fine-grained sentiment scale value in 11 points (between −5 to +5).

4.4.2 Task

The task definition of this experiment is same as SemEval 2015 task 11. The goal is to estimate a fine-grained sentiment score, which is eleven values from −5 to 5 for each tweet. The predicted scores are compared with the values annotated in the SemEval 2015 data to evaluate the performance of the sentiment analysis systems.

We performed two subtasks. One is to estimate the sentiment score by 5-fold cross validation on the training set. In this task, the effectiveness of individual features is mainly investigated. The results of this subtask will be reported in Subsection 4.5.1. The other is to estimate the sentiment polarity and intensity of the test set using the model learned from the training data. The performance of the proposed method is analyzed considering several types of tweets (sarcastic, ironic, metaphorical and non-figurative ones). The results of the second subtask will be reported in Subsection 4.5.2.

4.4.3 Evaluation measures

Cosine similarity and root mean squared error (RMSE) are used as the evaluation criteria of sentiment intensity estimation. They illustrate how similar the predicted values and the actual annotated values are. They can be calculated by using Equation (4.3) and (4.4), respectively.

$$Cosine[a,b] = \frac{\sum_{i=1}^{n} a_i \times b_i}{\sqrt{\sum_{i=1}^{n}(a_i)^2} \times \sqrt{\sum_{i=1}^{n}(b_i)^2}} \tag{4.3}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - b_i)^2} \tag{4.4}$$

- i refers to the value of tweet index.
- n refers to the number of tweets.
- a_i refers to the human-annotated sentiment score of tweet i.
- b_i refers to the estimated sentiment score of tweet i by the system.

Table 4.1: Results of the module 1 of 5-fold cross validation on the training data

Method	**Cosine**	**RMSE**
Avg. polarity (Baseline1)	0.818	1.985
UniToken (Baseline 2)	0.854	1.679
UniToken		
+BiToken	0.849	1.700
+DepTokenPair	0.851	1.673
+PolarityWin	0.852	1.657
+PolarityDep	0.854	1.643
+PolarityShiftWin	0.854	1.640
+PolarityShiftDep	0.854	1.640

4.5 Results and discussion

4.5.1 Results on the training data

Table 4.1 shows the results of the two baselines and those of the module 1 trained with UniToken and one additional feature. Baseline 1 is a system that estimates the polarity score by calculating the average of sentiment scores of the words in the tweet. Baseline 2 is the classifier trained only with UniToken feature. Surprisingly, the average polarity value (Baseline 1) and the classification based on UniToken feature (Baseline 2) were powerful predictor of the sentiment. Both methods achieved relatively high cosine values (i.e. 0.818 and 0.854, respectively). In particular, it is interesting to notice that Baseline 1 can performed well because the majority of the tweets were annotated with moderate negative values, varying from -2 to -3. On the other hand, the average polarity value of words computed by our baseline system also indicated the moderate negative range for many tweets.

Thus, Baseline 1 achieved a high accuracy and also became competitive with other methods. Next, let us discuss the effectiveness of individual features in the module 1.

Module 1

Next, let us discuss the effectiveness of individual features in the module 1.

- **BiToken and DepTokenPair**

  According to RMSE shown in Table 4.1, all features have taken part in the method to enhance the accuracy, except for BiToken. Adding BiToken feature caused decrease in the performance. Thus, we can easily conclude that BiToken is not a relevant feature for sentiment prediction of figurative tweets. DepTokenPair is not also so effective, although a little improvement is found by adding this feature.

- **PolarityWin and PolarityDep features**

  The features contributed some improvements to RMSE. The reason is that these features handle the negations, which often occurs within the figurative tweets.

- **PolarityShiftWin and PolarityShiftDep features**

  The result also indicates that PolarityShiftWin and PolarityShiftDep features contributed to some improvement towards RMSE. The difference between the most positive and negative items can represent the strength of the overall polarity and also indicate if there exists a conflict in a tweet, which may reveal either irony or sarcasm. As a result, we can conclude that the shift in polarity value has an impact on the sentiment analysis for figurative tweets.

Module 2

Table 4.2 shows the overall result of the module 2 and also how the results change as each feature is removed from the system with all features. Cosine value and RMSE of the module 2 were 0.829 and 1.369, which were better than Baseline 1 but worse than Baseline 2 in Table 4.1. Recall that, in the experiment in Subsection 3.6.2, the performance of the system using only our proposed features was worse than the system using N-gram feature in sarcasm identification task. In the sentiment score estimation task, the use of only the proposed features seem also insufficient. Next, we discuss the contribution of our proposed features and sub-modules.

- **Punctuations and special symbols**

  The feature contributed to some improvement to both the cosine and RMSE. Figurative

Table 4.2: Results of the module 2 of 5-fold cross validation on the training data

Method	**Cosine**	**RMSE**
All features (module 2)	0.829	1.369
– Sentiment contradiction	0.823	1.380
– Sentiment score	0.806	1.501
– Punctuations + symbols	0.825	1.376
– Coherence	0.819	1.419
– Concept level knowledge	0.785	1.649

tweets often contain emoticons and heavy punctuation marks to simulate the gestural signs, onomatopoeic expressions and also boosting the intensity of emotion. Therefore, the feature can be used to capture this particular characteristic.

- **The concept-level knowledge**

  Expansion of the concepts implemented in the first sub-module can also enhance the performance of the sentiment score estimation. Tweets are considered as unstructured and context free data. There are many words and slangs, which cannot be compiled in any dictionaries. Concept-level and common sense knowledge are applied to compensate to such lack of the words in the sentiment dictionary, which allows the system to compute the sentiment score more accurately.

- **Coherence identification**

  In our experiment, it is clearly shown that coherence feature has an impact on the improvement of the result. This is a proof that it is necessary to verify whether there are terms referring to each other across the sentences, in order to make the contradiction identification more effective.

Integration of two modules

Table 4.3 shows the comparison of the module 1, module 2 and integration of them (i.e. our proposed system SA-SAR). The results show that SA-SAR performs significantly better than the Baseline 2 (0.854 Cosine and 1.679 RMSE) that uses uni-gram feature.

Table 4.3: Results of the integrated system of 5-fold cross validation on the training data

Method	**Cosine**	**RMSE**
Module 1	0.859	1.256
Module 2	0.829	1.369
Integrated module 1 & 2 (SA-SAR)	0.891	1.147

Table 4.4: Results of the module 1 on the test dataset

Category	**Cosine**	**RMSE**
Sarcasm	0.896	0.997
Irony	0.918	0.671
Metaphor	0.535	3.917
Non-figurative	0.290	4.617
Overall	0.687	2.602

It is also clearly shown that the cosine value of the integrated system outperforms each module 1 and 2 by 0.032 and 0.062, respectively.

4.5.2 Results on the test data

In this subsection, our proposed systems are evaluated on the test data. Table 4.4 shows the results of sentiment prediction of the module 1 for four individual categories of tweets

Table 4.5: Results of the module 2 on the test dataset

Category	**Cosine**	**RMSE**
Sarcasm	0.949	0.730
Irony	0.916	0.846
Metaphor	0.396	4.155
Non-figurative	0.228	4.582
Overall	0.554	2.008

Table 4.6: Results of SA-SAR on the test dataset

Category	**Cosine**	**RMSE**
Sarcasm	0.954	0.715
Irony	0.923	0.820
Metaphor	0.572	3.892
Non-figurative	0.300	4.193
Overall	0.748	1.369

Table 4.7: Paired t-test for comparison between SA-SAR and each module

Pair	**Two-tailed P value**
1. Module 1 - SA-SAR	0.098
2. Module 2 - SA-SAR	0.041

(sarcasm, irony, metaphor and non-figurative) as well as all categories in the test data. The performance is good on sarcastic and ironic data, since the module 1 achieved the cosine value of 0.896 and 0.918, respectively. However, the performance is rather poor when we attempted to estimate the sentiment score for metaphor and non-figurative tweets. Table 4.5 shows the classification results of sentiment prediction of the module 2. Comparing to the module 1, the cosine value was higher for sarcastic tweets and comparable for ironic tweets. The major differences in the module 1 and 2 are the use of the sentiment lexicon, concept expansion and coherence feature. Sometimes, the usage of concept expansion can cause an error by reducing the actual polarity intensity of the words. Let us consider a sarcastic tweet "@dana_mrivas: Oh my lord .. I just love rumors #sarcasm smh text me boo". This tweet contains one positive word "love" and one negative word "rumors". In module 1, the method was able to identify all of sentiment words correctly since it used four sentiment lexicons to extract the polarity. However, in module 2, the method used only SenticNet and SentiStrength to get the sentiment scores of the words. When, the target word does not exist within the SenticNet and SentiStrength, the average sentiment score of the concept extracted by ConceptNet will be considered as the sentiment score of it. In this example, the average score obtained from the concept expansion was -1, which

Table 4.8: Comparison of our SA-SAR against five top systems participated in SemEval 2015 Task 11

System	All	Sarcasm	Irony	Metaphor	Non-figurative
ClaC	**0.758**	0.892	0.904	**0.655**	**0.584**
UPF	0.711	0.903	0.873	0.520	0.486
LLT_PolyU	0.687	0.896	0.918	0.535	0.290
LT3	0.658	0.891	0.897	0.443	0.346
elirf	0.658	0.904	0.905	0.411	0.247
Our system	0.748	**0.954**	**0.923**	0.572	0.300

Note: ClaC = Concordia university; UPF = Universitat Pompeu Fabra; LLT_PolyU = Hong Kong Polytechnic University; LT3 = Ghent University; elirf = Universitat Politecnica de Valencia

was a lot lower than the scores in the existing sentiment lexicons in the module 1. Such underestimate of the polarity intensity of the unknown words may cause more errors in the module 2 than the module 1. Nevertheless, both modules could guess the sentiment score of the tweet in sarcastic class accurately enough.

Table 4.6 shows the results of SA-SAR that is the integration of the module 1 and 2, clearly indicating that the overall result of the proposed method is much better than both the module 1 and 2. Statistical test is carried out for checking the significance of difference between SA-SAR and each module. Table 4.7 shows two-tailed P values for two pairs of the systems. SA-SAR is significantly better than Module 1 and 2 with 90% and 95% confidence interval, respectively. Thus, the feature sets of both modules can complement each other when they are integrated into a single method.

Table 4.8 shows the comparison of the cosine measure among our system and the five top systems participated in SemEval 2015 Task 11. Note that our system largely outperformed all the other 15 participating systems on the ironic and sarcastic tweets, although achieved second in the overall dataset.

The performance of our system as well as the participating systems in SemEval 2015 was much better for the sarcasm and irony than metaphor and non-figurative. It may be worthy noticing here that most of the mentioned models were developed keeping in

mind that sarcasm and irony mostly rely on incongruity (i.e. logical inconsistency), while metaphor and non-figurative texts rely on congruity[3]. Therefore, the systems designed to identify incongruity poorly perform on the congruous texts. It suggests that the sarcasm/irony and metaphor/non-figurative are needed to be handled differently.

4.6 Summary

In this research, we present SA-SAR, a model for the estimation of fine-grained sentiment score for sarcastic tweets. The method consists of two modules that extract two sets of features, one is based on the previous work for sentiment analysis of figurative language, the other is a set of features proposed for sarcasm identification. The results of the experiments indicate that our proposed method is better than the strong baselines, and integration of two modules achieves the best result among the participating systems in SemEval-2015 for the sarcastic and ironic tweets. Furthermore, the contribution of each feature has been carefully analyzed and reported.

As discussed before, SA-SAR works well for sarcastic and irony tweets, but not for metaphor and non-figurative tweets. It leads us an idea of two step analysis: the first step is to identify if the given tweet is sarcastic or not. The second step is to guess the sentiment score of the tweet. In the second step, two classifiers are trained. One is the system proposed in this chapter, the other is a classifier trained from normal (not sarcastic) tweets. The former is applied only for the tweets that is judged as sarcastic at the first step. The latter is applied for other tweets. Next chapter will report an attempt for this direction.

[3]In metaphor, a concept in a target domain is expressed by terms from a source domain, but there is no incongruity among the used terms and concepts.

Chapter 5

General Sentiment Analyzer for Microblogging

In this chapter, we develop a practical sentiment analyzer that can handle both sarcastic and non-sarcastic tweets. In general, a sentiment analyzer is a powerful tool that automatically extracts sentiments (positive and negative ones), opinions and emotions (liking, anger, disgust, etc.) from unstructured text. In a narrow sense, the sentiment analyzer in this chapter refers to a system that can predict the sentiment score for a given tweet. Generally, there are two approaches toward the implementation of a sentiment analyzer: the lexicon-based and the machine learning-based approach. Our system is constructed by combining three existing sentiment analyzers, one is lexicon-based method and two are machine learning-based methods, and our proposed system, which is also based on machine learning. Our proposed system also utilizes the sarcasm identification method proposed in Chapter 3. Hereafter we call this sentiment analysis system SA-GEN (**S**entiment **A**nalyzer for **Gen**eral tweets). The details about the implementation of SA-GEN will be explained in Section 5.1.

In addition, we explore the way to utilize the proposed sarcasm recognition method to other NLP application. We introduce a way to enhance the accuracy of the target-dependent sentiment analysis system. The purpose of this application is to classify the sentiments (positive, negative or neutral) expressed toward a target such as a product, person and company on Twitter. We would like to show that our proposed sarcasm identification method can be integrated or merged into an existing target-dependent sen-

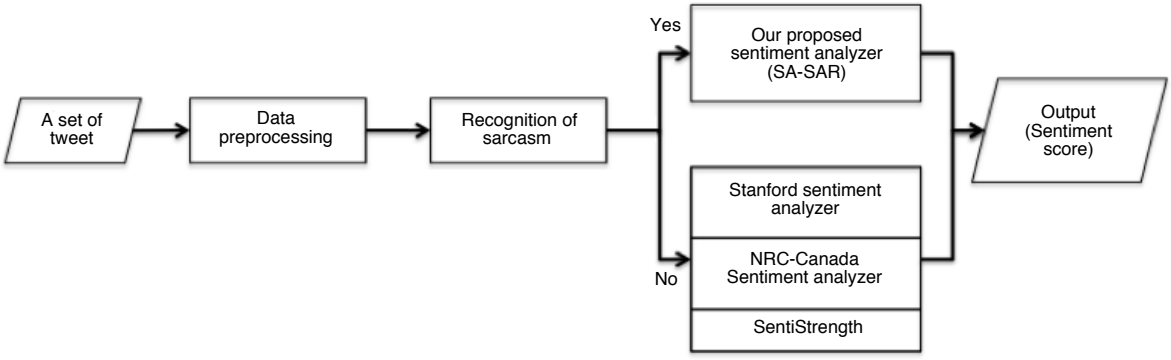

Figure 5.1: The overall method of our proposed sentiment analyzer for Microblogging

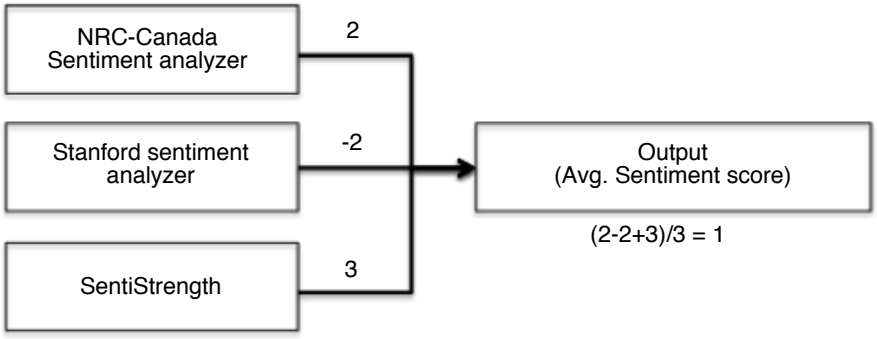

Figure 5.2: Calculation of sentiment score for normal tweets

timent analysis system proposed by Kaewpitakkun et al. [100] in order to improve its performance. The integration procedures will be explained in Section 5.3.

This chapter is structured into 4 parts. Section 5.1 begins with the explanation about the sentiment analyzer based on our proposed method and the existing systems. Section 5.2 will empirically evaluate the proposed sentiment analyzer. Then, Section 5.3 will explain how to integrate our proposed sarcasm identification system to a target-dependent sentiment analysis method. It will also report results of an experiment to evaluate the contribution of our method. Finally, the chapter will finish with a summary in Section 5.4.

5.1 Sentiment analysis of tweets

This section describes the implementation of our sentiment analyzer SA-GEN. It accepts a set of the tweets as an input, and estimates a sentiment score of each tweet. The sentiment score is an integer in a range from -5 to 5 that represent orientation and

Figure 5.3: The overall procedure of NRC-Canada sentiment analysis system

intensity of sentiment of a given tweet. Figure 5.1 shows the overall process of our proposed sentiment analyzer. The method consists of three main steps. First the input data (a set of tweets) is preprocessed, i.e. removing stop words, lemmatizing and so on. We use the same preprocessing procedures as explained in Section 3.1. Second, the method of sarcasm recognition described in Chapter 3 is applied to identify whether the input tweet is sarcastic or not. Finally, the output sentiment score will be calculated in different ways in terms of the result of sarcasm recognition. If the tweet is judged as sarcastic, the score is calculated by our proposed sentiment analyzer (SA-SAR) described in Chapter 4, which especially focuses on analysis of the sarcastic tweets. Otherwise, three general sentiment analyzer are applied: NRC-Canada sentiment analyzer [73], Stanford sentiment analyzer [74] and SentiStrength [49]. The output is the average of the scores of these three systems. The average of the scores can be described as the following equation:

$$Avg\_score = \frac{NRC\_score + Standford\_score + SS\_score}{3} \tag{5.1}$$

Figure 5.2 illustrates a simple example of calculation of sentiment score for normal (non-sarcastic) tweet. Next, we will give a brief summary about each sentiment analyzer in the following subsections.

5.1.1 NRC-Canada sentiment analyzer

NRC-Canada sentiment analyzer is sentiment analysis system participated in SemEval Workshop 2013. The system is created to detect the sentiment of messages such as tweets and SMS and also to detect the sentiment of a term within a message. Figure 5.3 shows the overall process of the NRC-Canada sentiment analyzer. The system is consists of three main steps. First, the tweets are preprocessed by normalization of URLs and user IDs: any URLs and user IDs are converted into the unified form 'http://someurl' and '@someuser' respectively. Also, the tweets are tokenized and tagged with their parts-of-speech (POSs) using the Carnegie Mellon University (CMU) Twitter NLP tool [101].

Second, each tweet is represented as a vector of features shown in Table 5.1. Note that for "Lexicons" feature (6th row), $score(w, p)$ stands for the sentiment score of the token w and the polarity p defined in the sentiment lexicons. In NRC-Canada, three manually constructed sentiment lexicons (NRC Emotion Lexicon [102, 103], MPQA [97], Bing Liu Lexicon [61]) and two automatically constructed lexicons (Hashtag Sentiment Lexicon [104] and Sentiment140 Lexicon [105]) are used. Finally, SVM trained from a set of the labeled tweets classifies the input tweet into positive, negative or neutral class.

Table 5.1: Summary of the features in the NRC-Canada sentiment analyzer

N-grams	Presence or absence of contiguous sequences of 1, 2, 3, and 4 tokens
Character N-grams	Presence or absence of contigous sequences of 3, 4, and 5 characters
All-caps	Number of words with all characters in upper case
POS	Number of occurrences of each part-of-speech tag
Hashtags	Number of hashtags
Lexicons	- Total count of tokens in the tweet with $score(w, p) > 0$ - Total score $\sum_{w \in tweet} score(w, p)$ - The maximum score $= max_{w \in tweet} score(w, p)$ - The score of the last token in the tweet with $score(w, p) > 0$
Punctuations	- Number of repeated sequences of exclamation marks, question marks, and both exclamation and question marks - Presence or absence of the last token contains an exclamation or question mark
Emoticons	- Presence or absence of positive and negative emoticons - Last token is a positive or negative emoticon
Elongated words	Number of words with one character repeated more than twice
Clusters	Presence or absence of tokens that belong to each of 1000 token clusters produced by the Brown clustering algorithm on 56 million English tweets.
Negation	Number of negated contexts (e.g. no, shouldn't)

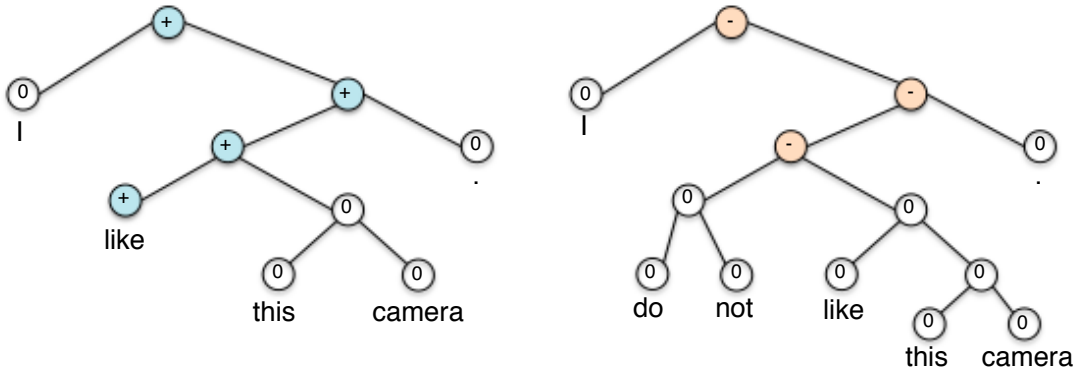

Figure 5.4: Example of positive (left) and negative (right) sentiment prediction based on the Recursive Neural Tensor Network

NRC-Canada sentiment analyzer can identify the polarity class (positive, negative or neutral) of a given sentence. To estimate the fine-grained sentiment score between -5 to 5, we train an SVM classifier with the same feature set shown in Table 5.1 from the training data annotated with the sentiment score.

5.1.2 Stanford sentiment analyzer

Stanford sentiment analyzer is a tool for identifying the sentiment of any given context. Many simple sentiment analysis tools estimate the sentiment score by just extracting the words in the context as bag-of-words, giving positive and negative score for each word and then summing up these scores. However, such methods are not effective since the important information such as the order of the words and the syntactic structure of the sentence is lost. Semantic vector space for single words has also been widely used as features for sentiment analysis [106]. However, similar to the bag-of-words model, the features are unable to capture the meaning of longer phrases properly.

The basic idea of Stanford sentiment analyzer is that the sentiment of a phrase or sentence can be determined by composition of the sentiments of words or smaller linguistic units, as the semantic interpretation of the sentence can be obtained by composition of the meaning of the phrases or words in it. It is based on deep learning, i.e. Recursive Neural Tensor Network (RNTN). Figure 5.4 illustrates how the sentiment of the sentence is identified by RNTN. The sentiment (indicated by + (positive) and − (negative)) of the internal nodes are determined from the sentiment of their children in bottom-up. RNTN

represents the words as the word vectors and computes the vectors for the higher nodes in the tree using the tensor-based composition function. From the vector representation of the sentences, the fine-grained sentiment labels are identified.

The Stanford Sentiment Treebank is used to train the parameters of RNTN in Stanford sentiment analyzer. It is a corpus with fully labeled parse trees. Thus a complete analysis of the compositionality of sentiment is possible. The corpus contains a large amount of labeled resources: 215,154 sentiment labeled phrases in 11,855 parse trees of the sentences. When the performance of Stanford sentiment analyzer was evaluated on the Stanford Sentiment Treebank, it achieved 80.7% accuracy on classifying the fine-grained sentiment labels.

5.1.3 SentiStrength

SentiStrength is a tool that estimates the strength of positive and negative sentiment in a short context. SentiStrength is a lexicon-based method that uses linguistic information and rules to detect sentiment strength in English text. The lexicon consists of all types of sentiment words, including booster words, emotion-bearing words, negating words, question words, slangs, idioms and emoticons. SentiStrength provides two integers as positive and negative sentiment score for a word. The score is scaled from 1 to 5 for both polarities, where 1 signifies weak sentiment and 5 signifies strong sentiment. If both positive and negative scores are 1, it means that the word has neutral polarity. Basically, the overall polarity of a context can be calculated by sum of positive sentiment scores of all words in the context subtracted by the sum of negative sentiment scores. Thelwall et al. reported that SentiStrength performed significantly above the baseline in six social web data sets that were substantially different in origin, length and sentiment content [49]. It proves that SentiStrength is a robust algorithm for sentiment strength detection on social web data.

5.2 Evaluation

An experiment is conducted to evaluate the performance of our proposed method. The task is to estimate a fine-grained sentiment score for each tweet in the dataset. We

Table 5.2: Results of individual sentiment analyzers

Method	**Cosine**	**RMSE**
SA-SAR	**0.748**	**1.369**
NRC-Canada sentiment analyzer	0.449	1.797
Stanford sentiment analyzer	0.428	1.980
SentiStrength	0.395	2.201

used the same training and test dataset as explained in Subsection 4.4.1. The dataset is distributed for SemEval-2015 Task 11 "Sentiment Analysis of Figurative Language in Twitter". All tweets are manually annotated with a fine-grained sentiment value in 11 points (between -5 to $+5$). Cosine similarity and root mean squared error (RMSE) are used as the evaluation criteria of sentiment intensity estimation of our application.

Results

Table 5.2 shows the results of sentiment score estimation of individual sentiment analyzers. Our proposed sentiment analyzer SA-SAR performed the best among the four systems. It achieved the cosine similarity score of 0.748 and root mean square error of 1.369. On the other hand, the results of NRC-Canada, Stanford and SentiStrength were much worse. It may be because these sentiment analyzers do not pay attention to sarcasm. Recall that 35% of the tweets in SemEval dataset are sarcastic.

Table 5.3: Improvement by integration of sentiment analyzers

Method	**Cosine**	**RMSE**
SA-SAR	0.748	1.369
+ Stanford sentiment analyzer	0.791	1.178
+ NRC-Canada sentiment analyzer	0.787	1.162
+ SentiStrength	0.760	1.283
+ ALL (SA-GEN)	**0.813**	**1.124**

Table 5.3 shows the results of sentiment score estimation of several sentiment analyz-

Table 5.4: Paired t-test results between Stanford, NRC-Canada or SentiStrength and SA-GEN

Pair	**Two-tailed P value**
1. NRC-Canada sentiment analyzer - SA-GEN	0.031
2. Stanford sentiment analyzer - SA-GEN	0.023
3. SentiStrength - SA-GEN	0.004
4. SA-SAR - SA-GEN	0.039

ers that integrates two or more systems. The third to sixth rows in Table 5.3 indicate the performance when our proposed sentiment analyzer is combined with one or all additional sentiment analyzers. Among three existing systems, the contribution of the NRC-Canada and Stanford sentiment analyzer was the best. Both methods achieved the cosine similarity score of 0.791 and 0.787 and root mean square error of 1.18 and 1.16, respectively. This could be because the methods are created based on machine learning models, which might enable the system to be more precise than the lexicon-based method (i.e. SentiStrength). Nevertheless, the result is the best when the sentiment score of normal tweet is computed using the average of three sentiment analyzers. It clearly shows that all of the sentiment analyzers contribute to improve the performance of the sentiment score estimation task.

Statistical test

In order to investigate the significance of the SA-GEN system, the difference between SA-GEN and NRC-Canada, Stanford, SentiStrength or SA-SAR is examined by a paired t-test. Table 5.4 shows two-tailed P values of four pairs: "pair 1" between NRC-Canada sentiment analyzer and SA-GEN, "pair 2" between Stanford sentiment analyzer and SA-GEN, "pair 3" between SentiStrength and SA-GEN, and "pair 4" between SA-SAR and SA-GEN. From these results, the differences of all pairs are considered to be statistically significant with 95% confidence interval.

5.3 Target-dependent sentiment analysis

5.3.1 Motivation and proposed method

This section focuses on a task of target-dependent sentiment analysis. The goal of this task is to identify the polarity (positive, negative or neutral) expressed toward a target such as a product, person, service or any kinds of entities. Some of sentiment analyzers aim at identifying the sentiment of a given sentence or document. However, even when people express sentiment in the overall sentence, they may not express the same sentiment toward the target. For example, the sentence "I feel down because I lost my surface pro." shows negative emotion, but not for the target "surface pro". In this section, we use a target-dependent sentiment analysis system proposed by Kaewpitakkun et al. [100], called **Ta**rget **S**pecific **K**nowledge **Sen**timent Classification (TASK-SEN). Let us review the overview of TAKS-SEN. Figure 5.5 illustrates the procedures of it. TASK-SEN is designed for the sentiment analysis on Twitter. First, three kinds of target-dependent resources, an add-on lexicon, extended target list and competitor list, are automatically constructed. The add-on lexicon complies specific words related to the target and their polarity scores. The extended target means a synonym or aspect of the target, while the competitor means an entity that can be compared with the target (e.g. "iPad" against the target "surface pro"). Then, the target-specific training data is constructed by the tweet-level sentiment analyzer and several heuristic[1] with the obtained target-dependent resources. Finally, SVM is trained from the data. The features used for training the classifier are uni-gram, part-of-speech, on-target sentiment feature and user-aware feature. On-target sentiment feature represents sentiment of the words near the target. While user-aware feature captures sentiment of the other tweets posted by the same user. For more detail, see Kaewpitakkun et al. [100].

However, one of the weaknesses of TASK-SEN is that it does not consider sarcasm in the tweets. We have already mentioned that many sarcastic tweets are posted on Twitter, and it is rather difficult to identify the sentiment of the sarcastic tweet. To improve the performance of target dependent sentiment classification, we integrate our proposed method of sarcasm recognition into TASK-SEN. We classify the tweets if they

[1]Neutral-to-Target Polarity Conversion and Competitor-to-Target Polarity Inversion in Figure 5.5.

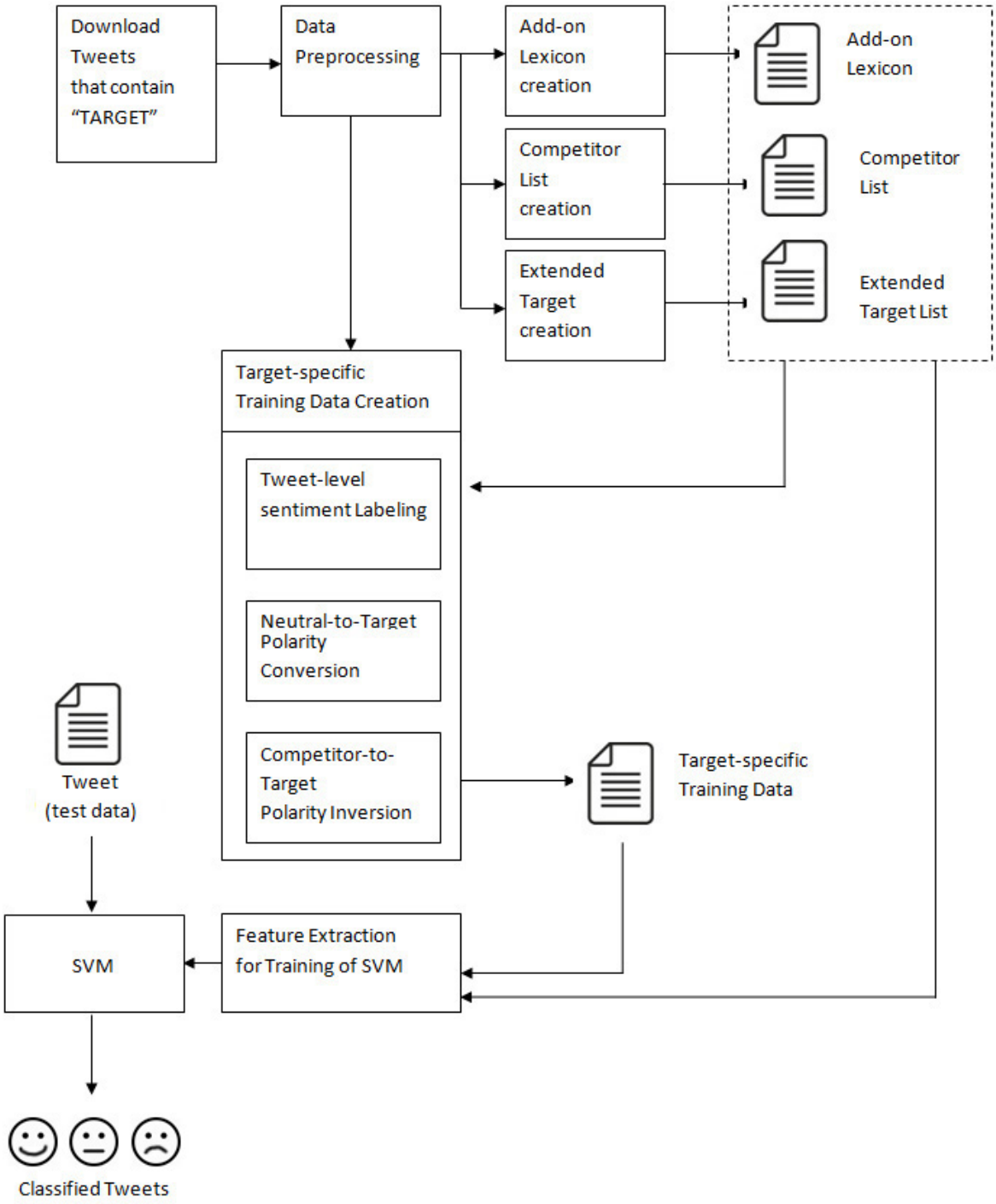

Figure 5.5: The overview of procedure of TASK-SEN system

are sarcastic by our proposed method, then the result of sarcasm recognition is used as a new feature (called sarcasm feature) in TASK-SEN. This allows the TASK-SEN system to recognize whether the tweet contains sarcastic expression.

5.3.2 Evaluation of target-dependent sentiment analysis

In this experiment, we evaluate the performance of integrated system of our proposed sarcasm recognition method and TASK-SEN system.

Data

The data presented in Kaewpitakkun et al. [100] is used for evaluation. In this data, three topics are chosen as the target, "iphone", "google" and "obama". For each target, 10,500 to 12,000 tweets containing the target keyword are retrieved via Twitter4J. It is used for constructing the target-specific knowledge and training data. For evaluation, another 300 tweets for each target are retrieved. They are manually annotated as positive, negative or neutral. F1-measure and accuracy on the test data are measured to evaluate our method.

Task

The task of this experiment is to classify the polarity of the target in the given tweet. We investigated the effectiveness of the sarcastic feature. The following two methods were compared.

- TASK-SEN without sarcasm feature: SVM classifier trained from the target-specific training data using uni-gram, POS, on-target sentiment and user-aware features.
- TASK-SEN with sarcasm feature: SVM classifier trained from the target-specific training data using uni-gram, POS, on-target sentiment and user-aware features as well as the sarcasm features.

Results

Table 5.5 and 5.6 shows the F1 measure and accuracy of the sentiment classification with and without the sarcasm feature, respectively. It is shown that our proposed sarcasm identification method can be integrated with TASK-SEN system to enhance the performance of target-dependent sentiment analysis. It is found that the sarcasm feature boost the performance of the sentiment classifier for three popular target tweets (iPhone, Google and Obama). We guess that the sarcastic tweets are often posted for popular targets such as iPhone, Google and Obama. However, the frequency of the sarcastic tweets may

Table 5.5: Results of target-level senitment analysis with sarcasm feature.

	TASK-SEN with Sarcasm	
Target	F1	ACC
iPhone	0.643	0.597
Google	0.678	0.620
Obama	0.559	0.513
Average	0.627	0.577

Table 5.6: Results of target-level senitment analysis without sarcasm feature.

	TASK-SEN without Sarcasm	
Target	F1	ACC
iPhone	0.633	0.587
Google	0.676	0.617
Obama	0.556	0.510
Average	0.622	0.571

depend on the target. That is, people often express their opinion in a sarcastic manner for some targets, while not for some other targets. It is necessary to conduct a large scale experiment that covers various types of the targets to investigate the contribution of the sarcastic feature in TASK-SEN.

5.4 Summary

In this chapter, a general sentiment analyzer, SA-GEN, that can estimate the fine-grained sentiment score for both sarcastic and non-sarcastic tweets. The method consists of our proposed sentiment analyzer and the other three sentiment analysis methods: NRC-Canada sentiment analyzer, Stanford sentiment analyzer and SentiStrength. The method achieved an outstanding accuracy of 0.813 for cosine similarity and 1.124 root mean square error on SemEval 2015 Task 11 dataset. As for the application of our proposed

sarcasm identification method, we introduce a way to enhance the accuracy of the target-dependent sentiment analysis system, i.e. TASK-SEN. Our method was integrated into TASK-SEN by using sarcasm labels as one of the features in the classification process. The classification results showed the integration of the sarcasm feature improved the performance.

Chapter 6

Conclusion

This chapter briefly reviews the proposed methods, shows answers for the research questions, clarifies the contribution of this study and discusses future directions.

6.1 Summary of Dissertation

In this thesis, we aimed to create a sentiment analysis system with a particular focus on sarcasm on microblogging.

First, we proposed a novel method to judge whether given tweets were sarcastic. The classifier for sarcasm identification was trained by a supervised learning algorithm. The proposed features were (1) sentiment score feature that captured emotional intension often found in sarcasm, (2) sentiment contradiction feature that captured inconsistency of the polarity in a tweet, and (3) punctuation & special symbol feature that also indicated strong emotion in the tweet. In addition to these features, ordinary N-gram features were also used. In the sentiment contradiction feature, the coherence among several sentences in a tweet was considered to improve preciseness of this feature. Two novel methods for coherence identification were proposed. One was based on heuristic rules, the other was Coherence Clustering with Feature Weight Optimization (CC-FWO). Furthermore, concept expansion and pruning were applied to guess the sentiment of unknown words to compensate insufficiency of the sentiment lexicons. The results of the experiment showed that our proposed method achieved the better accuracy (0.8320 on ARTK-300K and 0.7648 on SemEval dataset) compared to several baselines and previous approaches.

Second, a novel sentiment analysis system, called SA-SAR, which could guess a fine-grained sentiment scores for given tweets was proposed. It specially focused on analysis of sarcasm. It was trained by supervised learning with two kinds of the features. One is the features proposed by the previous work, which were N-gram, the polarity of the words, the difference of the sentiment scores of the most positive and negative words and so on. The other was the features used for sarcasm identification proposed by this thesis. The sentiment score predictor was trained by RepTree algorithm with these rich features. In the experiments, our proposed method was better than the strong baselines and achieved the best result among the participating systems in SemEval-2015 for the sarcastic tweets.

Third, we proposed a general sentiment analyzer, called SA-GEN, which could also estimate a fine-grained sentiment score and could handle both sarcastic and non-sarcastic tweets. It was implemented as two-step algorithm. In the first step, the given tweets were classified if they were sarcastic or not. In the second step, the sentiment score between -5 to 5 was estimated by different systems for each tweet. For the tweets judged as sarcasm by the first step, our SA-SAR was applied. On the other hand, for the tweets judged as non-sarcasm, three existing sentiment analyzers were applied. In other words, the proposed sentiment analyzer was ensemble of four kinds of sentiment analyzers including our proposed system as well as our proposed sarcasm recognition method. The results of the experiments showed that two-step sentiment analyzer was more effective than a single system.

Now the research questions presented in Chapter 1 can be answered as follows.

Q1 What are effective features to identify sarcasm in microblogging?

In Chapter 3, three kinds of features were proposed: sentiment score feature, sentiment contradiction feature and punctuation & special symbol feature. The results of the experiments indicated that each feature could contribute to improve the performance of the sarcasm identification task. Among them, the contribution of the sentiment contradiction feature was the best.

Coherence among sentences in a tweet should be considered to derive the sentiment contradiction feature, because the contradiction of the polarity in incoherent tweets does not always indicate sarcasm. It was supported by the fact that the sentiment

contradiction feature without coherence identification poorly performed in our experiment. To identify coherence in the sentences, we proposed the heuristic-based method and Coherence Clustering with Feature Weight Optimization (CC-FWO). In CC-FWO, the clustering of the tweets was performed to distinguish the coherent and incoherent tweets. Each tweet was represented by a feature vector for clustering using our proposed features, and the weights of the features were optimized by brute force search. The feature weighting could improve the classification accuracy by 1.2-1.6% in our experiment.

Q2 How to handle informal and short sentences in microblogging in sarcasm identification process?

Since the proposed method was heavily relies on the sentiment lexicons, we mainly tackled the problem that there were many unknown sentiment words in Twitter due to the usage of informal language. In Chapter 3, to guess the sentiment of the unknown words, the procedure of expanding the concepts of the words was introduced. Furthermore, to avoid expanding irrelevant concepts, the concept pruning based on word sense disambiguation was also introduced. In our experiments, the concept expansion could improve the accuracy by 0.1-1.9%, and concept pruning further improved the accuracy by 0.8-1.5%.

Q3 How to infer polarity and intensity of sentiment in microblogging, especially in sarcastic tweets?

In Chapter 4, a sentmeint analyzer with rich features was proposed. The features consisted of our proposed features for sarcasm identification and the features presented in the previous work. We showed that both features could be effective for predicting a fine-grained sentiment score via the experiment. The results of the experiments also showed that our sentiment analyzer could work well for the sarcastic tweets than the non-figurative tweet.

Q4 How to develop a general method to infer polarity and intensity of sentiment in microblogging?

Chapter 5 presented two-step sentiment analysis system consisting of the sarcasm identification step and sentiment score estimation step. In the second step, our

proposed SA-SAR and the existing sentiment analyzers were alternatively applied. The results of the experiments indicated that ensemble of these sentiment analyzers could improve the performance on the data set including both sarcastic and non-sarcastic tweets.

6.2 Contribution of our research

In this section, we discuss the contribution of our research on two aspects. First, we clarify what our research provides to the research field of natural language processing. Second, we discuss how our research can provide an impact to the society.

6.2.1 Research contributions

The research contribution of our methods can be summarized as follows.

- The novel feature that captured the sentiment contradiction in the coherent tweets was proposed. The results of our experiments indicated that coherence identification greatly contributed to improve the accuracy.

- The method of coherence identification based on heuristic rules was proposed. It was used to confirm that both positive and negative words in the tweet expressed the sentiment toward the same target; in such cases the sentiment contradiction strongly indicated sarcasm.

- The semi-supervised or distant supervision approach to identify the coherence in the tweets was also proposed. It was based on the unsupervised clustering and optimization of the weights of the feature vectors.

- The alternative way to tackle the data sparseness of the public sentiment lexicons was presented. It utilized ConceptNet as additional external knowledge. The effectiveness of our concept expansion and pruning were empirically investigated.

- The novel features that can identify the sentiment score of sarcastic tweets were proposed. The results of the experiments clearly showed the all our proposed features were effective for sentiment classification of sarcastic tweets.

- A general sentiment analyzer was built by integration of the proposed method and the existing sentiment analysis tools. It could improve the performance of the sentiment analysis for the data set including both sarcastic and non-sarcastic tweets.

Even for human, it is not easy to identify sarcasm in tweets because sarcasm often depends on common sense knowledge associated with the context of the tweets. It makes automatic identification of sarcasm difficult. We think that about 80% accuracy could be considered as a satisfying result.

6.2.2 Contribution on social impact

Our research can potentially provide a lot of benefits to the society. Recognition of sarcasm enables everyone to obtain more accurate information about people's opinions for various topics (e.g. commercial products, business, sports and politics). As mentioned earlier, the study of sentiment analysis on sarcasm have become very popular in the area of business, especially in the stock market and e-commerce. The method also allows companies or service providers to know precise opinions about their products or services, which are useful to improve their plans, decisions or business strategies. The method can prevent us from misinterpreting sentences whose meaning are opposite to their literal meaning. The proposed automatic sarcasm recognition would be helpful for us to overcome the difficulty in recognition of sarcasm which causes misunderstanding in our daily communication.

6.3 Future work

For future work, we intend to apply a personalization method to sentiment analysis task. Obviously, the intensity of the sentiment score of the words can be varied based on the personality and characteristic of each individual. Personalization is promising to improve the performance of the sentiment analysis of the both sarcastic and non-sarcastic sentences.

In addition, we would like to explore whether our proposed features are applicable in other domains. We plan to apply our feature set to different domains of dataset, including news, product reviews, chat dialogues etc. We are also interested in integrating proposed

methods into other NLP applications, including machine translation, text summarization and word sense disambiguation.

Bibliography

[1] L. O'Carroll, "Twitter active users pass 200 million," 2012. http://www.theguardian.com/technology/2012/dec/18/twitter-users-pass-200-million.

[2] O. Ashtari, "The super tweets of #sb47," 2013. https://blog.twitter.com/2013/the-super-tweets-of-sb47.

[3] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 1, pp. 1–14, 2015.

[4] M. R. Islam, "Numeric rating of apps on google play store by sentiment analysis on user reviews," in *Electrical Engineering and Information Communication Technology (ICEEICT), 2014 International Conference on*, pp. 1–4, April 2014.

[5] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content.," in *WebDB*, 2009.

[6] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The azfin text system," *ACM Trans. Inf. Syst.*, vol. 27, pp. 12:1–12:19, Mar. 2009.

[7] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using financial news articles," in *Americas Conference on Information Systems*, 2006.

[8] D. K. Pearce and V. V. Roley, "Stock Prices and Economic News," *The Journal of Business*, vol. 58, pp. 49–67, January 1985.

[9] G. McQueen and V. V. Roley, "Stock prices, news, and business conditions," *Review of Financial Studies*, vol. 6, no. 3, pp. 683–707, 1993.

[10] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1 – 8, 2011.

[11] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng, *Exploiting topic based twitter sentiment for stock prediction*, vol. 2, pp. 24–29. Association for Computational Linguistics (ACL), 1 2013.

[12] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, (Stroudsburg, PA, USA), pp. 151–160, Association for Computational Linguistics, 2011.

[13] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pp. 1320–1326, may 2010.

[14] P. Tungthamthiti, K. Shirai, and M. Mohd, "Recognition of sarcasms in tweets based on concept level sentiment analysis and supervised learning approaches," in *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation*, pp. 404–413, December 2014.

[15] H. Xu, E. Santus, A. Laszlo, and C.-R. Huang, "Llt-polyu: Identifying sentiment intensity in ironic tweets," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 673–678, June 2015.

[16] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[17] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. New York, NY, USA: Cambridge University Press, 2000.

[18] J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[19] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[20] R. G. D'Andrade, "U-statistic hierarchical clustering," *Psychometrika*, vol. 43, no. 1, pp. 59–67, 1978.

[21] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, (Berkeley, Calif.), pp. 281–297, University of California Press, 1967.

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

[23] F. Stringfellow Jr., *The Meaning of Irony: A Psychoanalytic Investigation*. State University of New York Press, 1994.

[24] H. P. Grice, "Logic and conversion," *Syntax and semantics 3: Speech arts*, pp. 41–58, 1975.

[25] R. Doerfler, "A comedy of errors or, how i learned to stop worrying and love sensibility-invariantism about 'funny'," *Pacific Philosophical Quarterly*, vol. 93, no. 4, pp. 493–522, 2012.

[26] R. K. Singh, "Humour, irony and satire in literature," *International Journal of English and Literature (IJEL)*, vol. 3, pp. 65–72, 2012.

[27] R. W. Gibbs Jr. and H. L. Colston, *Irony in Language and Thought: A Cognitive Science Reader*. Routledge, 2007.

[28] R. Kreuz and S. Glucksberg, "How to be sarcastic: The echoic reminder theory of verbal irony," *Journal of Experimental Psychology: General*, vol. 118, no. 4, pp. 374–386, 1989.

[29] J. M. Haiman, *Talk Is Cheap Sarcasm Alienation and the Evolution of Language*. Oxford University Press, 1998.

[30] D. Sperber and D. Wilson, *Irony and the use-mention distinction.* Academic Press, peter cole (ed.) ed., 1986.

[31] R. Schaffer, *Vocal Cues for Irony in English.* Ohio State University., 1982.

[32] D. Muecke, *The Compass of Irony.* Methuen library reprints, Methuen, 1969.

[33] R. J. Kreuz and R. M. Roberts, "On satire and parody: The importance of being ironic," *Metaphor and Symbolic Activity*, vol. 8, no. 2, pp. 97–109, 1993.

[34] R. W. G. Jr. and J. O'Brien, "Psychological aspects of irony understanding," *Journal of Pragmatics*, vol. 16, no. 6, pp. 523 – 530, 1991.

[35] A. Reyes, P. Rosso, and T. Veale, "A multidimensional approach for detecting irony in twitter," *Language Resources and Evaluation*, vol. 47, no. 1, pp. 239–268, 2013.

[36] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 704–714, 2013.

[37] A. Reyes and P. Rosso, "Making objective decisions from subjective data: Detecting irony in customer reviews," *Decision Support Systems*, vol. 53, no. 4, pp. 754–760, 2012.

[38] O. Tsur, D. Davidov, and A. Rappoport, "Icwsm – a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews," in *Proceedings of the Fourth International Conference on Weblogs and Social Media (ICWSM-2010)*, 2010.

[39] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcastic sentences in twitter and amazon," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pp. 107–116, 2010.

[40] K. Jahandarie, *Spoken and Written Discourse.* Standford, CT, 1999.

[41] J. Tepperman, D. Traum, and S. Narayanan, "Yeah right: Sarcasm recognition for spoken dialogue systems," in *Interspeech 2006*, (Pittsburgh, PA), sep 2006.

[42] A. Salvatore and E. Jodi, "Multimodal markers of irony and sarcasm," *Humor - International Journal of Humor Research*, vol. 16(2), pp. 243–260, 2003.

[43] P. Rockwell, "Vocal features of conversational sarcasm: A comparison of methods," *Journal of Psycholinguistic Research*, vol. 36, no. 5, pp. 361–369, 2007.

[44] P. Rockwell, "Empathy and the expression and recognition of sarcasm by close relations or strangers," *Perceptual and Motor Skills*, vol. 97(1), pp. 251–6, 2003.

[45] C. F. Burgers, *Verbal irony: Use and effects in written discourse.* PhD thesis, Radboud University Nijmegen, 2010.

[46] P. Carvalho, L. Sarmento, M. J. Silva, and E. de Oliveira, "Clues for detecting irony in user-generated contents: oh...!! it's so easy;-)," in *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pp. 53–56, 2009.

[47] F. Barbieri and H. Saggion, "Modelling irony in twitter," in *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 56–64, April 2014.

[48] G. A. Miller, "Wordnet: A lexical database for english," *Communications of the ACM*, vol. 38, pp. 39–41, Nov 1995.

[49] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social web," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 1, pp. 163–173, 2012.

[50] Y. Hao and T. Veale, "An ironic fist in a velvet glove: Creative mis-representation in the construction of ironic similes," *Minds Mach.*, vol. 20, pp. 635–650, Nov. 2010.

[51] K. Buschmeier, P. Cimiano, and R. Klinger, "An impact analysis of features in a classification approach to Irony Detection in Product Reviews," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, 2014.

[52] A. Scharl and A. Weichselbraun, "An automated approach to investigating the online media coverage of us presidential elections," *Journal of Information Technology and Politics*, vol. 5, no. 1, pp. 121–132, 2008.

[53] E. Boiy and M.-F. Moens, "A machine learning approach to sentiment analysis in multilingual web texts," *Information Retrieval*, vol. 12, no. 5, pp. 526–558, 2009.

[54] D. Maynard and A. Funk, *The Semantic Web: ESWC 2011 Workshops: ESWC 2011 Workshops, Heraklion, Greece, May 29-30, 2011, Revised Selected Papers*, ch. Automatic Detection of Political Opinions in Tweets, pp. 88–99. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[55] P. J. Stone, D. C. Dunphy, M. S. Smith, and D. M. Ogilvie, *The General Inquirer: A Computer Approach to Content Analysis*. Cambridge, MA, USA: MIT Press, 1966.

[56] R. M. Tong, "An Operational System for Detecting and Tracking Opinions in Online Discussion," in *SIGIR 2001 Workshop on Operational Text Classification*, Sept. 2001.

[57] V. Hatzivassiloglou and K. 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*, ACL '98, (Stroudsburg, PA, USA), pp. 174–181, Association for Computational Linguistics, 1997.

[58] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, (Stroudsburg, PA, USA), pp. 417–424, Association for Computational Linguistics, 2002.

[59] P. D. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Trans. Inf. Syst.*, vol. 21, pp. 315–346, Oct. 2003.

[60] J. Wiebe, "Learning subjective adjectives from corpora," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 735–740, AAAI Press, 2000.

[61] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, (New York, NY, USA), pp. 168–177, ACM, 2004.

[62] M. Taboada, C. Anthony, and K. Voll, "Methods for creating semantic orientation dictionaries," in *Conference on Language Resources and Evaluation (LREC)*, pp. 427–432, 2006.

[63] D. K. Gupta and A. Ekbal, "Iitp: Supervised machine learning for aspect based sentiment analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, (Dublin, Ireland), pp. 319–323, Association for Computational Linguistics and Dublin City University, August 2014.

[64] B. Liu, *Sentiment Analysis and Opinion Mining*. Synthesis digital library of engineering and computer science, Morgan & Claypool, 2012.

[65] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093 – 1113, 2014.

[66] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[67] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, (Stroudsburg, PA, USA), pp. 36–44, Association for Computational Linguistics, 2010.

[68] R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls : Linking text sentiment to public opinion time series," 2010.

[69] X. Zhu, S. Kiritchenko, and S. Mohammad, "Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets," in *Proceedings of the 8th International*

Workshop on Semantic Evaluation (SemEval 2014), (Dublin, Ireland), pp. 443–447, Association for Computational Linguistics and Dublin City University, August 2014.

[70] M. Giménez, F. Pla, and L.-F. Hurtado, "Elirf: a support vector machine approach for sentiment analysis tasks in twitter at semeval-2015," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.

[71] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, A. Reyes, and J. Barnden, "Semeval-2015 task 11: Sentiment analysis of figurative language in twitter," in *Proceedings of Int. Workshop on Semantic Evaluation (SemEval-2015)*, 2015.

[72] R. Selvarajan and A. Ekbal, "Iitpatna: Supervised approach for sentiment analysis in twitter," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, (Dublin, Ireland), pp. 324–328, Association for Computational Linguistics and Dublin City University, August 2014.

[73] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets," in *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, (Atlanta, Georgia, USA), June 2013.

[74] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *2013 Conference on Empirical Methods in Natural Language Processing. Proceedings of the Conference*, (Stroudsburg), pp. 1631–1642, Association for computation Linguistics, 2013.

[75] W. Lehnert, C. Cardie, D. Fisher, E. Riloff, and R. Williams, "University of massachusetts: Description of the circus system as used for muc-3," in *Proceedings of the 3rd Conference on Message Understanding*, MUC3 '91, (Stroudsburg, PA, USA), pp. 223–233, Association for Computational Linguistics, 1991.

[76] W. Lehnert, C. Cardie, F. D., J. McCarthy, E. Riloff, and S. Soderland, "University of massachusetts: Description of the circus system as used for muc-4," in *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, MUC4 '92, (Stroudsburg, PA, USA), p. 282288, Association for Computational Linguistics, 1992.

[77] W. Lehnert, J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan, and S. Goldman, "Umass/hughes: Description of the circus system used for MUC-5," in *Proceedings of the 5th Conference on Message Understanding*, MUC5 '93, (Stroudsburg, PA, USA), pp. 277–291, Association for Computational Linguistics, 1993.

[78] J. F. McCarthy and W. G. Lehnert, "Using decision trees for coreference resolution," in *Proceedings of the fourteenth international joint conference on artificial intelligence*, pp. 1050–1055, 1995.

[79] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[80] W. M. Soon, H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Comput. Linguist.*, vol. 27, pp. 521–544, Dec. 2001.

[81] R. Pandya and J. Pandya, "C5.0 algorithm to improved decision tree with feature selection and reduced error pruning," *International Journal of Computer Applications*, vol. 117, pp. 18–21, May 2015.

[82] V. Ng and C. Cardie, "Improving machine learning approaches to coreference resolution," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, (Stroudsburg, PA, USA), pp. 104–111, Association for Computational Linguistics, 2002.

[83] A. Culotta, M. Wick, R. Hall, and A. Mccallum, "First-order probabilistic models for coreference resolution," in *Proceedings of HLT-NAACL 2007*, 2007.

[84] E. Cambria, R. Speer, C. Havasi, and A. Hussain, "Senticnet: A publicly available semantic resource for opinion mining.," in *AAAI Fall Symposium: Commonsense Knowledge*, vol. FS-10-02 of *AAAI Technical Report*, AAAI, 2010.

[85] E. Cambria and A. Hussain, *Sentic Computing: Techniques, Tools, and Applications*. Springer, 2012.

[86] R. Mihalcea and E. Faruque, "Senselearner: Minimally supervised word sense disambiguation for all words in open text," in *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text* (R. Mihalcea and P. Edmonds, eds.), (Barcelona, Spain), pp. 155–158, Association for Computational Linguistics, July 2004.

[87] R. Mihalcea and A. Csomai, "Senselearner: Word sense disambiguation for all words in unrestricted text," in *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, (Stroudsburg, PA, USA), pp. 53–56, Association for Computational Linguistics, 2005.

[88] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *International Conference on Learning Representations*, 2013.

[89] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pp. 448–453, 1995.

[90] J. Nogueras-Iso, F. J. Zarazaga-Soria, and P. R. Muro-Medrano, *Geographic Information Metadata for Spatial Data Infrastructures: Resources, Interoperability and Information Retrieval*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[91] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013.

[92] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley & Sons, Inc., 2008.

[93] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, (Stroudsburg, PA, USA), pp. 1003–1011, Association for Computational Linguistics, 2009.

[94] S. Thaseen and C. A. Kumar, "An analysis of supervised tree based classifiers for intrusion detection system," in *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*, pp. 294–299, Feb 2013.

[95] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.

[96] F. rup Nielsen, "A new anew: evaluation of a word list for sentiment analysis in microblogs," in *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages* (M. Rowe, M. Stankovic, A.-S. Dadzie, and M. Hardey, eds.), vol. 718 of *CEUR Workshop Proceedings*, pp. 93–98, May 2011.

[97] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language Resources and Evaluation*, vol. 39, no. 2, pp. 165–210, 2005.

[98] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining," in *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, (Valletta, Malta), European Language Resources Association (ELRA), May 2010.

[99] D. Arlia and M. Coppola, *Experiments in Parallel Clustering with DBSCAN*, pp. 326–331. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.

[100] Y. Kaewpitakkun and K. 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.

[101] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features, and experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, (Stroudsburg, PA, USA), pp. 42–47, Association for Computational Linguistics, 2011.

[102] S. M. Mohammad and P. D. Turney, "Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, (Stroudsburg, PA, USA), pp. 26–34, Association for Computational Linguistics, 2010.

[103] S. M. Mohammad and T. W. Yang, "Tracking sentiment in mail: How genders differ on emotional axes," in *Proceedings of the 2Nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '11, (Stroudsburg, PA, USA), pp. 70–79, Association for Computational Linguistics, 2011.

[104] S. M. Mohammad, "#emotional tweets," in *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, (Stroudsburg, PA, USA), pp. 246–255, Association for Computational Linguistics, 2012.

[105] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *Processing*, pp. 1–6, 2009.

[106] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Int. Res.*, vol. 37, pp. 141–188, Jan. 2010.

Publications

Journal Article

[1] **Tungthamthiti P.**, Shirai K.,and Mohd, M., "Recognition of Sarcasm in Microblogging Based on Sentiment Analysis and Coherence Identification", *Journal of Natural Language Processing* (under review). [Chapter 3]

Refereed Conference Papers

[2] **Tungthamthiti, P.**, Santus, E., Xu, H., Huang, C.-R., and Shirai, K. (2015). "Sentiment Analyzer with Rich Features for Ironic and Sarcastic Tweets." *In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pp. 178–187, 2015, October. [Chapter 4]

[3] **Tungthamthiti, P.**, Shirai, K., and Mohd, M. (2014). "Recognition of Sarcasms in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches." *In Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation*, pp. 404–413, 2014, December. [Chapter 3]