Automatic Emotion Identification from Text

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

by

Wenbo Wang
M.S., Beijing University of Posts and Telecommunications, 2008
B.S., Central South University, 2005

2015
Wright State University

Wright State University
GRADUATE SCHOOL

September 4, 2015

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Wenbo Wang ENTITLED Automatic Emotion Identification from Text BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

Amit P. Sheth, Ph.D.
Dissertation Director

Arthur A. Goshtasby, Ph.D.
Director, Department of Computer Science
and Engineering

Robert E. W. Fyffe, Ph.D.
Vice President for Research and
Dean of the Graduate School

Committee on Final Examination

Keke Chen, Ph.D.

Kevin Haas, M.S.

Krishnaprasad Thirunarayan, Ph.D.

Ramakanth Kavuluru, Ph.D.

ABSTRACT

Wang, Wenbo. Ph.D., Department of Computer Science and Engineering, Wright State University, 2015. *Automatic Emotion Identification from Text.*

Peoples emotions can be gleaned from their text using machine learning techniques to build models that exploit large self-labeled emotion data from social media. Further, the self-labeled emotion data can be effectively adapted to train emotion classifiers in different target domains where training data are sparse.

Emotions are both prevalent in and essential to most aspects of our lives. They influence our decision-making, affect our social relationships and shape our daily behavior. With the rapid growth of emotion-rich textual content, such as microblog posts, blog posts, and forum discussions, there is a growing need to develop algorithms and techniques for identifying people's emotions expressed in text. It has valuable implications for the studies of suicide prevention, employee productivity, well-being of people, customer relationship management, etc. However, emotion identification is quite challenging partly due to the following reasons: i) It is a multi-class classification problem that usually involves at least six basic emotions. Text describing an event or situation that causes the emotion can be devoid of explicit emotion-bearing words, thus the distinction between different emotions can be very subtle, which makes it difficult to glean emotions purely by keywords. ii) Manual annotation of emotion data by human experts is very labor-intensive and error-prone. iii) Existing labeled emotion datasets are relatively small, which fails to provide a comprehensive coverage of emotion-triggering events and situations.

This dissertation aims at understanding the emotion identification problem and developing general techniques to tackle the above challenges. First, to address the challenge of fine-grained emotion classification, we investigate a variety of lexical, syntactic,

knowledge-based, context-based and class-specific features, and show how much these features contribute to the performance of the machine learning classifiers. We also propose a method that automatically extracts syntactic patterns to build a rule-based classifier to improve the accuracy of identifying minority emotions. Second, to deal with the challenge of manual annotation, we leverage emotion hashtags to harvest Twitter 'big data' and collect millions of self-labeled emotion tweets, the labeling quality of which is further improved by filtering heuristics. We discover that the size of the training data plays an important role in emotion identification task as it provides a comprehensive coverage of different emotion-triggering events/situations. Further, the unigram and bigram features alone can achieve a performance that is competitive with the best performance of using a combination of ngram, knowledge-based and syntactic features. Third, to handle the paucity of the labeled emotion datasets in many domains, we seek to exploit the abundant self-labeled tweet collection to improve emotion identification in text from other domains, e.g., blog posts, fairy tales. We propose an effective data selection approach to iteratively select source data that are informative about the target domain, and use the selected data to enrich the target domain training data. Experimental results show that the proposed method outperforms the state-of-the-art domain adaptation techniques on datasets from four different domains including blog, experience, diary and fairy tales.

Finally, we apply the proposed research to analyze cursing, an emotion rich activity, on Twitter. We explore a set of questions that have been recognized as crucial for understanding cursing in offline communications by prior studies, including ubiquity, utility, contextual dependencies, and people factors.

Contents

List of Figures

List of Tables

Acknowledgment

A lot of sweet stories vividly show up in my mind when I am writing this acknowledgement, which fills my heart with love, happiness and thankfulness. These stories are so touching that I wish I could write down all of them here. I want to thank every person involved with these stories. I hope that each and every one of you knows how much I appreciate your help and that I would really like to return the favor in the future.

First and foremost, I would like to thank my advisor, Dr. Amit P. Sheth. I have been greatly inspired by his passion and enthusiasm towards pursuing research problems with real world impact. I am grateful to him for his constant support, guidance, and encouragement. Dr. Sheth helped me to understand how to think clearly about research problems by asking the "why" question with use-case scenarios and real world applications. I found this method very beneficial to approach research problems. He helped me not only in developing my research and technical skills but also soft skills such as communication skills and good work ethics, which would be helpful for rest of my career. I greatly benefited from his education philosophy of "learning how to learn."

I would like to express my gratitude to my dissertation committee members. I have gained a lot from Dr. Keke Chen's rich knowledge and research experience. He has also shown me how to explore a research problem from different potential directions. I still remember many advices that Kevin Haas gave me during my Yahoo! internship and benefited a lot from these advices, one of which is "more Eclipse, less PowerPoint." Dr. T.K. Prasad always provided positive and constructive feedback for my research. I always became a happier person every time after I met him. Dr. Ramakanth Kavuluru has been inspiring me with his sincere curiosity and love for tackling research problems and making methodological contributions. He has been encouraging me throughout the program: he sent me a congratulations email whenever I had a paper accepted.

I would like to thank my seniors who have been offering help to me since the first day I joined Kno.e.sis: Christopher Thomas, Pablo Mendes, Cartic Ramakrishnan, Karthik

Gomadam, Meenakshi Nagarajan, Ajith Ranabahu, Satya Sahoo and Prateek Jain. I am grateful to have Christopher as my mentor and elder brother. Christopher took me under his wings and I will never forget the countless hours that he spent to teach me how to solve research problems. Pablo has always believed in my talent and ideas, although I sometimes doubt about them. Cartic fervently offered me his help, even before I got the chance to ask for his help. With his hacking spirit, Karthik demonstrated me how boring Computer Science theories can be applied in our daily lives, which gave me a fresh perspective.

I would like to thank my friends at Kno.e.sis center, with whom I spent wonderful time and shared great moments together: Delroy Cameron, Ming Tan, Harshal Patni, Raghava Mutharaju, Ashutosh Jadhav, Pramod Anantharam, Pavan Kapanipathi, Vinh Nguyen, Hemant Purohit, Fengguang Tian, Ashwin Manjunatha, Kalpa Gunaratna, Sarasi Lalithsena, Sujan Perera, Alan Smith, Maryam Panahiazar, Sanjaya Wijeratne, Huiqi Xu, Pramod Koneru, Revathy Krishnamurthy, Shreyansh Bhatt, Swapnil Soni, Jeremy Brunn, Lu Zhou, Gaurish Anand, Farahnaz Golroo and Dan Vanuch. Delroy taught me how to make smooth transitions in paper writing. Ashutosh gave me emotional support and excellent suggestions when facing difficult situations. Pavan provided me critical and constructive feedback on my papers and presentations. Alan has a great gift to polish an idea and make it a far better one by brainstorming. Vinh demonstrated to me that our lives could stay simple if we stay simple. I would also like to thank Tonya Davis for her friendliness and help.

My basketball/football and BBQ buddies made my graduate life much happier: Tian Xia, Shaodan Zhai, Zhongliang Li, Jian Chen, and Lebin Lin. Our weekly activity was a very important way for me to vent out stress and pressure, and have more fun. Thanks to Zhuo Wang, Yanzhe Gao, Tammy Jean, and Shuai Wang, for the wonderful time that I spent with them.

My special thanks go to my family. My wife, Lu, is the most beautiful thing that ever happened to me. Without her, I would have never finished my PhD. I feel so fortunate to love her and being loved by her every day. I am grateful to be the son of my mother, Yanli.

She keeps giving me endless, unconditional support and love. She has shown me how to stay positive and strong even in the toughest time of life. To my parents-in-law, Guilan and Baoshan, both of you keep thinking what else you can do for me and Lu, but never ask anything for return. To my grandmother-in-law, Yuwen, you are the most awesome grandmother in the world and I wish I could be as awesome as you are when I grow old. To my grandparents, Keying and Kaichen, thank you for giving me a happy childhood and spoiling me as much as you could. I wish you could witness my graduation.

This material is based upon work supported by the National Science Foundation under Grant IIS-1111182 "SoCS: Collaborative Research: Social Media Enhanced Organizational Sensemaking in Emergency Response." Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Dedicated to

my wife Lu, and my mom Yanli

Introduction

"Your emotions are the slaves to your thoughts,

and you are the slave to your emotions."

-Elizabeth Gilbert

Emotions influence most aspects of our lives with or without our notice (Dolan, 2002). This makes emotion analysis very important in various problem settings. For example, emotions play a fundamental role in most aspects of work behavior (Briner, 1999): positive emotions can promote the job performance of employees (Staw et al., 1994; Cropanzano and Wright, 2001). Positive emotions can also sharpen people's thought-action processing skills and improve well-being over time (Fredrickson, 2001). Moreover, from the business point of view, emotions have a predictive power on customer satisfaction and recommendation, and hence can help companies gain more loyal customers (Walden and Dibeehi, 2012). The traditional way to assess people's emotions is to invite them to fill up some questionnaires (Watson et al., 1988), which usually takes a large amount of time and is limited to a small group of respondents. On the other hand, more and more users are sharing their life moments on social networking sites: in 2014, 74% of online adults used social networking sites (Pew Research Center, 2014). Access to their social media posts provides us with a valuable opportunity to analyze people's emotions in an unobtrusive and scalable way.

The Merriam-Webster Online Dictionary defines *emotion* as "a strong feeling (such

Table 1.1: People express their emotions in social media (Twitter) posts

Emotions	**Examples**
Anger	"I hate when my mom compares me to my friends"
Fear	"When I see a cop, no matter where I am or what I'm doing, I always feel like every law I've broken is stamped all over my body"
Joy	"Omg I finally fit into one pair of my jeans from last year!!"
Love	"iloveyou, just the way you are"
Sadness	"I hate when I get the hiccups in class"
Surprise	"Today's going a lot better than I thought on no sleep"

as love, anger, joy, hate, or fear)" (Merriam-Webster Online Dictionary, 2014). In this dissertation, we are interested in automatically detecting people's emotions embedded in their text (e.g., notes, blog posts, diaries and tweets). Table 1.1 shows a few Twitter posts in which people describe what has happened in their lives and convey various emotions.

Identifying people's expressed emotions from text is very challenging for at least the following reasons. First, we aim at fine-grained emotion identification that usually involves at least six different emotions. Emotions can be expressed in a subtle way and different emotions may not be differentiated by simple keywords. For example, although both the examples for *sadness* and example *anger* in Table 1.1 contain the same keyword "hate", they express two different emotions. Second, emotions can be implicit and triggered by specific events or situations: text describing an event or situation that causes the emotion can be devoid of explicit emotion-bearing words. We can infer emotion *fear* from the example *fear* in Table 1.1 because of "see a cop", although its author did not apply fear-related adjectives, e.g., fearful, scared, frightened, horrific, awful and formidable. Third, it is time-consuming and error-prone to create a labeled dataset for experiments: it takes time for annotators to decide the most appropriate label out of multiple emotion labels; annotators may misinterpret the emotion expressed by the author because nobody knows the exact emotional state of the author except the author himself/herself. Fourth, it requires a large labeled dataset so that assorted emotion-triggering events and situations (e.g., "hiccups in class", "see a cop", "mom compares me to my friends") can be covered, while existing

labeled emotion datasets are relatively small in many domains. To tackle these challenges, this dissertation will approach the problem of emotion identification from the following three synergistic aspects.

Feature analysis: The first question we address is which features are effective for supervised emotion identification? We used two domains as our test bed: suicide notes written by people who committed suicide (Wang et al., 2012a) and emotional tweets (Wang et al., 2012b). We experimented with a variety of features, including n-gram features, knowledge-based (sentiment/emotion lexicon) features, syntactic features and context features for the machine learning classifier. For suicide notes, on top of the supervised learning approach, we proposed an algorithm to automatically extract features from a large feature pool to build a simple but effective rule-based classifier. By combining the machine learning classifier and the rule-based classifier, the hybrid system achieved improved performance over both component classifiers.

Self-labeled data creation: The second question we study is how to automatically create a large labeled emotion dataset. This would be a game changer because most existing labeled datasets are relative small, due to the labor-intensive nature of annotating emotions. Therefore, they cover emotional moments in people's lives at a limited scale. We applied a set of emotion hashtags to filter emotional moments from big 'Twitter data' and designed simple heuristics to clean collected data and infer emotion labels out of these hashtags (Wang et al., 2012b) based on the emotion taxonomy from a psychology study (Shaver et al., 1987). Our proposed approach to create a labeled emotion dataset is different from most existing approaches in two ways: (1) the emotion labels are assigned by the tweet authors instead of annotators who may misinterpret the authors' emotions, and (2) the emotion labeling process is automatically conducted, which can be used to glean a large, labeled dataset with minimum efforts. When we kept increasing the size of the training data from 1,000 to about 2 million, the performances of two classifiers demonstrated steady improvements, which shows that the large size of the training data plays an important role

in improving the performance of emotion identification.

Domain adaptation for emotion identification: The third question we explore is how to leverage large, self-labeled Twitter data to improve emotion identification in target domains where there is a short supply of labeled data. We proposed an effective data selection approach to iteratively select source data that are informative about the target domain, and used the selected data to enrich the target domain training data. We proposed to measure the informativeness of a source instance using three factors: consistency, diversity, and similarity. Experiments on four datasets show that our approach performs effectively for cross-domain emotion identification and outperforms several baseline approaches.

Finally, we apply part of the proposed research to study the cursing behavior, an emotion-rich activity, on Twitter (Wang et al., 2014). Specifically, we identified a set of research theories on cursing in offline environments by prior studies and explored whether they would be supported if tested on Twitter. For example, we tested and confirmed the hypothesis that the main objective of cursing is to express negative emotions on Twitter. After detecting emotions (*anger*, *joy*, *love*, *sadness* and *thankfulness*) from both cursing and non-cursing tweets, we confirmed that cursing tweets express more negative emotions (*sadness* and *anger*) than non-cursing tweets.

Related Work

Emotion analysis has attracted increasing attention in recent years due to its applications in many areas such as workplace performance (Briner, 1999), well-being of people (Fredrickson, 2001), customer relationship management (Walden and Dibeehi, 2012), suicide prevention (Pestian et al., 2012), and depression detection (Choudhury et al., 2013b). One fundamental technique in many of these applications is emotion classification; that is, classifying textual units into different emotion categories such as *joy*, *anger*, and *fear*. In this chapter, we review relevant work on emotion identification and analysis. We start with introducing manually created emotion lexicons and approaches on how to automatically glean emotion lexicons from corpus in Section 2.1. Since it is time-consuming and error-prone for annotators to manually label emotion datasets, we will cover research efforts on how to create a self-labeled emotion dataset in Section 2.2. We will discuss rule-based emotion classification approaches in Section 2.3, supervised machine learning approaches in Section 2.4, and hybrid emotion classification approaches in Section 2.5. Finally, we talk about recent studies that unobtrusively analyze people's emotions (e.g., happiness, subjective well-being, depression) on social media, and introduce existing studies on, cursing, an emotion rich activity that remains largely unexplored on social media in Section 2.7.

2.1 Emotion Resources

Emotion lexicons and knowledge bases are important for many emotion analysis applications because they are not only used to make identification rules in unsupervised emotion classification systems but also used to derive effective features for supervised machine learning-based approaches. In this section, we describe several popular emotion lexicons and knowledge bases that may benefit the field of emotion analysis.

LIWC: Linguistic Inquiry and Word Count (Pennebaker et al., 2014) is a text analysis program that calculates how often people use words in some psychologically meaningful categories based on an built-in dictionary. The basic idea is to measure people's "various emotional, cognitive, structural, and process components" through text analysis. The latest dictionary contains about 4,500 words and word stems from more than 70 categories such as *1st person singular*, *past tense*, *family*, *money*, and *health*. The emotion-related categories (with example words following in parentheses) include: *affective processes* (cried, abandon), *positive emotion* (nice, sweet), *negative emotion* (ugly, nasty), *anxiety* (worried, fearful), *anger* (kill, annoyed) and *sadness* (grief, crying).

WordNet-Affect: WordNet-Affect (Strapparava and Valitutti, 2004) is a lexical resource that provides a hierarchy of "affective domain labels". 2,874 synsets and 4,787 words in WordNet (Miller, 1995) are annotated with the emotion labels from this hierarchy. For example, in the hierarchy, labels *positive-emotion*, *negative-emotion*, *neutral-emotion* and *ambiguous-emotion* belong to the parent label *emotion*; the labels *joy*, *love*, and *affection* belong to the parent label *positive-emotion*; the labels *amusement*, *elation*, and *gladness* belong to the parent label *joy*; the WordNet words "gladfulness", "gladness", and "gladsomeness" are labeled as *gladness* from the label hierarchy.

ANEW: Affective Norms for English Words (Bradley and Lang, 1999) provides standardized quantified metrics on how people rate English words from the aspects of emotions and attention. Specifically, there are 1,034 words labeled by a group of students on three dimensions: *affective valence* (from unpleasant to pleasant), *arousal* (from calm to excited)

and *dominance* (from dominated to in control). For example, the word "happy" is generally rated towards pleasant (valence), excited (arousal) and in control (dominance), while the word "murderer" is generally rated towards unpleasant (valence), excited (valence) and dominated (dominance).

Norms of valence, arousal, and dominance for English lemmas: Similar to ANEW, it provides ratings from the same three dimensions (*valence*, *arousal* and *dominance*) to a set of English words (Warriner et al., 2013), but it extends ANEW to 13,915 words that are rated using Amazon Mechanical Turk [1]. In addition, it provides demographic information about people who labeled these words, e.g., gender, age, and education.

ISEAR: International Survey on Emotion Antecedents and Reactions (Swiss Center for Affective Sciences, 2014) is a database containing events invoking any of the seven emotions: *joy*, *fear*, *anger*, *sadness*, *disgust*, *shame*, and *guilt*. Specifically, it contains answers of about 3,000 student respondents from 37 countries to the questionnaire of experienced emotion events. For example, what was the event about? When did it happen? How intense was the feeling?

EmoLex: EmoLex (Mohammad and Turney, 2010) is an emotion lexicon annotated using Amazon Mechanical Turk. In total, it contains about 14K words/phrases that are selected from: frequent unigrams (adjective, adverbs, nouns, verbs) and bigrams (adjective phrases, adverbs phrases, nouns phrases, verbs phrases) from the Macquarie Thesaurus (Bernard, 1984), positive/negative words from General Inquirer (Stone et al., 1966), and affective words in WordNet-Affect (Strapparava and Valitutti, 2004). Each word/phrase is annotated with 1 (yes) or 0 (no) based on whether it is associated with any of the eight emotions (*anger*, *fear*, *anticipation*, *trust*, *surprise*, *sadness*, *joy*, and *disgust*) and two sentiments (*negative* and *positive*). For example, the word "annoy" is associated with anger, disgust and negative sentiment.

[1] https://www.mturk.com/mturk/welcome. Amazon Mechanical Turk is a crowdsourcing platform where a large number of turkers follow predefined instructions to perform designed tasks (e.g., assigning valence scores to different words) at a relatively small price.

GI: General Inquirer (Stone et al., 1966) has 11,788 words that are labeled with tags from a number of categories. The following categories are very relevant for emotion analysis: *EMOT* (311 words related to emotions), *PosAff* (126 words with positive affect), *NegAff* (193 words with negative affect), *Pleasur* (168 words expressing joy), *Pain* (254 words conveying hardship), *Feel* (49 words indicating feelings), and *Arousal* (166 words expressing excitement).

CLex: Different from the above general purpose lexicons, Volkova et al. (2012) build a specific emotion lexicon for colors through crowdsourcing. Specifically, for a given image of a specific color, each Mechanical turker will be asked to describe the color, the emotion triggered by the color, the concepts associated with the color, for example: "brown-darkness-boredom; red-blood-anger."

NRC Hashtag Emotion Lexicon: Besides efforts to create an emotion lexicon by manually annotating words with related emotions, there are some studies trying to do this automatically. Mohammad (2012) uses emotion hashtags (e..g, #anger, #joy, #fear) to filter Twitter streaming data and collect more than 20,000 emotion tweets that are automatically labeled by the emotion hashtags they contain. The association between a word and an emotion is calculated by $PMI(word, emotion) - PMI(word, \overline{emotion})$, where $PMI(word, emotion)$ is the pointwise mutual information between a word and an emotion, and $PMI(word, \overline{emotion})$ is the pointwise mutual information between a word and the complement of the specified emotion. When the association is greater than zero, this word has a stronger association with the specified emotion than other emotions. Since it is created out of tweets, the generated lexicon contains informal language, such as: "ewwwwwww", "#perv."

This above-mentioned PMI-based approach is similar to (Turney, 2002), where the the polarity of a word (e.g., positive, negative) is determined by treating the entire Web as an underlying corpus. Yang et al. (2007b) apply a variant of PMI to a collection of blog sentences that are automatically labeled by emoticons, such as: ":)", ":(", ":D". Bandhakavi

et al. (2014) observe that not all the words in a tweet contribute to its emotion label, for example, in "a nice Sunday #joy," "nice" is a good indicator of emotion joy, while "Sunday" is not. To capture this intuition, they apply a language model-based approach that models each tweet as a mixture of emotion terms ("nice") and general terms ("a", "Sunday"). Experiments show that the quality of the extracted lexicon is higher than that of the lexicon extracted by a PMI-based approach.

Xu et al. (2010) apply an iterative framework that starts with a few seed words that have emotion label information. Then, they apply a graph-based algorithm to propagate the label information from seed words to new words, based on their similarities computed on multiple resources, such as: synonym dictionary, and unlabeled news corpus. To further ensure the lexicon quality, annotators will manually double check the newly inferred emotion labels during each iteration. Perrie et al. (2013) propose to infer word-emotion relations from a large Web corpus: Google N-gram corpus. It contains unigrams to 5-grams and their corresponding frequencies, calculated from a large collection of Web pages. N-grams with frequencies less than 40 times are not included in this corpus. The proposed approach uses 1,000 seed words whose emotion information is obtained from an emotion lexicon. For a target word whose emotion information is unknown, Perrie et al. (2013) make use of the emotion information of its surrounding words in all the tri-grams in Google N-gram corpus to infer this target word's emotion.

2.2 Self-labeled Emotion Data Creation

Many emotion classification techniques require annotated training data. Some studies employ human annotators to manually label text with emotions (Alm et al., 2005; Aman and Szpakowicz, 2007; Neviarouskaya et al., 2010; Roberts et al., 2012). However, manual annotation of emotions is usually labor-intensive, time-consuming, and error-prone, because of which there is a lack of large labeled datasets for emotion research. It is appealing to

explore methods that automatically create labeled emotion datasets.

A few studies investigate automatic ways of assigning emotion labels to documents (Mishne, 2005; Leshed and Kaye, 2006). When a user posts a blog on the LiveJournal website [2], one can specify a mood label from a predefined list of 132 moods (e.g., *amused*, *tired*, *sleepy*, *happy*) or create a new label that reflects his/her mood at the time of posting. This user-specified label is used as the mood label for the entire post. Since one post may contain a mixture of various emotions in different sentences, this document-level labeling is coarse-grained and cannot be applied to the sentence level. Moreover, while most of the labels (e.g., *happy*, *cheerful*) indicate people's emotional states, some labels (e.g., *awake*, *busy*) do not refer to specific emotions.

Another line of work is to automatically label emotions at the sentence level. Yang et al. (2007a) label blog sentences with emoticons used in these sentences. For example, the sentence "hah hah :) I am getting lucky" is labeled by emoticon ":)". Tokuhisa et al. (2008) exploit the sentence pattern "I was ** that ...", in which "**" and "..." refer to an emotion word and the clause that conveys the emotion, respectively. Then, the sentence is labeled with the emotion indicated by the emotion word accordingly. For example, from the sentence "I was disappointed that it suddenly started raining.", we know that "it suddenly started raining." conveys the emotion *disappointment* (Tokuhisa et al., 2008).

Several studies (Kouloumpis et al., 2011; Go et al., 2009; Pak and Paroubek, 2010; Davidov et al., 2010) leverage hashtags and emoticons in tweets to build training datasets for sentiment analysis. The basic idea is to collect tweets containing sentiment hashtags (e.g., "#sucks", "#notcute") or emoticons (e.g., ":)", ":D", ":-("), and label each tweet as positive or negative according to the polarity of hashtags and emoticons. However, studies that explore the potential of using Twitter data for emotion identification are rare so far. In this dissertation, we propose to automatically create a large emotion dataset with self-reported emotion labels from Twitter by leveraging hashtags, and study the construc-

[2] http://www.livejournal.com/

tion of accurate fine-grained emotion classifiers with the dataset. Choudhury et al. (2012) filter tweets via mood hashtags, and their study focuses on analyzing users' mood expressions through affective space (valence and activation). Another study (Mohammad, 2012) collects emotional tweets with six emotion hashtags (one hashtag per emotion), while our study is at a much larger scale in terms of both the number of emotion hashtags and the number of collected tweets. Instead of focusing on building an accurate emotion classifier, Purver and Battersby (2012) focus on the investigation of the difference between two datasets collected via emoticons and hashtags, respectively. They cross-validate classifiers by training them using one dataset and testing them on the other dataset. They find that the classifiers achieve good performance on some emotions (i.e., happiness, sadness and anger) and poor performance on other emotions (i.e., fear, surprise and disgust). While many studies follow the six basic emotion categories by Ekman (i.e., anger, disgust, fear, joy, sadness and surprise), Suttles and Ide (2013) apply Plutchik's eight basic bipolar emotion categories (i.e., joy vs. sadness, anger vs. fear, trust vs. disgust, and surprise vs. anticipation) to collect emotion tweets.

2.3 Rule-based Emotion Classification

Because of the intuitiveness and efficiency, several studies (Zhe and Boucouvalas, 2002; Liu et al., 2003; Chaumartin, 2007; Kempter et al., 2014) on emotion identification usually applied manually crafted rules to identify emotions from text. Many of them start with building a knowledge base, in which each word is assigned weights indicating different emotions. For every word in an input sentence, they look up the knowledge base to retrieve its weights. Then, a set of grammar rules is applied to infer the emotion of a phrase/-clause/sentence out of the weights of its component words. For example, since the word "happy" is labeled with a weight (range [0:1]) of 1.0 to indicate emotion *joy* in the knowledge base, the phrase "not happy" will have a weight of 1.0 to indicate emotion *sadness*

because of the negation word "not"; the phrase "very happy" will have a weight of 2.0 to indicate emotion *joy* because the adverb "very" intensifies the emotion *joy*.

One of the early works (Zhe and Boucouvalas, 2002) applies eight types of rules to identify emotions from instant messages, e.g., negation rule, intensify rule. Some of the rules are quite naive, e.g., if an affective word is negated, then the negation rule assumes that the corresponding sentence does not convey any emotion. Liu et al. (2003) use predefined unambiguous emotion words (e.g., "happy", "depression", "love") to retrieve sentences. Within the sentences, the emotion-indicating weights of these emotion words are propagated to other uncertain words. For example, from the sentence "Car accidents can be scary", it learns that "car accidents" can trigger emotion *scary*. Once it finishes populating emotion-indicating weights to new words, for a new sentence, it can apply the emotion-indicating weights of both unambiguous words and new words to decide the most likely emotion. Task 14 of SemEval-2007 (Strapparava and Mihalcea, 2007; Chaumartin, 2007; Andreevskaia and Bergler, 2007) was to annotate news headlines with emotions: 250 headlines for development and 1,000 headlines for testing purpose. Chaumartin (2007) utilizes a language parser to locate the *head word* (the root of the dependency tree) in a sentence and aggregates emotion-indicating weights of all the component words, with the weight of the head word multiplied by six because they believe the head word has a much higher influence among all the words in triggering emotions. Both (Chaumartin, 2007) and (Andreevskaia and Bergler, 2007) employ dependency tree to detect valence shifts, e.g., negation, increment, and decrement. After looking up emotion-indicating weights for each word in a sentence, Neviarouskaya et al. (2011) follow a set of detailed rules to propagate the weight from a word to a phrase and from a phrase to a sentence (in cases of a simple sentence, a compound sentence, a complex sentence and a complex-compound sentence).

While many rule-based approaches heavily rely on a predefined emotion lexicon resource, Sahlgren et al. (2007) take a "resource-poor but data-rich" approach: a positive point and a negative point are represented by vectors that consist of context words of eight

manually chosen positive and negative words from a news corpus. Then, they project news headlines into the same space and observe whether this vector is closer to the positive vector or negative vector by calculating corresponding cosine similarities. Instead of using two vectors, Danisman and Alpkocak (2008) use five vectors to represent five emotions and each vector is the mean of the vectors of the documents belonging to this emotion. Moreover, emotion lexicons, such as WordNet-Affect, are also employed. Strapparava and Mihalcea (2008) share a similar idea of comparing the news headline vector with different emotion vectors but in a low dimension space obtained by Latent Semantic Analysis technique. Moreover, it uses emotion words from WordNet and WordNet-Affect to define emotion vectors to achieve a better representation.

Rule-based algorithms are generally easy to interpret and debug. Since the rules are manually crafted, they usually provide good precisions and are suitable for solving problems where only a small amount of labeled data is available. However, the performance of rule-based systems is usually limited by the quality and coverage of the underlying emotion knowledge base. While it is relatively easy to craft rules to cover most of the cases in small datasets, improving the recall on large datasets remains a challenge.

2.4 Supervised Emotion Classification

Supervised machine learning algorithms require a sufficient amount of labeled data for training. The ever growing user-generated content describing people's emotional moments becomes an invaluable source for feeding these algorithms. One line of work (Leshed and Kaye, 2006; Mishne, 2005) is to identify the overall emotion of a blog post at the document level by utilizing the mood labels chosen by the authors at the time of writing these posts. Specifically, Leshed and Kaye (2006) apply an SVM classifier with standard information retrieval techniques, e.g., stop word removal, bag-of-words model, and TF-IDF weighting. Mishne (2005) retrieves labeled blog posts in the same fashion, and experiments

with more features, e.g., document length, polarity orientation of the post (i.e., the number of positive/negative words), PMI-IR (the pointwise mutual information between a word and a mood, calculated by utilizing the entire Web as the corpus (Turney, 2002)), and emoticons. However, the contribution of different types of feature is not demonstrated. Ni et al. (2007) tackle the problem of separating blog posts with rich emotions from ones with rich information by performing a binary classification. There are some studies on predicting the emotions that are triggered by different news articles (Lin et al., 2007; Bao et al., 2012).

Another line of work is to classify emotions at the sentence level. Yang et al. (2007a) collect blog sentences containing emoticons and employ emoticons to infer the emotion label of a sentence. Instead of separately predicting emotions of multiple sentences in the same post, they cast the problem as a structure prediction problem and predict the emotions of all the sentences at one time so that the previous sentence and the following sentence can be taken as context to improve the emotion prediction of the current sentence. Aman and Szpakowicz (2008) demonstrate the usefulness of features derived from two knowledge bases: WordNet-Affect (Strapparava and Valitutti, 2004) and Roget's Thesaurus (Jarmasz and Szpakowicz, 2001). Martineau et al. (2014) explore the use of active learning approach to improve the quality of the emotion annotation labels. Tokuhisa et al. (2008) apply a two-step approach for emotion classification: a positive/negative/neutral classification followed by further fine-grained emotion classification among positive and negative emotions. Besides blog posts, there are some works dealing with fairy tales (Alm et al., 2005) and news headlines (Katz et al., 2007). Alm et al. (2005) add many folk-tale specific features such as the thematic story type and the story progress (the position of the sentence in the whole story) to tailor the model to fairy tales. Katz et al. (2007) combine the outputs of three classifiers (Naive Bayes, Decision Lists, and Nearest Neighbor Cosine) to make joint decisions of the emotion categories of news headlines. Besides answering the question "what is the expressed emotion?" there are work on "who experienced the expressed emotion?"

(Mohammad et al., 2014), "what triggered the emotion?" (Chen et al., 2010; Mohammad et al., 2014), and how to identify emotions in real time (Janssens et al., 2013).

In this dissertation, we focus on how to collect a large self-labeled emotion tweet dataset and build accurate emotion classifiers out of it. Since the performance of the classification model largely depends on the quality of the labeled dataset, we first investigate the quality of emotion hashtags on Twitter, i.e., whether the emotion hashtags truly indicate the authors' emotional states, and design filtering heuristics to remove noisy tweets from the dataset. We study a comprehensive set of features for building the emotion classification models for tweets, and compare our findings with those of contemporary research on other types of textual content such as blog posts. Since our emotion classification is at a much larger scale compared with the prior emotion analysis work, we also explore the performance gain that can be achieved by increasing the size of training data.

2.5 Hybrid Emotion Classification

Hybrid approaches usually apply both a rule-based technique and a supervised machine learning technique as both techniques can complement each other to better identify emotions. Machine learning techniques are suitable and effective when there is a large amount of labeled training data; rule-based approaches are beneficial in expressing well-known aspects of emotion-expression (declarative knowledge) when the labeled data is so sparse that machine learning algorithms cannot effectively capture signals. Take the suicide notes dataset (Pestian et al., 2012), for example: there are 16 categories in total, with 13 emotion-related categories (e.g., *love*, *sorrow*, *anger*), and three other categories (*information*, *instruction* and *other*). While 80% of the instances in the labeled dataset belong to the most frequent six categories, the remaining categories make up the remaining small portion. Because of this imbalanced data distribution, one would apply machine learning techniques to detect the most frequent emotions and craft rules to identify infrequent emotions. Prior

studies (Yang et al., 2012; Sohn et al., 2012; Xu et al., 2012b) usually make use of existing emotion lexicons to define rules and manually go through the data to spot more emotion words to add new rules. To build machine learning classifiers, Yang et al. (2012) combine four classifiers (SVM, Maximum Entropy, Naive Bayes, and Conditional Random Field) to make a joint decision (any vote, majority vote, and combined vote) on the most frequent emotions. Sohn et al. (2012) apply majority votes to the decisions from an ensemble of five MNB classifiers (with different information gain thresholds) and nine RIPPER models (combinations of different pruning sizes and random seeds) to detect the most frequent emotions. Xu et al. (2012b) apply spanning n-gram features to detect frequent emotions, normal n-grams for information and instruction, and pattern matching for infrequent emotions.

In this dissertation, we also explore the hybrid approaches for emotion classification of suicide notes. Specifically, we propose an algorithm to automatically extract effective syntactic and lexical patterns from training examples to build the rule-based classifier, and investigate a variety of lexical, syntactic and knowledge-based features to build the machine learning classifier. The hybrid system outperforms both the rule-based classifier and the machine learning classifier.

2.6 Instance-based Domain Adaptation

Domain adaptation has attracted attention recently (Pan and Yang, 2010). Previous work along different lines includes techniques for domain adaptation via pivot features (Blitzer et al., 2006, 2007), regression-tree adaptation (Chen et al., 2008), feature alignment (Pan et al., 2010), dimensionality reduction (Pan et al., 2011), a simple mixture distribution model (Daumé III and Marcu, 2006), deep learning (Glorot et al., 2011), and hierarchical Bayesian prior (Finkel and Manning, 2009). Due to its being simple to understand and to interpret, we explore this problem from the instance adaptation perspective and hence we

will limit our attention to the instance-based approaches.

Jiang and Zhai (2007) train an adaptive classifier on the union of both source and target domain instances, where the target domain labeled data are assigned larger weights since they are more representative of the target domain. Dai et al. (2007) extend the AdaBoost algorithm to adjust the weights of training instances. Some studies (Jiang and Zhai, 2007; Xu et al., 2011) apply a classifier trained on target domain labeled data to identify "good" and "bad" instances from source data: a source instance is considered to be a good (or bad) one if it can be correctly (or incorrectly) classified by the classifier. We attempt to select *informative* instances from the incorrectly classified instances rather than correctly classified ones, because their being incorrectly classified may suggest that they contain knowledge that the target data lacks. Data selection has been frequently used in machine translation to select sentences that are very similar to these in target data (Hildebrand et al., 2005; Lü et al., 2007; Foster et al., 2010; Axelrod et al., 2011). By doing so, the formed training data can hopefully better match the target data in text contents. In this dissertation, we aim at leveraging self-labeled tweets to improve the emotion identification in target domains, e.g., blogs and diaries. We apply a bootstrapping framework to iteratively select informative tweets to enrich the target domain training data. We define the informativeness of a source instance using three factors: consistency, diversity, and similarity. Moreover, we find that selecting tweets that are similar to target data is not sufficient, because tweets with nearly identical content can have contradicting labels. This fact makes it necessary to check whether tweets contain consistent knowledge about the target data or not before being selected.

Adapting self-labeled emotion tweets for emotion identification in other domains remains largely unexplored. Mohammad (2012) applies the feature augmentation proposed in (Daumé, 2007) to make use of Twitter data to improve emotion identification from news headlines, but domain adaptation is not the main focus of their work. We systematically study the problem of domain adaptation for emotion analysis using a large number of self-

labeled emotion tweets.

2.7 Emotion Analysis on Social Media

With users generating more and more data on social media, the data becomes a valuable source for studying people's emotions in an unobtrusive way. There are many studies on public happiness and subjective well-being. Mihalcea and Liu (2006) use a LiveJournal blog corpus to study "happiness factors" and happy moments (e.g., time of day and day of week). Bollen et al. (2011) track six dimensions of mood (tension, depression, anger, vigor, fatigue, confusion) of the public via Twitter and find that major events and holidays in the physical world have "significant, immediate and highly specific effect" on the public mood. Schwartz et al. (2013) predict the life satisfaction in one county via tweets posted from that area. Similarly, Quercia et al. (2012) use tweets to track the subjective well-being of communities in London and find the emotion scores of tweets and the socio-economic well-being to be highly correlated. Kramer (2010) proposes calculating the positive and negative emotion scores from Facebook status updates, and aggregating them to measure the overall happiness of a nation. Dodds et al. (2011) study the temporal patterns of happiness and information on Twitter, and find interesting weekly and daily happiness patterns. Besides the study of positive emotions, there are studies focusing on negative emotions, such as depression and stress. Choudhury et al. (2013b) find that clues such as decreased social activity and increased negative emotions, contribute to the detection of a major depressive disorder in Twitter users. Park et al. (2015) discover that depression is correlated with the frequency and diversity of interactions on Facebook among young adults. Coppersmith et al. (2014) train a statistical classifier to identify military personnel who suffer from Post Traumatic Stress Disorder by leveraging language trails. There are also some studies on how people's emotions get affected due to changes in life. Choudhury et al. (2013a) show that there are significant changes in online activity and expressed emotions for about 15% of

the new mothers after their childbirth. Kramer et al. (2014) conduct a large-scale emotional contagion experiment and find that people's emotions can get affected by the positive and negative emotions expressed in their news feeds.

In contrast with the above studies, our work focus on an emotion-rich activity that has been examined extensively in the physical world but remains largely unexplored on social media: cursing. Cursing is not uncommon during conversations in the physical world: 0.5% to 0.7% of all the words we speak are curse words, given that 1% of all the words are first-person plural pronouns (e.g., we, us, our). On social media, people can instantly chat with friends without face-to-face interaction, usually in a more public fashion and broadly disseminated through a highly connected social network. Will these distinctive features of social media lead to a change in people's cursing behavior? Before answering the question, we first consider the previous research on cursing in offline communications, and organize them into four groups corresponding to the four questions we address in this study.

The Ubiquity of Cursing: Cursing is more common than people might think. Jay (1992, 2009b) find 70 curse words in an 11,609-word tape-recorded conversation of elementary school students and college students. In another study, Mehl and Pennebaker (2003) report that curse words occurred at a rate of 0.5% over two 2-day periods among undergraduate students, which may not seem significant except that the first person plural pronouns – words like *we*, *us*, and *our* – occurred at a 1.0% rate. They also find substantial differences among individuals regarding curse words usage: the word rates varied from a minimum of 0% to a maximum of 3.4%. Some recent studies suggest that people have been hearing and using profanity words more often than ever before (Associated Press, 2006). More (a 69% increase) and harsher curse words have been used in TV programs in 2010 compared to 2005 (Parents Television Council, 2010). Jay (1992) also finds that a few most frequently used words (e.g., *fuck*, *damn*, *hell*, and *shit*) account for most of the cursing expressions in conversations (a long tail phenomenon).

The Utility of Cursing: Cursing is not as negative or harmful as it may seem at first

glance. Prior studies (Jay, 1992, 2000; McEnery, 2006; Nasution and Rosa, 2012) suggest that the main reason that people use swearing words is to express some strong emotions, especially anger and frustration, for emphasis. As a common conversational practice, cursing rarely results in obvious harm. Only when cursing occurs in the form of insults toward others, such as name-calling, harassment, hate speech, and obscene telephone calls, it becomes harmful (Jay, 2009a). Researchers also find that other positive effects could be achieved by swearing. For example, Stephens et al. (2009) report that swearing increased pain tolerance and decreased perceived pain compared with not swearing. In addition, people may find relief and positive effects of laughing at jokes, humor and sarcasm in which curse words are used (Jay, 2000, 2009a).

Contextual Variables: Prior studies suggest that cursing is very sensitive to contextual influences (Jay, 2000). More specifically, people's propensity to curse, the particular curse words people use, and how others perceive the cursing behavior, are dependent on various contextual variables. Generally, the context of cursing activity is defined by those variables about *when*, *where*, *how*, *who* and *with whom* the cursing occurs. Among those variables, while the *who* and *with whom* variables have attracted the most attention from researchers, the physical location has also been recognized as important (Jay, 1992, 2000; Jay and Janschewitz, 2008). Jay and Janschewitz (2008) find that "people are more likely to swear in relaxed environments than in formal environments" (e.g., pub vs. office). Since such observations are made in the setting of oral communication in the physical world, it is not clear whether the physical location still matters for cursing on social media that occurs as written messages in the digital world. In addition, little is known about how the *when* and *how* factors may affect cursing. In this study, we examine the effect of location variable as well as the variables of time of day, day in a week, and message types, on cursing on social media.

Who Says Curse Words to Whom: There have been a considerable number of studies on understanding the characteristics of people who use and hear curse words. A set

of important variables have been identified and investigated (Jay, 1992, 2000; McEnery, 2006), including gender, age, race, religion and power. Unfortunately, many of these variables such as age, race and religion remain difficult to measure on Twitter, thus, we limit our focus on gender and power. Some well-recognized patterns about gender in swearing research include: (1) Gender affects cursing frequency. Many studies (Jay, 1980; Mehl and Pennebaker, 2003; Thelwall, 2008; Pilotti et al., 2012) suggest that men curse more frequently than women. (2) Gender affects the choice of curse words. For example, according to (McEnery, 2006), women use the words *god*, *hell*, *bitch*, and *piss* more often than men, and men use the words *fuck* and *cunt* more often than women. (3) People are more likely to curse in same-gender contexts than in mixed-gender contexts (Jay and Janschewitz, 2008; Pilotti et al., 2012). People's power or social rank also plays a role in cursing. McEnery (2006) finds that frequency of cursing is inversely proportional to the social rank.

Cursing on social Media: Only a few efforts have been made to explore cursing on social media. Thelwall (2008) studies the use of curse words in MySpace profiles and the effects of gender and age factors. Sood et al. (2012) investigate profanity usage in Yahoo! Buzz communities and found that different communities (e.g., politics or sports) use profanity with varied frequencies and in different ways. Turning to Twitter, Bak et al. (2012) study self-disclosure behavior in Twitter conversations. As one aspect of self-disclosure, profanity is used more frequently between users with higher relationship strength. While other researchers have mainly focused on investigating algorithms to automatically detect offensive tweets (Xiang et al., 2012; Xu et al., 2012a), our understanding of the basic questions regarding the use of offensive words on Twitter still remains unexplored, such as why people use curse words, who uses it, and whether these words are always harmful and should be removed. The insights gained in our study can shed light on these questions.

Emotion Classification

In this chapter, we study the problem of emotion identification by casting it as a classification problem. We investigate a variety of lexical, syntactic, knowledge-based, context-based features and show how their relative contributions to the performance of the supervised machine learning classifier when the training data are not sparse. We also propose an algorithm to automatically extract effective syntactic and lexical patterns from training data to build the rule-based classifier when training data are sparse. We evaluate the proposed approach in two different domains: suicide notes and Twitter data.

3.1 Overview

Previous text mining research has mostly focused on dealing with the factual aspects in the text, such as identifying entities (e.g., people's names, organizations), classifying articles based on whether the article discusses a given topic (e.g., sports, finance), and extracting relationships (e.g., is located at, was born in). More recently, increasing attention has been paid to the analysis of sentiments in subjective text. Such sentiment analysis is concerned about the polarity of people's opinions regarding certain entities (Chen et al., 2012a). For example, does the person like or dislike the restaurant mentioned in the text? Does the person support this presidential candidate or not (Chen et al., 2012b)? However, relatively few studies focus on our own emotional states. For example, is this person suffering from depression?

With the rise in the popularity of social networking sites, users interact with each other on these sites and a large amount of User Generated Content (UGC) is being produced: more than 500 million tweets are posted every day [1]. These social networking sites encourage users to record their thoughts and moments in life, e.g., Facebook ("what's on your mind?"), Twitter ("what's happening?"). Because of this, a lot of UGC naturally reflects people's emotional states, e.g., "when you talk to that person and she makes you smile. what a feeling. #happy " (a status update from Twitter) and "So excited for a brand new day. The future is so exciting!!!" (a post from Facebook). This provides an invaluable opportunity for social media analytics (Sheth et al., 2014), such as analyzing people's emotions in an unobtrusive way.

However, identifying the expressed emotions in text is very challenging for at least the following reasons. First, emotions can be implicit and triggered by specific events or situations. Text describing an event or situation that causes the emotion can be devoid of explicit emotion-bearing words. Consider the examples in Table 1.1: in the second example, fear is inferred from "see a cop"; in the first example, anger is inferred from "mom compares me to my friends"; and in the fifth example, embarrassment (a subcategory of sadness) is inferred from "hiccups in class". We can recognize fear from the text even though there is no explicit reference to words such as "scare" and "panic". Second, gleaning distinctions between different emotion categories purely on the basis of keywords can be very subtle. The first example and the fifth example in Table 1.1 have similar sentence patterns and contain the same emotion-bearing word "hate", but they belong to different emotion categories (i.e., *anger* and *sadness*).

We tackle the problem of emotion classification by studying the effectiveness of different features (lexical, syntactic, knowledge-based, context-based, and class-specific features) on two domains: suicide notes and Twitter data. For suicide notes, we find that a combination of lexical, knowledge-based, syntactic and class-specific features are effective

[1] https://blog.twitter.com/2013/new-tweets-per-second-record-and-how

for identifying majority emotion classes. For improving the identification of minority emotion classes, we propose an algorithm to automatically discover beneficial features that are ignored by the supervised classifier and to build a rule-based classifier consisting of these features. The experiments demonstrate that the final hybrid system, consisting of both the supervised classifier and the rule-based classifier, achieves a better performance than its component classifiers. For Twitter data, the combination of lexical, POS (part-of-speech) and knowledge-based features achieves the best accuracy; we also find that lexicon-based features become less important on large training data.

3.2 Problem Definition

For an input instance (i.e, a sentence or tweet) s, the problem of emotion classification is to decide the most appropriate emotion class $c_j \in C(1 \leq j \leq w)$ for the instance s, where $C = \{c_1, c_2, \ldots, c_w\}$ represents a set of candidate emotion classes. Because there are many different emotions with different granularities, the research community has not reached a consensus on the exact emotion classes. Ekman et al. (1972) propose six basic emotion classes: anger, disgust, fear, joy, sadness, and surprise. In this dissertation, the emotion classes may vary from one dataset to another, but they usually contain these basic emotion classes.

3.3 Methods

Supervised machine learning algorithms do not directly deal with input text; instead they make decisions based on the feature vector representation of each input text. Thus, how we abstract and represent input text using features is an important research question, which will be covered in Section 3.3.1. We show how to spot beneficial features that are ignored by the supervised machine learning algorithm and how to build the rule-based classifier

with these spotted features in Section 3.3.2.

3.3.1 Features for the Supervised Classifier

N-gram features: N-gram features are widely used in a variety of tasks in text classification, including emotion analysis. Despite its simplicity, unigrams (n=1) have been used for emotion classification effectively (Aman and Szpakowicz, 2008). Bigrams (n=2) and trigrams (n=3) capture context better than unigrams do (Tokuhisa et al., 2008). In our study, we experimented with unigrams, bigrams, trigrams and their combinations. Punctuation (e.g., !!, ??) and emoticons (e.g., :P, <3, </3) were also included in the n-gram model. Stop words elimination was not used because prior studies observe that stop words might be helpful in capturing the properties of some categories.

N-gram Position: Similar to (Pang et al., 2002), we also hypothesize that the words located towards the end of a sentence are more important than other words, because people usually summarize or highlight their points in the end. For example, "I hate it when stuff like that happens,.. thank god it worked out.<3 #thankful. ". Although "hate" appears in the first half of the sentence, the overall emotion is dominated by "thank" in the latter half. We encoded the position information into a feature by attaching a number (i.e, 1 or 2) to each n-gram to indicate whether it is in the first half or the second half of the instance. For example, if an instance has 10 unigrams, then the first 5 unigrams belong to the first half of the instance and are attached with 1. We also experimented with dividing an instance into three parts, but the results got worse.

Knowledge-based features: These are features based on prior knowledge about the subjectivity, sentiments, or semantic categories of English words. Specifically, we used MPQA (Wilson et al., 2005), LIWC (Pennebaker et al., 2014), and WordNet-Affect (Strapparava and Valitutti, 2004). MPQA is a subjective lexicon, which provides the sentiment polarities (positive/negative/neutral) and strength (strongsubj/weaksubj) of 8,211 words. For each input sentence, we count the numbers of positive, negative, neutral, strongsubj,

and weaksubj words according to MPQA as features. LIWC is a text analysis program with a built-in dictionary. For each piece of input text, the program outputs a vector with 69 dimensions covering positive/negative sentiments, casual words, numbers, etc. From WordNet-Affect, we collected the words from 32 direct subcategories of *positive-emotion* (e.g., joy, and love.), *negative-emotion* (e.g., anxiety, and sadness.), *neutral-emotion* (e.g., apathy.), and *ambiguous-emotion* (e.g., surprise.) defined in WordNet-Affect. For each instance, we used 32 features, each of which represents the number of words from one of the 32 subcategories.

Syntactic features: These are features based on syntactic information of the text, including dependency relation, POS tag, and sentence tense. We first apply the Stanford Parser (Klein and Manning, 2003) to parse each sentence and get corresponding collapsed dependencies. For each collapsed dependency d, we define the associated relation feature as $(d.name, d.gov, d.dep)$, where $d.name$ is the type of dependency, $d.gov$ is the stemmed governor token of dependency d, and $d.dep$ is the stemmed dependent token of dependency d. Take an artificial sentence "Please pay them," for example; we generate the following relation features: *(dep, please, pay)* and *(dobj, pay, them)*. Moreover, considering that some types of words (e.g., adverbs, adjectives, etc) are likely to convey sentiments, we obtain POS features by counting the numbers of words with the following POS tags: adjective (JJ/JJR/JJS), adverb (RB/RBR/RBS), noun (NN/NNS/NNP/NNPS), pronoun (PRP), present verb (VB/VBG/VBP/VBZ), past verb (VBD/VBN), and modal (MD). To explore whether there are associations between different categories and sentence tenses, we use the counts of different verb tenses in each sentence as features.

Context features: We hypothesize that the sentiments of the surrounding sentences may affect the sentiment of the current sentence. So we use the MPQA feature K_m and POS count feature S_p of the previous and next sentences. If the previous or the next sentence is missing, we set the corresponding features to 0.

Besides the above generic features which don't focus on a specific class, we propose

a few class-specific features targeting the *information* and *instruction* classes, which are special classes in the suicide notes dataset. Note that these features are sophisticated syntactic features. Sentences about the details of money, bank accounts and papers are labeled with *Information*, such as "You will find the keys and some of my papers and money in the side pocket of my brown coat in the closet." Sentences that ask families to perform specific tasks are labeled with *Instruction*, such as: "Teach them love and understanding & truth."

Information features: We observe that sentences indicating the location of property are more likely to be labeled as *information*. For example, "my/PRP$ books/NNS **are/**VBP up/**RP** under/**IN** the/**DT** cash/**NN**." is labeled with *Information*. One feature we can use is the frequency of the sequence that the word "is" or "are" is followed by zero or one particle (*RP*) or adverb (*RB*), a location preposition (e.g., *in*, *at*, *above*, and *under*.), zero or one determiner (*DT*), and a noun (*NN/NNS/NNP/NNPS*). Similarly, another feature can be the frequency of the sequence in which a noun is followed by a location preposition (*IN*), zero or one determiner (*DT*), and another noun. For example, "$/$ 100/CD in/IN travelers/NNS checks/VBZ and/CC check/VBP book/**NN** in/**IN** glove/**NN** compartment/NN ./."

Instruction features: We also observe that sentences that ask other people to do something, or to give something to someone, are usually labeled as *instructions*. To verify the observation, we sort the subject, direct object and indirect object of an action into three types: the writer himself/herself (e.g., *I*, *me*, *myself*), other people (e.g., *NNP*, *PRP*, *wife*, and *brother*.) and anything else, and count the frequency of each type as a feature. More specifically, we take the governor of nominal subject relation (*nsubj*) as the subject, the dependent of direct object relation (*dobj*) as the direct object, and the dependent of to-prepositional-modifier relation (*prep_to*) and indirect object relation (*iobj*) as the indirect object. For example, in the sentence "John J. Johnson please notify my wife at 3333 Burnet Ave. Tel.", there are the relations *nsubj(please, Johnson)* and *dobj(notify, wife)*, in which the subject of the verb "please" is "Johnson" (other people), and the object of the verb "notify" is "wife" (other people). In the sentence "All my fortune will go to Pat Johnson," there

exists the to-preposition-modifier relation *prep_to(go, Johnson)*, and the indirect object is "Johnson" (other people).

3.3.2 Constructing the Rule-based Classifier

Supervised machine learning classifiers may ignore features that are beneficial but infrequent for minority emotion classes. To compensate for this, we developed a rule-based classifier that leverages lexical and syntactic patterns to detect minority emotions from sentences. Manually constructing such a set of lexical and syntactic patterns in different categories can be laborious and time-consuming, especially when the patterns should be collected for many categories. Therefore, we propose an algorithm to automatically extract patterns from the training data set.

Let $P = \{p_1, p_2, \ldots, p_n\}$ be the set of patterns, which will be used by the rule-based classifier, and $C = \{c_1, c_2, \ldots, c_w\}$ be the set of categories (excluding neutral categories). Later, we will define the *g-measure* of pattern p_i with respect to category c_j, denoted $g(p_i, c_j)$ ($0 \leq g(p_i, c_j) \leq 1$). Intuitively, a higher value of $g(p_i, c_j)$ indicates that the sentence containing p_i is more likely to belong to c_j. Our algorithm takes the training data set as input and outputs the pattern set P and corresponding values of *g-measure* for pattern-category pairs.

The algorithm starts by extracting candidate patterns, which include (1) n-grams up to length four (e.g., "leave my", "i can not face"), (2) consecutive POS tags up to length five (e.g., *JJ NNP NN*, *VBP TO VB VBN*), (3) dependency relations (e.g., *cop(cause,be)*, *nsubj(burden,i)*), and (4) generalized dependency relations (e.g., *prep_in(put,place)* $\rightarrow$ *prep_in(put,∗)*, *prep_in(∗,place)*).

After generating all candidate patterns, the χ^2 test is applied to reduce pattern search space. The χ^2 test is a commonly used method for feature selection in machine learning. More specifically, for each category in C, we calculate χ^2 scores (Yang and Pedersen,

Table 3.1: The number of sentences in different categories

		Contains pattern p'?	
		yes	no
Belongs to category c?	yes	N_a	N_c
	no	N_b	N_d

1997) for each candidate pattern p' as follows:

$$\chi^2(p', c) = \frac{(N_a + N_b + N_c + N_d)(N_a N_d - N_b N_c)^2}{(N_a + N_c)(N_b + N_d)(N_a + N_b)(N_c + N_d)} \tag{3.1}$$

where c is a category ($c \in C$), N_a is the number of sentences that belong to category c and contain p', N_b is the number of sentences that do not belong to category c and contain p', N_c is the number of sentences that belong to category c but do not contain p', and N_d is the number of sentences that do not belong to category c or contain p' (refer to Table 3.1). For each category, the candidate patterns are sorted in descending order according to their χ^2 scores. Overall we get one sorted list of patterns for each emotion category. We keep only the top M patterns in each list, and put these top patterns into a pattern set P. Since the same pattern can be included in different lists, we have: $|P| \leq M(|C| - 1)$, where -1 stands for the exclusion of neutral emotion class. Here, we set $M = 1000$.

In the following step, the algorithm estimates the value of $g(p_i, c_j)$. As discussed before, the higher $g(p_i, c_j)$ suggests that a sentence containing the pattern p_i is more likely to belong to category c_j. Following this intuition, we take the conditional probability $p(c_j|p_i)$ as the *g-measure*:

$$g(p', c) = p(c|p') = \frac{N_a}{N_a + N_b} \tag{3.2}$$

We remove patterns with *g-measure* values equal to 1.0 from P, because we observe that χ^2 is biased towards rare patterns that, by chance, co-occur with the same category in a few sentences, and are not strong indicators of that category. Note that we do not use χ^2

score as the *g-measure*, because it does not match our requirement of the *g-measure*. Unlike the *g-measure*, χ^2 score evaluates the usefulness of a pattern for classification. Intuitively, a pattern p gets a high χ^2 score with respect to a given category c in two cases: a) its presence is associated with the presence of the category (i.e., high N_a in Equation 3.1) or b) its absence is associated with the absence of the category (i.e., high N_d in Equation 3.1). In our case, the value of the *g-measure* is not related to N_d.

Based on the pattern set P and the *g-measure* values, the rule-based classifier is created for the multi-class classification of the sentences in suicide notes. The general idea is that a sentence s is assigned to category c if there is a pattern p present in s and $g(p, c)$ is the highest among the values of all patterns in s with any categories. We use a threshold τ to tune the performance of the classifier. The sentence s is labeled as category c only if $g(p, c) > \tau$. Otherwise, s is not classified into any category. We investigate the effect of varying values of τ through experiments.

3.4 Experiments

In this section, we present and discuss the experimental results on suicide notes and Twitter data.

3.4.1 Experiments on Suicide Notes

There are a total of 900 suicide notes, 600 of which were used as the training set, and the other 300 notes were used for testing. Each note was manually annotated at the sentence level. The annotation schema consists of 15 categories, among which 13 categories are emotion-related, including *abuse*, *anger*, *blame*, *fear*, *forgiveness*, *guilt*, *happiness-peacefulness*, *hopefulness*, *hopelessness*, *love*, *pride*, *sorrow*, and *thankfulness*, and the remaining two categories are *information* and *instructions*. Each sentence can have a single

label, multiple labels, or not have any label. Note that there are actually 16 categories, if we consider "no annotation" (i.e., do not belong to any of the 15 categories) as one category. The classification results were evaluated using micro-averaged F-measure.

Our preprocessing serves two purposes: (1) to normalize the input text so that the language parser can achieve a higher accuracy, and (2) to make generalization over raw text so that syntactically different but semantically similar signals can be aggregated. For (1), we corrected misspellings (e.g., "buth" $\Rightarrow$ "but"), replaced symbols with their formal expressions (e.g., "+" $\Rightarrow$ "and"), and normalized various forms of expressions (e.g., "couldn't, couldnt" $\Rightarrow$ "could not"). For (2), we applied regular expressions to replace phrases denoting money (e.g., "$ 1,000.00", "$ 147.00"), phone number (e.g., "513-636-4900", "6362051"), name (e.g., "John", "Bill") and address (e.g., "burnet ave") with "type" symbols "$MONEY$", "937-888-8888", "NAME" and "ADDRESS_SYMBOL " respectively. Take phone numbers for example, what matters is whether a phrase refers to a phone number or not, rather than the specific digits in the number.

The Machine Learning Classifier: SVM is an off-the-shelf supervised learning approach that has been shown to be highly effective for text classification. SVM maps input vectors into a higher dimension space by a kernel function and then draws a separating hyperplane to maximize the margin distance between the plane and the nearest vectors. We used LIBSVM (Chang and Lin, 2011), an open source SVM implementation, which supports multi-class classification by applying a "one-against-all" approach. We chose Radial Basis Function (RBF) as the kernel function, and we applied a grid search script in the LIBSVM package to find the optimal values for parameters C and γ. The main idea is to list different value combinations of C and γ, and choose the combination with the highest performance. The original evaluation metrics for performance in LIBSVM was changed from accuracy to micro-averaged F-measure. We applied the MIT Java WordNet Interface[2] for stemming. In addition, we required the minimum occurrence for each n-gram feature

[2]http://projects.csail.mit.edu/jwi/

Table 3.2: Candidate feature notations

N-gram Features	**Notation-N**
unigram	N_u
bigram	N_b
trigram	N_t
Knowledge-based Features	**Notatin-K**
the numbers of strongsubj, weaksubj, positive, negative and neutral words regarding MPQA	K_m
feature vector generated by LIWC software	K_l
Syntactic Features	**Notation-S**
collapsed dependency relations by Stanford Parser	S_d
the numbers of adjectives, adverbs, nouns, pronouns, present verbs, past verbs and modals	S_p
the numbers of different verb tenses	S_t
Context Features	**Notation-CO**
K_m of the previous and the next sentences	CO_m
S_p of the previous and the next sentences	CO_p
Class-specific Features (InFormation Features)	**Notation-F**
the numbers of two types of location phrases	F
Class-specific Features (InstRuction Features)	**Notation-R**
whether POSs of the first two words are VB/VBZ respectively the numbers of subjects that are the writer, other people, and anything else respectively the numbers of direct objects that are the writer, other people, and anything else respectively the numbers of indirect objects that are the writer, other people, and anything else respectively	R

to be equal to or greater than 3. A variety of features used by the classifier are divided into the following groups in Table 3.2.

We first conducted experiments using the SVM classifier and the rule-based classifier separately, and then examined the performance of the hybrid classifier created by combining both SVM and the rule-based classifiers. We first trained all the classifiers on the training dataset and then applied them to the testing dataset. All the results below were obtained from 300 testing suicide notes.

Evaluation of the Machine Learning Classifier: All the experiments were done using 5-fold cross validation. Our baseline method is an SVM classifier using unigrams

only. Table 3.3 gives the results of the SVM classifier using different feature combinations. Since there are many different features, we applied a greedy approach to find an optimal feature combination. We started by combining features in an n-gram category and found the optimal n-gram feature combination. Then, based on this optimal feature combination, we incorporated features from the next category, and searched for a new feature combination with a better result. We repeated the above procedures until all the feature categories had been explored. For each feature category in Table 3.3, we highlight the best feature combination if its performance is better than the best one in the previous feature category.

Applying selected features from n-gram, knowledge-based, syntactic, and class-specific feature categories, we got the best micro-averaged F-measure, the best recall and the second best precision. The best F-measure, 0.4883, is 3.9% higher than the F-measure of the baseline using only unigrams. More specifically, we want to analyze the utility of different features. For n-gram features, the combination of unigrams and bigrams gets an F-measure of 0.4707, while adding trigrams decreases the F-measure to 0.4542. For knowledge-based features, it is interesting to see that MPQA or LIWC features alone decreases the performance, but applying both of them together increases the performance by 0.43%. Among individual syntactic features, adding sentence tense features increases F-measure by 0.48%, which verifies that sentence tense features are useful for differentiating different categories. It is surprising that adding context features does not improve the result, which may suggest that it is not sufficient to capture context with only MPQA features K_m and POS features S_p of the previous and next sentences. Applying class-specific features for *instructions* improves the F-measure by 0.65%, which shows that sophisticated syntactic features like different types of subjects, direct objects, and indirect objects can be effective.

In summary, the SVM classifier achieves the best performance by applying ngram (unigram and bigram), knowledge-based (MPQA and LIWC), syntactic (POS count and verb tense count), and class-specific (information and instruction) features.

Evaluation of the Rule-based Classifier: Following the approach described in Sec-

Table 3.3: Performance of the SVM classifier with different feature combinations on the testing data

	Feature Set	**Micro-averaged F-measure**	**Precision**	**Recall**
N-gram Feature	N_u	0.4492	0.5971	0.3601
	N_u+N_b	**0.4707**	0.6505	0.3687
	N_u+N_b+N_t	0.4542	0.6128	0.3609
Knowledge-based Features	N_u+N_b+K_m	0.4623	0.5946	0.3781
	N_u+N_b+K_l	0.4650	0.6161	0.3734
	N_u+N_b+K_m+K_l	**0.4750**	0.6525	0.3734
Syntactic Features	N_u+N_b+K_m+K_l+S_d	0.4781	0.6667	0.3726
	N_u+N_b+K_m+K_l+S_p	0.4783	0.6553	0.3766
	N_u+N_b+K_m+K_l+S_t	0.4798	0.6584	0.3774
	N_u+N_b+K_m+K_l+S_t+S_p	**0.4818**	0.6612	0.3789
	N_u+N_b+K_m+K_l+S_t+S_p+S_d	0.4804	0.6657	0.3758
Context Features	N_u+N_b+K_m+K_l+S_t+S_p+CO_m	0.4697	0.6218	0.3774
	N_u+N_b+K_m+K_l+S_t+S_p+CO_p	0.4758	0.6508	0.3750
	N_u+N_b+K_m+K_l+S_t+S_p+CO_m+CO_p	0.4787	0.6593	0.3758
Class-specific	N_u+N_b+K_m+K_l+S_t+S_p+R	**0.4883**	0.6667	**0.3852**
	N_u+N_b+K_m+K_l+S_t+S_p+R+F	**0.4878**	**0.6694**	0.3837
All	All features	0.4720	0.6279	0.3781

tion 3.3.2, we first generated more than $50K$ candidate patterns extracted from training data, including: n-grams up to length four, consecutive POS tags up to length five, dependency relations, and generalized dependency relations. It's important to mention that we intentionally differentiated the patterns used by the rule-based classifier and the ones by the SVM classifier. The reason is that the performance of applying all the features is worse than that of applying a carefully-selected subset of features for SVM.

We studied the effect of varying values of the threshold τ ($0 \leq \tau \leq 1$) on the performance of the rule-based classifier. The classifier finds a pattern p and a category c for an input sentence using the algorithm described earlier, and assigns the label of c to the sentence only if $g(p, c) > \tau$. Figures 3.1 and 3.2 show the precision-recall curve and the micro-averaged F-measure of the results, respectively.

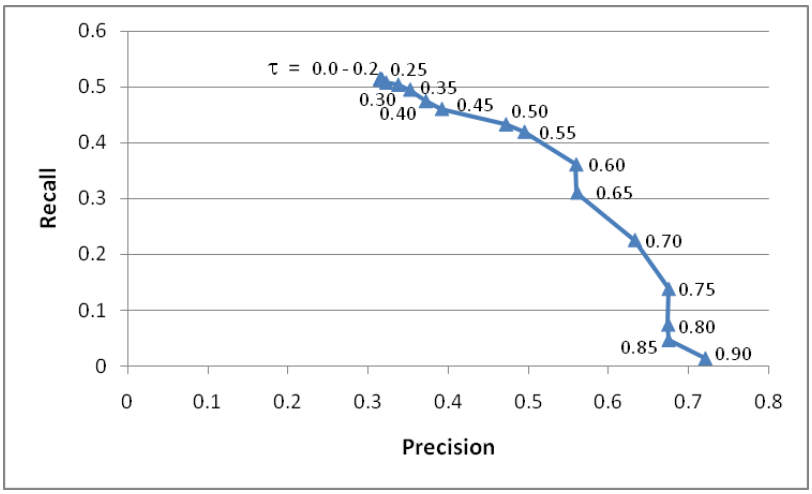

Figure 3.1: Precision-recall curve of the rule-based classifier with varying threshold τ on the testing data

According to Figure 3.1, the precision increases and the recall decreases with the threshold τ increasing. It is because the increased threshold leads to fewer patterns with higher quality being used for classification, and as a result, this raises the precision while brings down the recall. Figure 3.2 shows that the F-measure improves as the threshold τ increases from 0 to 0.55, and the best F-measure 0.4536 is achieved at τ=0.55. Note that

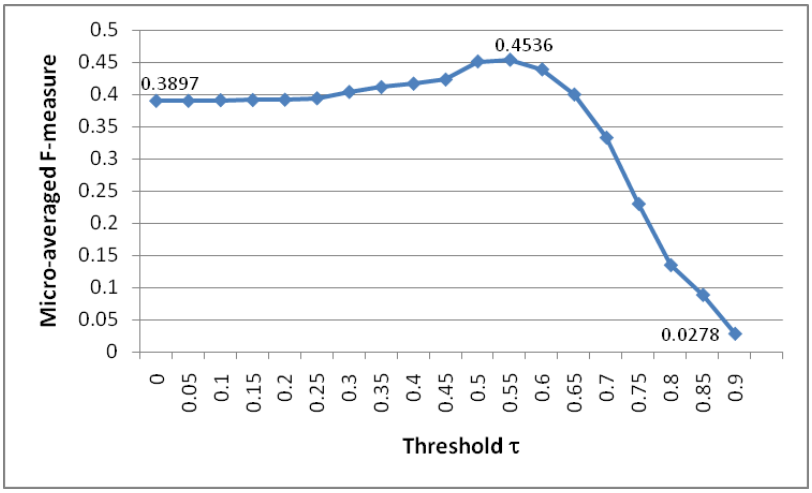

Figure 3.2: F-measure of the rule-based classifier with varying threshold τ on the testing data

it outperforms the machine learning baseline (0.4492). However, when we keep increasing the threshold, the F-measure goes down. It can be explained by the precision-recall curve in Figure 3.1, from which we can see that the precision rises faster than recall falls until the threshold reaches 0.55, and after that recall decreases faster than precision increases. Note that when τ is decreased to 0, the classifier still achieves the F-measure as 0.3897, which verifies that χ^2 test is effective to select patterns.

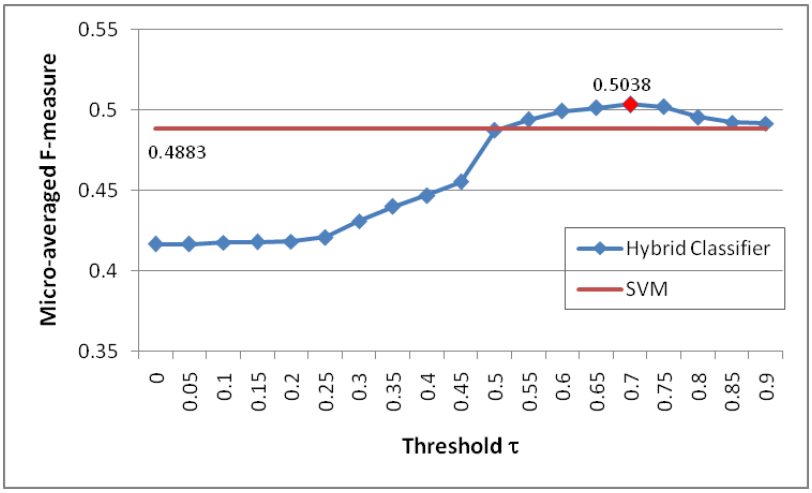

Figure 3.3: F-measure of the combined classifier on the test data

Evaluation of the Hybrid Classifier: A hybrid classifier was created by combining

the SVM classifier and the rule-based classifier. Since the SVM classifier exhibited the property of a relatively high precision and a low recall, we considered using the rule-based classifier to improve the recall. Following this idea, we applied a simple combination algorithm. Each sentence was fed to both the SVM classifier and the rule-based classifier to get the judgements respectively. If a sentence is assigned the label of any of the 15 categories by the SVM classifier, we keep the label; otherwise, we accept the label given by the rule-based classifier. For example, if a sentence s_1 is labeled as *love* by the SVM classifier, as a result, no matter what the label is given by the rule-based classifier, s_1 is classified into the category *love*. A sentence s_2 is not classified into any of the 15 categories according to the SVM classifier, but it is labeled as *guilt* by the rule-based classifier, consequently, the final label of s_2 is *guilt*.

We combined the SVM classifier that got the best result in the previous experiments with different rule-based classifiers tuned by the threshold τ. Figure 3.3 shows the results in terms of the micro-averaged F-measure. Observing the figure, we can see that the hybrid classifier outperforms the SVM classifier, when $\tau \geq$0.55. The best F-measure achieved by the hybrid classifier is 0.5038 at the point τ=0.7, which is 1.55% higher than the best F-measure achieved by the SVM classifier.

3.4.2 Experiments on Twitter Data

In this section, we explore which features are effective for emotion identification in a large Twitter dataset. Since both sentiment and emotion are subjective information, we also experiment with features that are known for sentiment analysis. The comparative study of the useful features for identifying sentiments and emotions may provide us a better understanding of emotion identification.

The Twitter data was collected by filtering a tweet stream with emotion hashtags and the hashtags were used to infer the emotion label of each tweet (Details will be explained in Section 4.3). It consisted of 248,898 tweets for training (**Tr1**) and 250,000 tweets for

testing (**Te**). Each tweet was labeled with the following emotions: anger, fear, joy, love, sadness, surprise and thankfulness. We lower-cased all the words; replaced user mentions (e.g., @ladygaga) with @user to anonymize users; replaced letters/punctuation that was repeated more than twice with the same two letters/punctuation (e.g., "cooool" → "cool", "!!!!!" → "!!"); normalized some frequently used informal expressions (e.g., "ll" → "will", "dnt" → "do not"); and stripped hash symbols ("#tomorrow" → "tomorrow").

We tried several classifiers, including C4.5 and k-nearest neighbors, and found that most of the classifiers either take too much time to train models on our large training dataset or consume more RAM than a single machine has. Finally we selected LIBLINEAR (Fan et al., 2008) and Naive Bayes (NB) to use, since they are very efficient even when handling a large number of tweets. LIBLINEAR is an open source machine learning library for large-scale, linear classification, and here we use its logistic regression branch to enable probability estimation. NB is a frequently adopted classifier for text classification and sentiment analysis (Pang et al., 2002). We employed Weka's implementations (Hall et al., 2009) for NB. We used default values for all parameters in both classifiers, and evaluated the quality of multi-emotion classification in terms of accuracy.

We trained LIBLINEAR and NB classifiers on **Tr1**, and applied the classifiers to the test dataset **Te**. Table 3.4 shows the classification accuracy achieved by different feature combinations. Note that it is a multi-class classification and each tweet is classified into one of the seven emotion categories.

N-gram features: We used only n-grams that appear in at least five different tweets. Since a tweet is limited to 140 characters, and most n-grams appear only once in a tweet, it does not make a big difference whether we use boolean features or counts to represent n-grams. Thus we have chosen to use a boolean feature for each n-gram, which is set to true if and only if the n-gram is present in the tweet. For n-gram features, as we increase n (n=1,2,3), higher order n-grams are considered to better capture the context information, but will that lead to better results? As shown in line 3-5 in Table 3.4, the best accuracy of

NB classifier is achieved by bigram features as 58.53%, followed by 57.75% with unigrams and 51.86% with trigrams. For the LIBLINEAR classifier, the accuracy of using unigrams (60.31%) is better than that of using any n-grams (n=1,2,3) with the NB classifier, but as we increase the value of n, the accuracies steadily decrease to 57.68% for bigrams and 50.65% for trigrams.

In addition, we also experimented with combined n-gram features. Although unigrams, bigrams and trigrams are not conditionally independent of each other, which violates the conditional-independence assumptions of NB, the NB classifiers using the combinations of them beat the ones that use only unigrams, bigrams, or trigrams. Specifically, combining unigrams and bigrams (line 5) increases the accuracy to 61.13%, and further incorporation of trigrams (line 6) slightly decreases the accuracy to 60.96%, which is still better than the accuracy for using one of them alone. We observed a similar pattern for LIBLINEAR classifiers. The best accuracy of 61.56% is obtained by LIBLINEAR classifiers using unigrams and bigrams, which is slightly better than that achieved by NB classifiers (61.13%).

Our experimental results show that combining unigrams and bigrams yields better performance than using unigrams alone. This is different from existing discoveries on sentiment classification (Pang et al., 2002), where using unigrams alone is better than applying either bigrams or a combination of unigrams and bigrams. It might suggest that bigrams are effective at capturing context information in our settings, which leads to the performance gain. Trigrams do not show such effectiveness. The previous research on emotion classification does not provide comparative study of different orders of n-grams. Aman and Szpakowicz (2008) use only unigram features, and Tokuhisa et al. (2008) use a combination of unigrams, bigrams, and trigrams for classification, but without comparing it with the classifiers that use unigrams, bigrams or trigrams alone, or other combinations.

N-gram Position: We appended unigrams and bigrams (the best feature combination so far) with their position information indicating whether they are in the first half or the

second half of a tweet. Contrary to our intuitions, the accuracy (line 7) of injecting position information gets slightly worse with both the NB and LIBLINEAR classifiers. In fact, injecting position information into n-grams is also found to decrease the performance in sentiment classification (Pang et al., 2002).

Knowledge-based features: We applied three sources of knowledge: LIWC, MPQA, and WordNet-Affect. For LIWC features, we collected emotion words from positive emotion category (408 words) and negative emotion category (499 words) in the LIWC2007 dictionary. For each tweet, we counted the number of positive/negative words based on the set of collected emotion words, and used the percentage of words that are positive and that are negative as features. In a similar way as we obtained LIWC features, we got the percentage of words with positive/negative polarity in a tweet as features using the MPQA lexicon. Similarly, we counted the number of words occurring in 32 emotion subcategories from WordNet-Affect as features. Adding knowledge-based features does not greatly improve the performance (see lines 8-10 in Table 3.4) in our settings of emotion identification. In contrast, lexicons have been widely used and shown to be effective in sentiment analysis (Pang and Lee, 2008). We suspect that emotions can be expressed implicitly and subtly. The intuition we have is that people tend to use sentiment words (positive or negative) when they talk about their likes and dislikes, but may not always share their emotions through specific emotion words. We also suspect that our self-labeled emotion dataset is so large that it implicitly contains most of the knowledge in three knowledge sources.

Syntactic Features: In one early sentiment classification study (Pang et al., 2002), an NB classifier is reported to achieve an accuracy of 77% using only adjective features, which is very close to the performance of using bigrams (77.3%), and not far from the accuracy resulting from using unigrams (81%). But the situation is different for emotion identification. Line 1 in Table 3.4 shows that the accuracy for using only adjectives as features (34.74% and 35.03% with NB and LIBLINEAR classifiers, respectively) is only about 60% of that for using unigrams (57.75% and 60.31% with NB and LIBLINEAR

Table 3.4: Accuracies of NB and LIBLINEAR on Tr1 dataset with different feature sets: boolean value (presence) is used for all n-gram features; percentages were used for LIWC, MPQA and POS features; frequency (counts) were used for WordNet-Affect features

#	Features	Accuracy(%)	
		NB	LIBLINEAR
1	adjective	34.74	35.03
2	n-gram(n=1)	57.75	60.31
3	n-gram(n=2)	58.53	57.68
4	n-gram(n=3)	51.86	50.65
5	n-gram(n=1,2)	61.13	61.56
6	n-gram(n=1,2,3)	60.96	61.55
7	n-gram(n=1,2),n-gram position	60.40	60.76
8	n-gram(n=1,2),LIWC	61.13	61.59
9	n-gram(n=1,2),MPQA	**61.15**	61.57
10	n-gram(n=1,2),WordNet-Affect	**61.15**	61.57
11	n-gram(n=1,2),POS	61.12	61.62
12	n-gram(n=1,2), LIWC, MPQA, WordNet-Affect, POS	**61.15**	**61.63**

classifiers, respectively). This suggests that emotions are expressed in an implicit way compared to sentiments are, and accurate results cannot be obtained with only emotion or sentiment bearing keywords. Recall that the example *fear* in Table 1.1, in which the writer expresses fear emotion without explicitly saying something like "I am scared."

We used LingPipe[3] for POS tagging, and trained the tagger on a POS annotated tweet corpus (Gimpel et al., 2011). Then, we calculated the percentage of words that belong to each POS[4] in a tweet as features. Line 11 in Table 3.4 shows the accuracy after incorporating POS features. However, it does not improve the performance much.

[3]http://alias-i.com/lingpipe/

[4]Refer to Table 1 in paper (Gimpel et al., 2011) for a complete list of POS tags

3.5 Conclusions and Future Work

In this chapter, we studied the problem of automatically identifying authors' emotions from text by casting it as a classification problem. We studied a variety of features and their contributions to the supervised machine learning approach. Since a supervised machine learning approach is not effective at gleaning features from minority emotion classes of sparse labeled sentences, we proposed an algorithm to automatically spot beneficial features to construct our rule-based classifier. We experimented with the proposed approach on two datasets: suicide notes and Twitter data. On suicide notes, we found that: (1) a combination of unigram, bigram, knowledge-based, syntactic, and class-specific features achieve the best micro-average f-measure; (2) The hybrid classifier, consisting of the supervised classifier and the rule-based classifier, achieved better performance than its two component classifiers, which shows that both component classifiers complement each other on identifying majority and minority emotions. On Twitter data, we found that: (1) The feature combination of N-grams (unigram and bigram), knowledge-based features (sentiment and emotion lexicon), and syntactic features (POS) achieves the best accuracy; (2) Knowledge-based features become less important in the case of a large amount of training data.

We assigned at most one label to each sentence in suicide notes, but more than 10% of all the labeled sentences should have 2 or more labels. As the next step, we plan to explore assigning multiple labels to a sentence and see if it improves the performance. An interesting future work is to apply online learning to constantly train machine learning algorithms with the incoming tweet stream. Another interesting future work is to explore how to apply ensemble learning techniques to further improve the performance.

Self-labeled Data Creation

In this chapter, we explore how to automatically create a large, self-labeled emotion dataset by leveraging the hashtag phenomenon in Twitter data. We apply emotion hashtags to retrieve emotion-related tweets, develop filtering heuristics to improve the quality of these retrieved tweets, and automatically annotate these tweets with emotion labels with the help of an emotion taxonomy. Our evaluation shows that the automatically created emotion labels have a reasonably high accuracy and that large labeled data is necessary for training supervised machine learning classifiers to tackle the emotion identification problem.

4.1 Overview

A large, labeled dataset is a necessity for the study of emotion identification, because emotions can be triggered by various specific events or situations in our daily lives. Most current emotion identification research relies on relatively small, manually annotated datasets. Manual annotation of data by human experts is very labor-intensive and time-consuming. Moreover, in contrast with other annotation tasks such as entity or topic detection, human annotators' judgment of emotions in text tends to be subjective and varied, and hence, less reliable. Consequently, existing emotion datasets are relatively small, of the order of thousands of entries, and they fail to provide a comprehensive coverage of emotion-triggering events and situations.

While there is a lack of sufficient labeled data for emotion research, many social media

services have entered the big data era. Twitter, a popular microblogging service, provides more than 500 million tweets per day [1] on a wide variety of topics, and a significant part of it is about "What is happening" in our daily lives captured using emotion hashtags. For example, "leaving for the hospital... #nervous". Now the question is *"Can this big Twitter data be harnessed to tackle the emotion identification problem?"*

In this chapter, we explore some of the challenges and opportunities of creating a large, self-labeled emotion dataset, harnessing the big Twitter data. Specifically, we focus on investigating the following questions: Can we automatically create a large emotion dataset with high quality labels from Twitter by leveraging the hashtag phenomenon? If so, how? Does the scale of large training data play an important role in identifying emotions? How much performance will be gained by increasing the size of the training data?

To answer the above questions, we used 131 emotion hashtags as keywords and collected 5 million tweets for 7 emotion categories in six weeks. To improve the quality of collected tweets, a set of heuristics were developed to retain relevant tweets, which contain the emotion hashtags that correctly annotate the expressed emotions. The evaluation of these heuristics shows that they can be used to create a high quality emotion dataset by effectively identifying relevant emotion tweets with a precision of 93.16%. After applying these heuristics, we obtained an emotion tweet corpus containing 2.5 million tweets. To investigate the contributions of the large training data, we increased the size of training data from 1,000 to 2 million and achieved an absolute gain of 22.16% on accuracy.

4.2 Problem Definition

We aim at designing an approach to collect a large emotion dataset that can cover different emotional moments in our daily lives. Moreover, we want to automatically annotate the instances in this dataset with corresponding emotion labels so that the human involvement

[1] https://blog.twitter.com/2013/new-tweets-per-second-record-and-how

can be greatly reduced.

4.3 Methods

We first collected 7 sets of emotion words for 7 different emotions (e.g., word "annoying" for emotion *anger*) from existing psychology literature, and then utilized the Twitter streaming API to collect tweets that had one of these emotion words in the form of a hashtag (e.g, #annoying). Each collected tweet was automatically labeled with one emotion according to its emotion hashtag, and the hashtag itself was removed from the tweet. For example, from an incoming tweet "I hate when my mom compares me to my friends.**#annoying**", we obtained the following training example: "I hate when my mom compares me to my friends." labeled with anger emotion, since it contains the "#annoying" hashtag.

4.3.1 Collecting Emotion Hashtags

Our source of the emotion words is a highly cited psychology paper (Shaver et al., 1987). In this paper, the authors organize emotions into a hierarchy in which the first layer contains six basic emotions (i.e., *love*, *joy*, *surprise*, *anger*, *sadness* and *fear*), and the second layer contains 25 secondary emotions that are subcategories of the six basic emotions. Each secondary emotion has a list of emotion words. We expanded the list of emotion words by including their lexical variants, e.g., adding "surprising" and "surprised" for *surprise*. In addition, we removed ambiguous words. For example, "glee" denotes "great delight" in dictionary and is used to indicate the emotion *joy*, but it is also the name of a popular TV series in the U.S. For each basic emotion, we used the emotion words of all its secondary emotions to collect tweets. Besides the aforementioned six basic emotions, we added one more emotion *thankfulness* which we think is not covered by (Shaver et al., 1987). Table 4.1 shows the seven emotions, sample emotion hashtags, example tweets and

the number of tweets after filtering irrelevant tweets.

4.3.2 Filtering Heuristics

Totally, we collected 5 million tweets during a six-week period. Before using these tweets as training examples, it is necessary to verify their quality, i.e., whether the emotion hashtags truly indicate the authors' emotional states. For this purpose, we randomly sampled a set of 400 tweets. Two annotators first independently annotated each tweet as relevant/irrelevant, and when there was a disagreement on the annotation of a tweet, they collaborated to reach an agreement. A tweet is labeled as relevant if the emotion hashtag in the tweet reflects the writer's emotion. Otherwise, it is labeled as irrelevant. The result showed that only 46% of the tweets are labeled as relevant, which further suggested the need for filtering irrelevant tweets.

A set of filtering heuristics was developed on the aforementioned set of 400 tweets. (1) We kept only the tweets with the emotion hashtags at the end of them. Based on our observation, if the emotion hashtag is not at the end of a tweet, it is less likely that the hashtag indicates the author's emotional state. This observation has been supported by another study (Choudhury et al., 2012). (2) We discarded tweets that had less than five words, since they may not provide sufficient context to infer emotions. (3) We removed the tweets that contain URLs or quotations. We found that a large number of tweets with URLs are information-oriented, which do not convey emotions. For those tweets that quote others' words, the quoted content may be the target of the emotion, from which we cannot infer the emotion. For example, in the tweet " 'For you, i was a chapter. For me, you were the book.' -Tom McNeal #LOVE", the love emotion cannot be inferred from the content of the tweet, since the quote is the emotion target, but not the emotion expression. Furthermore, we removed all the retweets and the tweets that were not in English.

Table 4.1: Emotion words used for collecting tweets and the number of collected tweets for each emotion (after filtering)

Emotion	**Hashtag Word (#)**	**# of Tweets**	**Tweet Example**
joy	excited, happy, elated, proud (36)	706,182	“Omg I finally fit into one pair of my jeans from last year!!” #excited
sadness	sorrow, unhappy, depressing, lonely (36)	616,471	“im losing both of my semi-final games #depressing”
anger	irritating, annoyed, frustrate, fury (23)	574,170	“Ugh I have no money but payday tomorrow #Irritating”
love	affection, lovin, loving, fondness (7)	301,759	“iloveyou, just the way you are #love”
fear	fear, panic, fright, worry, scare (22)	135,154	“Calculus test today #studying #nervous”
thankfulness	thankfulness, thankful (2)	131,340	“The Maury show makes me realize my life isn’t so bad. #thankful”
surprise	surprised, astonished, unexpected (5)	23,906	“Today’s going a lot better than I thought on no sleep #surprised”
TOTAL	**(131)**	**2,488,982**	

4.4 Experiments

In this section, we first evaluate the performance of filtering heuristics: what is the quality of the self-labeled dataset after applying filtering heuristics? We then evaluate the importance of utilizing large training data for emotion identification by varying the sizes of the training data.

4.4.1 Evaluation of Filtering Heuristics

To evaluate the filtering heuristics, we randomly sampled another disjoint set of 400 tweets, annotated each tweet as relevant/irrelevant in the same manner as the first 400 tweets, and used it as the test dataset. We applied the heuristics on both the development dataset and the test dataset. The precision and recall on the development dataset are 95.08% and 94.57%, while the precision and recall on the test dataset are 93.16% and 93.65%. Thus our filtering heuristics are effective in removing irrelevant tweets. After applying the heuristics on all the collected tweets, we finally obtained a collection of 2,488,982 tweets. The distribution of tweets per emotion is summarized in Table 4.1.

4.4.2 Evaluation of Benefits with Large Training Data

Out of 2,488,982 tweets in Table 4.1, we randomly sampled 250,000 tweets as a test dataset **Te**, reserved another randomly sampled 247,798 tweets as a development dataset for parameter tuning, and the remaining 1,991,184 tweets (denoted as **Tr**) were used for training. Note that the test, development, and training datasets are disjoint. We divided **Tr** into eight subsets (denoted as **Tr1**, **Tr2**, ..., **Tr8**, respectively), and each of which comprises 248,898 tweets. **Tr1** was used for exploring effective features, and all eight subsets were used to examine the effect of increasing the size of the training data.

We selected LIBLINEAR (Fan et al., 2008) and Naive Bayes (NB) as the supervised machine learning classifiers, since they are very efficient even when handling millions of

tweets. We evaluated the quality of multi-emotion classification in terms of accuracy.

We examined the effect of increasing the size of the training dataset on the accuracy of the LIBLINEAR and NB classifiers. Since most existing work on emotion identification (Aman and Szpakowicz, 2008) is conducted on datasets of thousands of sentences, we expect to derive new insights and benefits from "big data". We also started with thousands of tweets. We randomly sampled a set of 1,000 tweets, and another disjoint set of 9,000 tweets from Tr1, denoting them as Tr11 and Tr12. Finally we created a sequence of datasets with increasing sizes: Tr11, Tr11∪Tr12, Tr1, Tr1∪Tr2, Tr1∪Tr2∪Tr3, ..., Tr1∪Tr2∪...∪Tr8, in which each smaller dataset is contained in the subsequent dataset.

We experimented with a few feature combinations (line 6 and 12 in Table 3.4), but the results do not differ much from using the combination of unigrams and bigrams. Thus, we trained LIBLINEAR and NB classifiers on each dataset in the sequence with unigram and bigram features. Figure 4.1 shows the accuracies of applying the classifiers on the test dataset Te.

Benefits of Increasing Training Data: Observing from Figure 4.1, as training data increases from 1,000 to about 2 million, we get an absolute accuracy gain of (65.57%-43.41%=) 22.16% with the LIBLINEAR classifier. Specifically, accuracy grows by $(52.92\% - 43.41\% =)$ 9.51%, $(61.56\% - 52.92\% =)$ 8.64%, and $(65.57\% - 61.56\% =)$ 4.01% when training data increases from 1,000 to 10,000 tweets, 10,000 to about 250K tweets, and about 250K to about 2M tweets, respectively. This result demonstrates that learning from large data can play an important role in emotion identification.

LIBLINEAR vs. NB: With 1,000 training tweets, the accuracy of the NB classifier (45.80%) is 2.39% higher than that of LIBLINEAR classifier (43.41%). But LIBLINEAR classifier benefits more from increasing the training data. With about 2 million training tweets, the LIBLINEAER classifier (65.67%) beats the NB classifier (63.50%) by 2.17%. The better accuracy of the LIBLINEAR classifier comes at the price of much longer training time. It takes 9 hours and 22 minutes for the LIBLINEAR classifier to train on 2 million

tweets while it takes the NB classifier only 1.5 minutes to train on the same dataset.

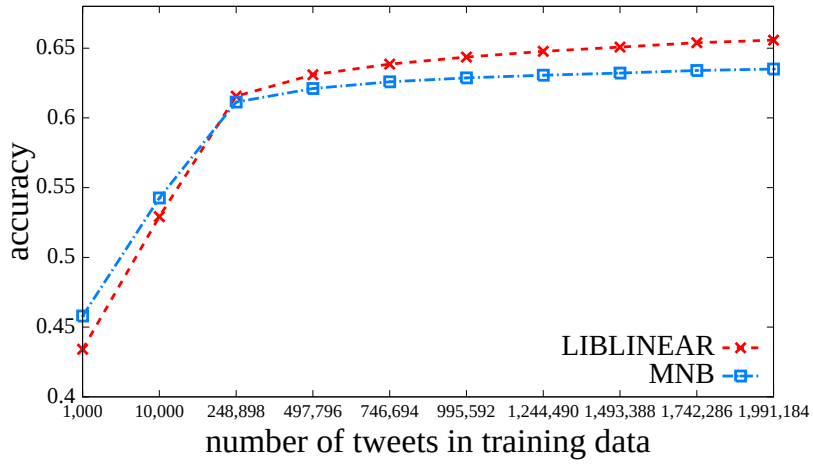

Figure 4.1: Accuracies of LIBNEAR and NB with varied sizes of training data

Table 4.2 shows the performance of the LIBLINEAR classifier (trained with all tweets in Tr) on each emotion category. We have the following discoveries. For the three most popular emotions – joy, sadness, and anger, which account for 76.2% of all tweets, the classifier achieves precisions of over 62%, recalls of over 66%, and F-measures of over 64% for each of the three emotions. Performance declines can be seen on three less popular emotions (i.e., love, fear and thankfulness), which consist of 22.8% of all the tweets in our dataset. The precisions of these three emotion categories are relatively high (with the lowest precision being 58.1%) compared with the recalls, but because of the low recalls, the classifier achieves F-measures of only 51.5%, 43.9%, and 57.1% for love, fear and thankfulness categories, respectively. For the remaining minority emotion (i.e., surprise, only 1% of all tweets), the classifier gets the lowest precision, recall, and F-measure because of the heavily imbalanced emotion distribution in the training data.

To get more insights from the result, we analyze the confusion matrix to figure out what are the top misjudged cases. Out of the 86,071 misjudged tweets, there are 20,799 (24.2%) tweets misclassified between sadness and anger (i.e., either sadness tweets were misclassified as anger or vice versa), 13,400 (15.6%) tweets misclassified between love and

Table 4.2: Detailed result of LIBLINEAR with the largest training data

Emotion	Precision(%)	Recall(%)	F-measure(%)
joy	67.6	**77.3**	**72.1**
sadness	62.6	66.8	64.7
anger	**69.8**	73.3	71.5
love	58.1	46.2	51.5
fear	59.7	34.7	43.9
thankfulness	66.6	50.0	57.1
surprise	44.7	8.2	13.9

joy, and 11,709 (13.6%) tweets misclassified between joy and sadness. This is in line with the fact that some emotion pairs (anger and sadness, joy and love) are naturally related to each other. Moreover, different people might have different emotions faced with similar events. For example, "My phone bout to die too..uggghhhhh #annoyed" vs. "It's dark so I can't read, my phone is about to die so no music. #sad". It is interesting to see that misjudging also happens between emotions with opposite polarities, e.g., joy and sadness. The reason for this counter-intuitive behavior is that words carrying opposite emotions can co-exist in the same tweet. For example, "That was the sexiest punch I've ever seen #happy" was labeled as sadness because of the negative word "punch".

4.5 Discussions

In this section, we share our observations and experiences playing with Twitter data for emotion identification. We also talk about the challenges we faced and the potential applications of our research.

Compared with manually annotated emotion datasets, the corpus of automatically collected tweets with emotion labels (i.e., hashtags) shows its advantages in several aspects. Firstly, the emotion hashtags of tweets are provided by their writers, which are more reliable than the emotion labels of other data given by a few annotators. In contrast, the tra-

Table 4.3: The diversity of emotions of tweets containing "miss you"

Emotion	#/Percentage	Example
sadness	2849 / 57.6%	"miss you, you need to be here! #lonely"
love	1187 / 24.0%	"Good night @user i love you and miss you< 3 #love #hugs"
joy	507 / 10.3%	"I miss you too! must see you this weekend. #excited!"
anger	220 / 4.4%	"i miss you < /3 why can't this feeling just go awayyyyyy #annoyed"
thankfulness	93 / 1.9%	"Miss you too. I had an great day with you and Jax it was exactly what I needed. #thankful"
fear	62 / 1.3%	"@user where are ya I miss you .... #worried"
surprise	26 / 0.5%	"Gosh man! I can't even sayy ii miss youu* -__- #surprised"

ditional method of annotation requires annotators to infer the writers' emotions from text, which may not be accurate. Secondly, it is very labor-intensive and time-consuming to manually annotate the data, which greatly limits the sizes of training datasets. Utilizing the Twitter streaming API, we can efficiently collect large training data for studying people's emotions. Based on the fact that we collected 2.5 million tweets with high-quality emotion hashtags in six weeks, over 20 million such tweets can be collected from Twitter per year using the free public API (using the full Twitter corpus requires paid access, and the size may be even larger). As we have shown, larger training data will lead to higher accuracy of emotion identification. Thirdly, the large collection of tweets can provide a comprehensive coverage of emotional moments in our daily lives. In the BLOG dataset (Aman and Szpakowicz, 2008), all of the three sentences containing "miss you" are labeled as sadness. However, Table 4.3 shows that in our dataset there are 4,944 "miss you" tweets covering all seven emotions: sadness (57.6%), love (24.0%), joy (10.3%), anger (4.4%), thankfulness (1.9%), fear(1.3%) and surprise (0.5%). Readers can refer to the examples in Table 4.3 to figure out why "miss you" is even associated with anger and surprise emotions.

However, improvements need to be done to further refine the collected Twitter data. Firstly, we cannot manually verify all the emotion tweets because of its large scale. Although we have removed some irrelevant tweets using heuristics, there are still some tweets not being correctly labeled. In our case, after the filtering process, 93.16% of tweets are relevant on a test bed of 400 tweets. Secondly, it does not contain tweets with neutral labels, which is a common problem for automatically labeled emotion training data (Tokuhisa et al., 2008; Yang et al., 2007a). Davidov et al. (2010) have shown that many hashtags indicate no sentiment, so one possible direction is to identify hashtags that can be used to retrieve no emotion tweets. Thirdly, the distribution of emotions in our Twitter dataset is imbalanced, e.g., only 1% of the tweets belong to *surprise*, and the classifiers do not perform well on less popular emotions. To further improve the performance, one possible solution is to increase the number of tweets for minority emotions by using more of their hashtags to collect tweets. Also, undersampling tweets from majority emotions might be another potential solution to try (Chawla et al., 2004).

We also face several challenges. First, text normalization by dictionary lookup alone is not sufficient on tweets, because the dictionary lookup method may turn many out-of-vocabulary words into dictionary words incorrectly. For example, "b4" is a shortened form of "before", but the lookup approach may convert "b4" to "be" because they have a high string similarity score. Also, different words can be shortened into the same short form. For example, "n" is short for "and" in "sweet n sour chicken", but it stands for "in" in "finally n bed". Second, most existing classifiers take too much time to be trained on large data. Based on our aforementioned estimation, 21.6 million emotion tweets per year can be collected for training, and how to finish the training on such a large data in a reasonable amount of time is a big challenge.

We believe our approach can improve human computer interaction experience by extracting emotion signals from text. Emotion signals extracted from facial expressions and body gestures (Pantic et al., 2007) have been envisioned as important input signals in the

next generation user interface. Our approach can provide emotion signals extracted from different sources: text written by users or converted from speech. Imagine the following conversation between a smart phone (SP) and a human (H). H: "I overcame my fear of talking in front of people". SP: "That is awesome! I am so proud for you!" Also, social network websites may benefit from identifying users' emotions in their status updates. For example, every time a user posts a new status, the website can automatically change the "skin style" of the personal homepage or play different types of background music to accommodate the user's emotion. Also, emotion signals can contribute to the detection of whether a person is suffering from depression through this person's social media posts (Choudhury et al., 2013b). Moreover, aggregating emotions identified in status updates from individual users can be used to compute "Gross National Happiness" (Kramer, 2010), a metric representing the happiness in a country.

4.6 Conclusions and Future Work

In this chapter, we studied the problem of automatically creating self-labeled tweets by leveraging large data from social media. We collected about 2.5 million labeled tweets in six weeks via the publicly accessible Twitter API, which demonstrates that it is practical for leveraging the hashtag phenomenon and creating a large scale emotion dataset. By applying simple filtering heuristics, we effectively improved the quality of the self-labeled dataset. We found that: (1) as we increased the training data from 1,000 to 2M, we achieved an absolute gain of 22.16% on accuracy with the LIBLINEAR classifier, which supports the importance placed on the role of large training data for emotion identification; (2) on large training data, LIBLINEAR achieved a better accuracy, while NB was more efficient; (3) we achieved reasonably high f-measures for the three most popular emotions (joy, sadness and anger) and reasonably high precisions for the three less popular emotions (love, fear and thankfulness).

As part of future work, we want to explore how to automatically collect neutral tweets so that the system can support the detection of neutral emotion. We plan to add more emotion hashtags to increase the number of tweets for less popular emotions (especially surprise and fear) to reduce the imbalance of the dataset. We also plan to apply online learning (Bifet et al., 2010) to constantly train machine learning algorithms with the tweet stream as well as to adopt large-scale machine learning algorithms such as MLlib in Spark (Meng et al., 2015).

Domain Adaptation for Emotion Identification

While there is a short supply of labeled data for emotion studies in many domains (e.g., blog posts and fairy tales), a massive number of emotion labeled tweets can be automatically collected from Twitter by utilizing emotion-related hashtags. These labeled tweets, possibly with a different distribution, share a common base of emotion expressions with instances in other domains, and can be harnessed to help identify emotions. In this chapter, we adapt labeled tweets from Twitter (the source domain) to improve emotion identification in target domains where only a small number of instances are labeled. To fill the distribution gap between domains, we selectively and iteratively identify useful instances from the source domain and add them into the training data to train an adaptive classifier for the target domain. We create a large emotion-labeled Twitter dataset (of 100K tweets), and perform extensive experiments on four existing emotion datasets from four domains: blogs, experiences, diaries and fairy tales.

5.1 Overview

Emotion identification aims to automatically identify people's emotions expressed in text, e.g., anger, disgust, fear, joy, sadness, and surprise. As the emotion-rich content grows

rapidly on the Web, there is an increasing need to develop tools and techniques for emotion identification from various domains. Some recent research efforts have been devoted to identifying emotions from fairy tales (Alm et al., 2005), blog posts (Aman and Szpakowicz, 2008; Neviarouskaya et al., 2011), music and lyrics (Yang and Lee, 2009; Mihalcea and Strapparava, 2012).

In Chapter 3, we show that a potential bottleneck for emotion identification is the lack of sufficient labeled training data. Statistical classification algorithms usually require a large amount of labeled data to train a reliable classifier. However, manually labeling emotions in text is labor-intensive and time-consuming. Moreover, compared with other annotation tasks such as entity or topic detection, a human annotator's judgment of emotions in text tends to be more subjective and varied, and hence, it is more difficult to create a high-quality labeled dataset for emotion studies.

To tackle this challenge, recall that in Chapter 4, we exploit emotion hashtags in tweets to automatically infer their emotion labels. For example, we can obtain the following self-labeled instance ⟨"Exactly one month until christmas! Woot **#excited**", *joy*⟩, where the tailing emotion hashtag "#excited" is stripped from the tweet and is used to label this tweet with emotion *joy*. In this way, a large number of self-labeled emotion tweets can be automatically collected from Twitter. It is appealing to adapt these tweets to help identify emotions in target domains where the labeled instances are in short supply.

Adapting tweets to target domains is a challenging task for at least the following reasons. First, the emotion label of a tweet might not be consistent with its content. A tweet may convey a mixture of emotions, whereas not all the emotions can be inferred if the author did not put up hashtags for all the embedded emotions. For example, the first half of tweet #1 in Table 5.1 conveys emotion *joy* that is not included in its label – *fear*. Second, in order to help identify emotions in a target domain, we prefer to select the source domain instances that complement the knowledge which the target domain lacks, rather than select the instances that add the redundant knowledge which the target domain already contains.

Table 5.1: Emotion tweets: the emotion label in front of each tweet is inferred from the emotion hashtag in bold; informal expressions (misspellings, abbreviations and multi-word concatenations) are underlined.

#1	Fear: "Amazing night with my baby. Hope she liked our anniversary present. Alil early but whatever. :) hopefully tmmrw goes as planned. **#fear**"
#2	Sadness: "Why does my phone have to die so early in the morning. #canttweet **#depressing"**
#3	Anger: "My phone batt dies so quiick ....! **#annoyed"**

Third, features on Twitter and target domains (e.g., blogs and fairy tales) are different. As Table 5.1 shows, informal expressions, such as misspellings ("quiick" in tweet #3), abbreviations ("Alil" and "tmmrw" in tweet #1, "batt" in tweet #3) and multi-word concatenations ("#canttweet" in tweet #2) are common on Twitter. However, these expressions may be rarely used in target domains.

In this chapter, we focus on the domain adaptation problem for emotion identification. Our idea is to apply a bootstrapping framework to iteratively select informative tweets to enrich the target domain training data. We propose to define the **informativeness** of a source instance using three factors: consistency, diversity, and similarity. **Consistency** measures the confidence of a tweet's label being consistent with its content, estimated by the labeled data from both source and target domains. **Diversity** is introduced to encourage the selection of source instances containing features that are infrequent or underrepresented in the target domain. **Similarity** prompts source instances that are very similar to test instances in target domain. We evaluate the proposed approach on four target datasets. Results show that our approach is effective for cross-domain emotion identification and outperforms several baseline approaches.

5.2 Problem Definition

We first define the problem. Let X be the observable feature space to represent the data in, and Y be the label space. In this study, $Y = \{anger, disgust, fear, joy, sadness, surprise\}$.

The labeled tweet set (i.e., source domain labeled data) is denoted by $D_l^s = \{(x_i^s, y_i) \in X \times Y \mid y_i$ is the label associated with the instance $x_i^s\}$. Let D_l^t be the target domain labeled data, D_u^t be the target domain unlabeled data, and $D^t = D_l^t \cup D_u^t$ be the overall target domain data. Our objective is: Given a large source domain labeled dataset D_l^s and a target domain labeled dataset D_l^t ($|D_l^s| >> |D_l^t|$), construct a classifier $\hat{c} : X \rightarrow Y$ that will be used to predict emotion labels for target domain unlabeled instances.

5.3 The Proposed Approach

In the following, we first describe the bootstrapping framework, and then present a scoring function that calculates source instances' informativeness using three factors: *consistency*, *diversity* and *similarity*. Through informative measurement, highly informative source domain tweets will eventually be selected and added to enrich target domain training data so that we can train a more accurate target domain classifier using the enriched training data.

5.3.1 The Bootstrapping Framework

We provide a reference to the notation used throughout this chapter in Table 5.2. The goal of the bootstrapping framework is to augment target domain labeled data D_l^t with a subset of instances from source domain labeled data D_l^s to improve overall classification accuracy on the target domain unlabeled data D_u^t. For this purpose, we first build a classifier using D_l^t and apply it to D_l^s to select a subset of informative instances. If the label of a source domain instance is correctly predicted by the classifier, this instance is regarded as redundant, i.e., this knowledge is already contained in the target domain instances. If the predicted label is incorrect, then we consider this source domain instance as a candidate for addition, because it may contain knowledge that is lacking in the target domain labeled data. A scoring function (as presented in Section 5.3.2) is used to determine the informativeness of the

Table 5.2: Table of notation

Symbol	Description
c	Classifier
c_0	Initial classifier trained on D_l^t
$\hat{c}$	Final adaptive classifier
δ	Threshold for selecting informative instances
$\gamma(\cdot,\cdot)$	Scoring function for the consistency between the content of an instance and a label
k	The number of informative instances to be selected per iteration
$\phi(\cdot,\cdot)$	Scoring function for informativeness
$\lambda^c(\cdot,\cdot)$	Scoring function for consistency factor
$\lambda^d(\cdot)$	Scoring function for diversity factor
$\lambda^s(\cdot,\cdot)$	Scoring function for similarity factor
$\pi^c(\cdot,\cdot)$	Scoring function for the content similarity between two instances
$\pi^l(\cdot,\cdot)$	Scoring function for the label similarity between two instances
$\pi^u(\cdot)$	Scoring function for the uncertainty factor
D_l^s	Source domain labeled data
D_l^t	Target domain labeled data
D_u^t	Target domain unlabeled data
$D^t = D_l^t \cup D_u^t$	Overall target domain data
T	Training Data for classifier c
$T_{correct}^t$	Set of instances from D_l^t that can be correctly classified by c_0
T_{wrong}^t	Set of instances from $T_{correct}^t$ that are misclassified by c
T^s	Remaining source domain labeled data after selecting informative instances in each iteration
T_{wrong}^s	Set of instances from T^s that are misclassified by c
T_{info}^s	Set of informative instances selected from T^s
X	The observable feature space
Y	The label space

candidate, and decide whether to select the candidate.

The addition of informative source instances to D_l^t can be used to obtain a new classifier. Ideally, one would expect this new classifier to correctly classify more target domain instances. However, it may misclassify the target domain labeled instances that were correctly classified initially, if a few false informative instances containing inconsistent knowledge were selected. When such misclassification happens, we resort to a "counter-balancing" process to recover. This is achieved by adding these misclassified target domain labeled instances with their correct labels to improve the classification accuracy. In other

Algorithm 1: The bootstrapping framework

Input: D_l^s, D_l^t, D_u^t, k, δ
Output: Adaptive classifier $\hat{c} : X \rightarrow Y$

1 Train an initial classifier c_0 with D_l^t;
2 $T_{correct}^t \leftarrow$ Set of instances from D_l^t that can be correctly classified by c_0;
3 Initialize $T \leftarrow D_l^t$, $T_{info}^s \leftarrow \emptyset$, $T_{wrong}^t \leftarrow \emptyset$, $T^s \leftarrow D_l^s$;
4 **repeat**
5 $\quad T \leftarrow T \cup T_{info}^s \cup T_{wrong}^t$;
6 $\quad$ Train a classifier c with T;
7 $\quad T_{wrong}^s \leftarrow$ Set of instances from T^s that are misclassified by c ;
8 $\quad T_{info}^s \leftarrow$ Top k instances with informativeness $\phi(\cdot, \cdot)$ greater than δ from T_{wrong}^s;
9 $\quad T^s \leftarrow T^s - T_{info}^s$;
10 $\quad T_{wrong}^t \leftarrow$ Set of instances from $T_{correct}^t$ that are misclassified by c;
11 **until** $|T_{info}^s| < k$;
12 **return** c

words, those misclassified instances are given extra weight in the training data.

Algorithm 1 illustrates the bootstrapping framework. Specifically, the algorithm takes as input D_l^s, D_l^t, D_u^t, a natural number k indicating the number of source instances to be added per iteration, and a real number δ indicating the informativeness threshold for selecting source instances. The output is an adaptive classifier $\hat{c}$.

We start with training an initial classifier c_0 using D_l^t (line 1). We initialize $T_{correct}^t$ with instances from D_l^t that can be correctly classified by c_0 (line 2). We initialize the overall training data T to D_l^t, newly selected informative source domain instances T_{info}^s to $\emptyset$, counterbalancing target domain instances T_{wrong}^t to $\emptyset$, and source domain candidate instances T^s to D_l^s (line 3).

In every iteration, we first add the newly selected informative instances T_{info}^s and counterbalancing target domain instances T_{wrong}^t into the overall training data T (line 5) that will be used to train a new classifier c (line 6). We set T_{wrong}^s to the instances in T^s whose labels are different from those predicted by classifier c (line 7). As discussed earlier, these instances have a potential to augment target domain training data by complementing them with the knowledge that they lack. We then set T_{info}^s to the top k informative

instances selected from T^s_{wrong} based on a scoring function that will be explained in Section 5.3.2 (line 8). We remove the newly selected informative source instances T^s_{info} from source domain instances T^s (line 9). If a few false informative instances that contain inconsistent knowledge were selected and added to the training data, classifier c may misclassify instances in $T^t_{correct}$ that were initially correctly classified by c_0. To counterbalance such effect, we set T^t_{wrong} to the instances in $T^t_{correct}$ that are misclassified by classifier c (line 10). The instances in T^t_{wrong} will be added to the training data again (i.e., given extra weight) in a new iteration. As we iteratively select informative instances out of T^s, the remaining informative instances in T^s will be less and less. The whole process will stop when we cannot select sufficient number (a predefined number k) of instances in an iteration (line 11). The classifier c trained during the last iteration will be returned as the adaptive classifier.

5.3.2 Selecting Informative Instances

To select informative instances from T^s_{wrong}, we define a source instance's informativeness score as the product of its consistency (λ^c), diversity (λ^d), and similarity (λ^s) factors:

$$\phi(x^s_i, y_i) = \lambda^c(x^s_i, y_i)\ \lambda^d(x^s_i)\ \lambda^s(x^s_i, y_i), \tag{5.1}$$

so that the instance will achieve a large informativeness score only when all the three factors are large. If one factor is small, the informativeness will be penalized after the multiplication. We now show how to calculate each score separately.

Consistency

We want to add a source domain instance that is unambiguously associated with a single label. Moreover, this label should be consistent with the expressed emotion in text, verified by labeled data in both source and target domains. As a negative example, in addition to its

single label *fear*, tweet #1 conveys another emotion *joy*. Such instances contain inconsistent knowledge and therefore should not be selected. Specifically, we seek to select instances whose features provide very strong support for its label and very little support for other emotions, based on source domain labeled data D_l^s and target domain labeled data D_l^t.

Let $x_a \in X$ be an arbitrary source or target instance, and $y_b \in Y$ be an arbitrary emotion label. We want to construct a **consistency function** $\gamma(x_a, y_b)$ to estimate the confidence of label $y_b \in Y$ being consistent with instance $x_a \in X$, verified using D_l^s and D_l^t. For x_a and all its present features $x_{a,m}$ (i.e., its component words), we define $x_{a,u}$ and $x_{a,v}$ as the strongest supporting features for label y_b based on D_l^s and D_l^t, respectively:

$$x_{a,u} = \arg\max_{x_{a,m}} \{p^s(y_b|x_{a,m})\} \tag{5.2}$$

$$x_{a,v} = \arg\max_{x_{a,m}} \{p^t(y_b|x_{a,m})\}, \tag{5.3}$$

where $p^s(y_b|x_{a,m})$ and $p^t(y_b|x_{a,m})$ stand for the conditional probabilities of y_b given $x_{a,m}$ based on D_l^s and D_l^t, respectively. For tweet #1, the strongest supporting features for its label *fear* would be "hope" and "present" as: $p^s(fear|\text{“}hope\text{”}) = 0.5094$, $p^t(fear|\text{“}present\text{”}) = 0.2143$.

Similarly, we define $x'_{a,u}$ and $x'_{a,v}$ as the strongest supporting features of x_a for any emotion y'_b other than y_b ($y'_b \in Y \wedge y'_b \neq y_b$), based on D_l^s and D_l^t, respectively:

$$x'_{a,u} = \arg\max_{x_{a,m},\, y'_b} \{p^s(y'_b|x_{a,m})\} \tag{5.4}$$

$$x'_{a,v} = \arg\max_{x_{a,m},\, y'_b} \{p^t(y'_b|x_{a,m})\}. \tag{5.5}$$

For tweet #1, the strongest supporting features for any label other than *fear* would be "tmmrw" and "night" as: $p^s(joy|\text{“}tmmrw\text{”}) = 0.5625$, $p^t(joy|\text{“}night\text{”}) = 0.5962$. Next, we use the **margin** between the largest conditional probability supporting y_b and that sup-

porting y'_b to define the consistency function as follows:

$$\begin{aligned}\gamma(x_a, y_b) = &\max\left\{p^s(y_b|x_{a,u}), p^t(y_b|x_{a,v})\right\} \\ &- \max\left\{p^s(y'_b|x'_{a,u}), p^t(y'_b|x'_{a,v})\right\},\end{aligned} \tag{5.6}$$

where a large value indicates that: (1) x_a has a strong supporting feature for its label y_b, i.e., large $p^s(y_b|x_{a,u})$ and/or $p^t(y_b|x_{a,v})$; (2) according to both D_l^s and D_l^t, and the chance that x_a expresses any emotion y'_b other than y_b is small, i.e., both $p^s(y'_b|x'_{a,u})$ and $p^t(y'_b|x'_{a,v})$ are small. The larger the value, the more consistent the label y_b is with the expressed emotion in x_a. A negative value indicates that the label y_b is not the most likely label for x_a. As an example that has a negative value, consider the score of tweet #1: $max(0.5094, 0.2143) - max(0.5625, 0.5962) = -0.0868$, which suggests that besides emotion *fear*, it expresses the emotion *joy* too. Such instances with negative consistency scores are likely to contain inconsistent knowledge and therefore are not selected.

Now, we apply the consistency function to measure the **consistency** between source instance x_i^s and its label y_i:

$$\lambda^c(x_i^s, y_i) = \gamma(x_i^s, y_i). \tag{5.7}$$

Diversity

The measure of diversity emphasizes source domain instances that have distinctive features which are infrequent in the target domain training data. A distinctive feature usually carries effective knowledge to identify the emotion, but if this feature frequently appears in the training data, it may suggest that the target domain already has abundant knowledge about this feature; and therefore adding the instances that contain this feature may not further improve the classifier. Rather than selecting the source instances with the redundant knowledge, it is preferable to select the instances that complement the knowledge which the target domain lacks, i.e., the instances with distinctive features that are less frequent in

the training data.

To be specific, for x_i^s, we apply Equations 5.2, 5.3 to find its two supportive features $x_{i,u}^s$ and $x_{i,v}^s$ for its label y_i based on D_l^s and D_l^t, respectively. We select the one with the larger conditional probability out of the two features as the most supportive feature:

$$x_{i,w}^s = \begin{cases} x_{i,u}^s, & \text{if } p^s(y_i|x_{i,u}) \geq p^t(y_i|x_{i,v}) \\ x_{i,v}^s, & \text{otherwise.} \end{cases} \tag{5.8}$$

For tweet #1, since $p^s(fear|\text{“}hope\text{”}) \geq p^t(fear|\text{“}present\text{”})$, “hope” is the most supportive feature. If “hope” is infrequent in the training data, we want to promote this tweet to increase the diversity; otherwise, we want to demote this tweet. Let $df(x_{i,w}^s)$ be the number of instances that contain feature $x_{i,w}^s$ in the training data T (i.e., the document frequency). We define the diversity of x_i^s using the **exponential decay** of the document frequency of its most supportive feature $x_{i,w}^s$:

$$\lambda^d(x_i^s) = e^{-\theta\ df(x_{i,w}^s)}, \tag{5.9}$$

where θ is a decay constant. The smaller the $df(x_{i,w}^s)$, the larger the diversity with a max value of 1. In the case that the most supportive feature is present only in the source domain and is not present in the target domain (e.g., informal tweet-specific features: slangs, abbreviations), we will use the next most supportive feature that is present in the target domain.

Similarity

Prior studies (Eck et al., 2004; Lü et al., 2007) have shown that the adaptability of machine translation models can be improved by selecting source domain sentences that are similar to target domain sentences, because these sentences can better match the test data in the target domain. In our problem settings, besides the content similarity, we also need to examine the

label similarity. Otherwise, we may select source instances (tweets) with nearly identical content but labeled with different emotion hashtags by the authors. Both tweet #2 and #3 in Table 5.1 describe similar scenarios of a phone that is running out of battery, but they are labeled with *sadness* and *anger*, respectively. Moreover, we emphasize the unlabeled instances that the classifier c is uncertain about, and select source instances $(x_i^s, y_i) \in T^s$ that are similar to them. When c is uncertain about an instance $x_j^t \in D_u^t$, it suggests that the target domain is lacking the corresponding knowledge to make a confident prediction.

Specifically, to encourage the selection of source instances that share high content and label similarities with target domain unlabeled instances that classifier c is uncertain about, we define the similarity factor of x_i^s as:

$$\lambda^s(x_i^s, y_i) = \max_{x_j^t \in D_u^t} \{\pi^c(x_i^s, x_j^t)\ \pi^l(x_j^t, y_i)\ \pi^u(x_j^t)\}, \tag{5.10}$$

where $\pi^c(x_i^s, x_j^t)$ denotes the content similarity between x_i^s and x_j^t, $\pi^l(x_j^t, y_i)$ indicates how likely x_j^t and x_i^s share the same label y_i, and $\pi^u(x_j^t)$ represents the uncertainty of classifier c regarding x_j^t.

To quantify the **content similarity** between x_i^s and x_j^t, we apply cosine similarity to their weight vectors $\overrightarrow{V^s}(x_i^s)$ and $\overrightarrow{V^t}(x_j^t)$:

$$\pi^c(x_i^s, x_j^t) = \frac{\overrightarrow{V^s}(x_i^s) \cdot \overrightarrow{V^t}(x_j^t)}{|\overrightarrow{V^s}(x_i^s)||\overrightarrow{V^t}(x_j^t)|}. \tag{5.11}$$

The purpose of using weight vector representations is that we want to boost the weights of important words, so that x_i^s and x_j^t are similar to each other only when they share important words. For $(x_i^s, y_i) \in T^s$, we want to assign larger weights to words that are strong indicators of its label y_i. For the m-th present feature $x_{i,m}^s$ of instance x_i^s, we apply the

conditional probability of y_i given this feature based on D_l^s as its weight:

$$weight_{i,m}^s = p^s(y_i|x_{i,m}^s). \tag{5.12}$$

For a target domain unlabeled instance $x_j^t \in D_u^t$, we cannot apply the above equation to calculate the conditional probability of the label given a feature because its label is unknown. Thus, we use a TF-IDF weighting scheme to assign weights instead. Since we are conducting sentence-level emotion identification and most features usually occur once in a sentence, we skip the TF part and apply only the prob IDF from SMART notation (Manning et al., 2008). Specifically, the weight of the n-th present feature $x_{j,n}^t$ of the instance x_j^t is:

$$weight_{j,n}^t = \max\left\{0, \log_{10}\frac{N - df(x_{j,n}^t)}{df(x_{j,n}^t)}\right\}, \tag{5.13}$$

where $N = |T|$ and $df(x_{j,n}^t)$ is the number of instances that contain feature $x_{j,n}^t$ in training data T (i.e., document frequency).

Besides the content similarity, we also need to consider the **label similarity**. Otherwise, we may add instance x_i^s that is similar to x_j^t in terms of content but has a contradicting label. Since the label of x_j^t is yet to be predicted, we cannot directly compare the labels of x_j^t and x_i^s. Instead, we cast this problem as how likely that x_j^t shares the same emotion label y_i of x_i^s. For x_j^t, we apply the consistency function (Equation 5.6) to measure the confidence that x_j^t has the same label y_i as x_i^s:

$$\pi^l(x_j^t, y_i) = \gamma(x_j^t, y_i). \tag{5.14}$$

The larger the value, the more likely that x_j^t and x_i^s share the same label y_i. The value's being negative indicates that it is likely that the label of x_j^t is different from that of x_i^s.

Let y_j^* be the most likely label predicted by c for x_j^t. We define the **uncertainty** of

classifier c regarding x_j^t as:

$$\pi^u(x_j^t) = 1 - p(y_j^*|x_j^t; c). \tag{5.15}$$

In summary, the informativeness scoring function achieves a large value when the following three conditions are all satisfied for a source instance: (1) its label is consistent with its content, (2) it contains a distinctive feature that is infrequent in target training data, and (3) it is similar to a target domain unlabeled instance whose label cannot be predicted by the classifier c with confidence.

5.4 Experiments

Here, we describe our experiments on a collection of emotion tweets and four target datasets, and show the effectiveness of our proposed algorithm for cross-domain emotion identification.

5.4.1 Data and Experimental Setting

Twit: Following Chapter 4, we used the same 122 emotion hashtags as filtering keywords and collected 100K tweets for six emotions as the source data.

The following sentence-level emotion datasets have been used as target domain data.

Blog: Blog sentences retrieved using seed words from emotion-rich blog posts containing real-world emotion expressions (Aman and Szpakowicz, 2008).

Diary: A collection of diary-like blog sentences expressing thoughts and feelings (Neviarouskaya et al., 2011).

Exp: Text extracted from personal stories about daily life experiences (Neviarouskaya et al., 2010).

Fairy: Stories obtained and annotated from fairy tales, including Grimms, H.C. Andersen's, and B. Potter's stories (Alm et al., 2005).

Table 5.3: Dataset statistics

Name	Instance #
Source Domain	
Twit	100,000
Target Domains	
Blog	1,290
Diary	507
Exp	384
Fairy	1,722

These datasets have different emotion labels, which brings extra complexity to the experiments. For example, Diary has the emotions *interest* and *shame*, but Blog does not. To concentrate on the adaptation problem, we focus on emotions which are common to every dataset: *anger*, *disgust*, *fear*, *joy*, *sadness*, and *surprise*. Statistics of all the datasets are shown in Table 5.3. We observe that these datasets are relatively small compared with Twit (100K) – Diary and Exp contain about 500 sentences. We believe the domain adaptation algorithms that exploit Twitter data could be helpful for emotion identification in these cases.

We performed the same data preprocessing on all the datasets. Specifically, we lowercased all the words; replaced letters/punctuation marks that are repeated with the same two letters/punctuation marks (e.g., "cooool" $\rightarrow$ "cool", "!!!!!" $\rightarrow$ "!!"); normalized some frequently used informal expressions (e.g., "ll" $\rightarrow$ "will"). For Twit data, we replaced user mentions (e.g., "@BarackObama") with "@user" to anonymize users; and stripped hash symbols ("#christmas" $\rightarrow$ "christmas").

We used a logistic regression classifier in LIBLINEAR (Fan et al., 2008) for classification because: (1) it is very fast, and (2) it natively supports probability output that is used to calculate uncertainty in Equation 5.15. We experimented with different feature representations: unigrams, bigrams, unigrams and bigrams, and found that the unigram representation achieves the best performance for all the baseline approaches. So we report results using unigrams in this chapter. We performed frequency-based feature selection: the unigrams

appearing in at least five different tweets in Twit or at least two different sentences in other datasets were selected as features. For each dataset, we applied five-fold cross validation where four folds were used as target domain labeled data and the remaining fold was used as test data. We repeated this five times and the average of micro-averaged F_1 scores in five folds was used for performance measurement.

We use **CDS** to abbreviate the proposed method – **C**onsistency, **D**iversity and **S**imilarity. We set the exponential decay constant $\theta = 0.05$. We set the number of selected informative instances per iteration $k = 0.05\ |D_l^t|$, proportional to the number of labeled instances in target domains. We will study the effect of changing this proportion in Section 5.4.3. We empirically set the informativeness threshold $\delta = 0.0005$ as its default value and study its effect in Section 5.4.3. Throughout the chapter, we used add-0.5 smoothing (Manning et al., 2008) to estimate the conditional probabilities of a label given a feature: e.g., $p^s(y_i|x_{i,m}^s)$ on source data and $p^t(y_i|x_{i,m}^s)$ on target data.

5.4.2 Baseline Approaches

We compare CDS against the following five approaches:

Source Only (SO): We train classifiers using only Twit. The results can be used as a "starting point". Without any adaptation, we determine the performance by directly applying the classifier trained on source Twit to target datasets.

Target Only (TO): Since the target domain training data is more representative of target domains than Twit is, we train classifiers using only the target domain training data.

Feature Augmentation (FA): The idea of this approach is to "augment the feature space of both the source and target data and use the result as input to a standard learning algorithm" (Daumé, 2007). By doing so, the classifier can select and apply distinctive features from the augmented feature space when applied to target domain data.

Feature Injection (FI): The idea is to first train a source classifier using only the source data. Then, this classifier is applied to both the labeled and unlabeled data in the target do-

main, and its probability outputs (i.e., the probabilities of x_j expressing different emotions) will be injected into target data as additional features. A target classifier will be trained using the target data after feature injection (Daumé, 2007).

Balance Weight (BW): Given that labeled instances in the target domain are more representative of the target domain than the source instances, the idea is to assign larger weights for the target instances so that the weight sum of target instances equals that of source instances (Jiang and Zhai, 2007). The weight of every instance in D_l^t is set to $\frac{|D_l^s|}{|D_l^t|}$, and then a classifier is trained on $D_l^s \cup D_l^t$.

It is important to mention that despite their simplicity, prior studies (Daumé, 2007; Barbara, 2011) find some of the above baselines surprisingly difficult to beat.

5.4.3 Evaluations on Domain Adaptation

Table 5.4 presents the experimental results in terms of the micro-averaged F_1 metric obtained by all approaches on four datasets. We observe that: (1) in descending order of the means of their micro-averaged F_1 across all datasets, all approaches rank as follows: CDS (0.6703), BW (0.6404), FA (0.6235), FI (0.6191), TO (0.5756), SO (0.4849); (2) CDS outperforms all the baseline approaches on every dataset; however, the difference between CDS and BW on F_1 metric is not statistically significant (with p-values in parenthesis): Blog(0.204), Diary(0.151), Exp(0.092), Fairy(0.164); part of the reason could be the high variance caused by the relatively small number of target domain instances in experiments. (3) (Daumé, 2007) finds that FA performs worse when the source and target domains are very similar (i.e., SO performs similar to or better than TO); In contrast, if the source and target domains are different (i.e., SO performs worse than TO), FA tends to outperform other approaches. This is corroborated in our experiment: FA performs the best among all the baselines on Blog and Fairy, where SO performs worse than TO. Lastly, (4) BW outperforms other baselines on Diary and Exp, where the performance of SO is similar to or better than that of TO. This seems to suggest that BW can complement FA on datasets

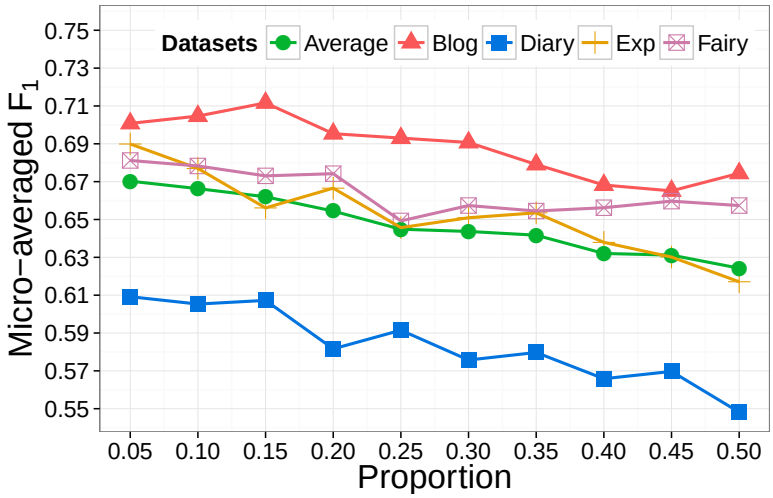

Figure 5.1: Effects of different sample sizes

where the source and target domains are very similar.

Table 5.4: Results for all approaches on four target datasets. For each row, the **best** approach is in bold, the second best is underlined, and the third best is under-waved.

	Micro-averaged F_1					
Datasets	SO	TO	FI	FA	BW	CDS
Blog	0.5054	0.6488	0.6930	0.6969	0.6922	**0.7008**
Diary	0.4870	0.4910	0.5423	0.5383	0.5621	**0.6092**
Exp	0.5261	0.5053	0.5729	0.5834	0.6379	**0.6899**
Fairy	0.4210	0.6574	0.6684	0.6754	0.6702	**0.6812**
Average	0.4849	0.5756	0.6191	0.6235	0.6404	**0.6703**

Influence of the Parameters: k, δ

We vary, k, the number of selected informative instances per iteration, from $0.05\ |D_l^t|$ to $0.5\ |D_l^t|$, to show how it impacts the results in Figure 5.1. The general trend is that micro-averaged F_1 slowly decreases as we select more instances per iteration across all the datasets, because target labeled data gets diluted faster with source data. The best result in micro-averaged F_1 is achieved when the proportion is 0.05.

The informativeness score can be a negative value under two conditions: either consistency $\lambda^c(x_i^s, y_i) < 0$ or label similarity $\pi^l(x_j^t, y_i) < 0$. In practice, we skipped the

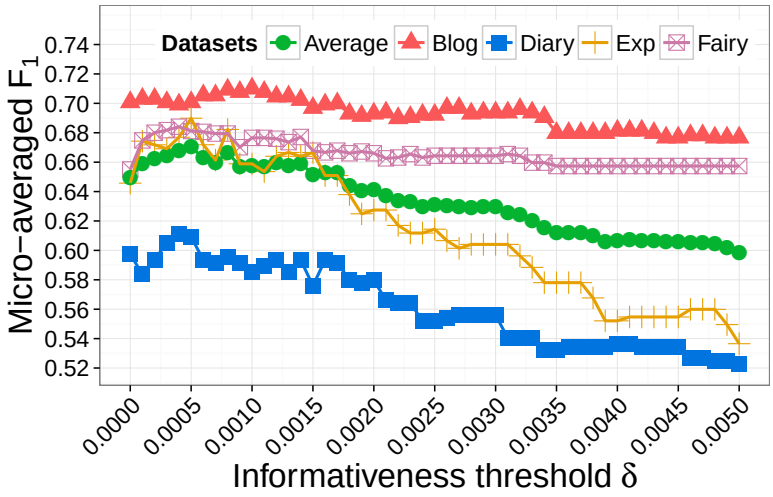

Figure 5.2: Effects of various gap thresholds

instances that satisfy either condition because such instances are likely to contain inconsistent knowledge and therefore are not selected. We increase the informativeness threshold δ from 0 (the minimum informativeness value) and show how it influences results in Figure 5.2. When we increase δ from 0 to 0.0002, the average of micro-averaged F_1 increases from 0.6498 to 0.6621, because we are selecting better tweets of larger informativeness by increasing δ. When δ is between 0.0002 and 0.0008, the average of micro-averaged F_1s on all datasets is above (or very close to) 0.66. When we increase δ beyond 0.0008, the general trend is that F_1 starts decreasing on almost all the datasets, while the decreasing is faster on Diary and Exp. By further increasing δ, we make the bar for selecting informative tweets so high that we cannot obtain enough number of informative tweets.

Evaluations of Instance Selection Strategies

We evaluate the strategies on the selection of informative instances to show the effectiveness of selecting instances out of T^s_{wrong} (instead of T^s). We define several variants of CDS with the following changes: **CDS-ALL** selects instances from T^s; **CDS-CORR** selects instances from T^s that are *correctly* classified by c. **Random** is a baseline approach that randomly selects instances from T^s during each iteration. We let each approach run up to

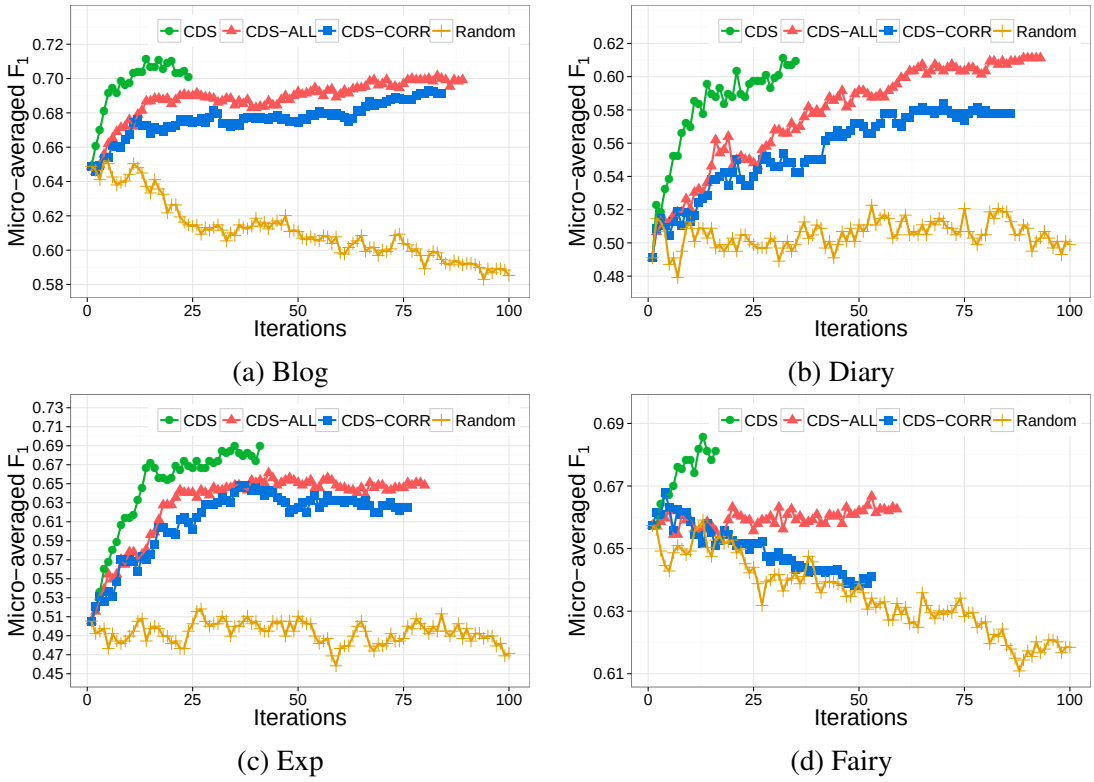

Figure 5.3: Results of applying different strategies to select informative instances on four datasets.

100 iterations and the result remains at the value of the last iteration if one approach meets the stopping condition early.

We show the results of applying these four selection strategies in Figure 5.3. In descending order of the micro-averaged F_1, the strategies rank as follows: CDS >CDS-ALL >CDS-CORR >Random, which is consistent across all datasets, with CDS-ALL (0.6112) performing marginally better than CDS (0.6092) on Diary. Among all the strategies, CDS improves F_1 with the least number of iterations. The reason why CDS improves F_1 faster than CDS-ALL and CDS-CORR is that we feed CDS with instances from T^s_{wrong} which are incorrectly classified by classifier c. Some of these instances contain knowledge that is lacking in the target domain. Since the input of CDS-ALL is a super set of that of CDS, it usually achieves a close result in the end, but it takes far more iterations for CDS-ALL to terminate.

5.5 Conclusions and Future Work

In this chapter, we studied the problem of leveraging automatically-labeled Twitter data to improve emotion identification across different domains via instance selection. We proposed a bootstrapping framework that iteratively selects tweets that are informative about target domains using criteria based on consistency, diversity, and similarity. Quantitative evaluation against baseline approaches show that: (1) our approach performs the best across four target domains; (2) it is superior to select source domain instances that cannot be correctly classified by the target classifier, because these instances are likely to contain the knowledge that is lacking in the target domain labeled data. One interesting direction is to study the characteristics of the tweets posted by verified accounts on Twitter and use them as the source domain dataset, because these tweets might be more formal compared with other tweets and have greater potential to improve emotion identification in target domains where the texts are formal. We plan to explore domain adaptation with multiple source domain data for emotion identification to further improve the performance.

Cursing in English on Twitter

The task of emotion identification that we presented in earlier chapters has applications in different problems such as depression detection, Gross National Happiness estimation, personal happiness index calculation and customer satisfaction measurement. In this chapter, we adapt the proposed algorithm in Chapter 3 to study cursing on social media as one of its applications. Cursing is an emotion-rich behavior that has been studied a lot in the physical world but it remains largely unexplored in online social media. We examine the characteristics of cursing activity on Twitter, involving the analysis of about 51 million tweets and about 14 million users. In particular, we explore a set of questions that have been recognized as crucial for understanding cursing in offline communications by prior studies, including the ubiquity, utility, contextual dependencies, and people factors. We adapt the proposed emotion identification to study people's cursing behavior on Twitter from the emotion point of view. We identify five types of emotions from both cursing tweets and non-cursing tweets with reasonably good precisions. Experimental results empirically confirm the findings by existing psychology studies: cursing is often used to vent out negative emotions.

6.1 Motivation

Do you curse? Do you curse on social media? How often do you see people cursing on social media (e.g., Twitter)? Cursing, also called swearing, profanity, or bad language,

is the use of certain words and phrases that are considered by some to be rude, impolite, offensive, obscene, or insulting (Profanity - Wikipedia, 2013). In this chapter, we use cursing, profanity, and swearing interchangeably. As Jay (2009b) points out, cursing is a "rich emotional, psychological and sociocultural phenomenon", which has attracted many researchers from related fields such as psychology, sociology, and linguistics (Jay, 2009a; Jay and Janschewitz, 2008).

Over the last decade, social media has become an integral part of our daily lives. According to the 2012 Pew Internet & American Life Project report (Pew Internet, 2013), 69% of online adults use social media sites and the number is steadily increasing. Another Pew study in 2011 (Pew Internet, 2011) shows that 95% of all teens with ages 12-17 are now online and 80% of those online teens are users of social media sites. People post on these sites to share their daily activities, happenings, thoughts and feelings with their contacts, and keep up with close social ties, which makes social media both a valuable data source and a great target for various areas of research and practice, including the study of cursing. While the CSCW community has made great efforts to study various aspects (e.g., credibility (Morris et al., 2012), privacy (Almuhimedi et al., 2013)) of social network and social media, our understanding of cursing on social media still remains very limited.

The communication on social media has its own characteristics that differentiate itself from offline interaction in the physical world. Let us take Twitter, for example. The messages posted on Twitter (i.e., tweets) are usually public and can spread rapidly and widely through the highly connected user network, while offline conversations usually remain private among the persons involved. In addition, we may find that more of our actual exchange of words in the physical world happens through face-to-face oral communication, while on Twitter we mostly communicate by writing/typing without seeing each other. Will such differences lead to a change in people's cursing behavior? Will the existing theories on swearing during offline communication in the physical world still be supported if tested on social media?

To address such differences, this chapter examines the use of English curse words on the micro-blogging platform Twitter. We collected a random sampling of all public tweets and the data of relevant user accounts every day for four weeks. We first identified English cursing tweets in the collection, and extracted numerous attributes that characterize users and users' tweeting behaviors. We then evaluated the effect of these attributes with respect to the cursing behaviors on Twitter. Our study aims to improve our understanding of cursing on social media by exploring a set of questions that have been identified as crucial in previous cursing research on offline communication. The answers to these questions may also have valuable implications for the studies of language acquisition, emotion, mental health, verbal abuse, harassment, and gender difference (Jay, 2009b).

Specifically, we examine four research questions:

- Q1 (*Ubiquity*): How often do people use curse words on Twitter? What are the most frequently used curse words?

- Q2 (*Utility*): Why do people use curse words on Twitter? Previous studies (Jay, 2009b) find that the main purpose of cursing is to express emotions. Do people curse to express emotions on Twitter? What are the emotions that people express using curse words?

- Q3 (*Contextual Variables*): Does the use of curse words depend on various contextual variables such as time (when to curse), location (where to curse), or communication type (how to curse)?

- Q4 (*People factors*): Who says curse words to whom on Twitter? Previous research (Jay, 2000; McEnery, 2006) suggest that the gender and social rank of people play important roles in cursing; do they also affect people using or hearing curse words on Twitter?

6.2 Method and Analysis

We first describe how we collect and clean a collection of a random sample of Twitter data. After constructing a cursing lexicon and identifying tweets using the cursing lexicon, we conduct experiments to address the previously proposed four research questions.

6.2.1 Data Collection and Cleansing

Twitter provides a small random sample of all public tweets via its *sample API* in real time [1]. Using this API, we had been continuously collecting tweets for four weeks from March 11th 2013 to April 7th 2013. We kept only the users who specified 'en' as their language in profiles. Further, we utilized the Google Chrome Browser's embedded language detection library to remove non-English tweets[2]. In total, we gathered about 51M tweets from 14M distinct user accounts.

Spam on Twitter may impede the delivery of quality results from data analysis. To examine the spammers in our dataset, a random set of 200 user accounts were selected and manually verified based on the content of tweets and their profile (using the number of friends, followers, etc.) attached with each account. Of the 200 accounts, 5 (2.5%) were identified as spammers, and there were 88 tweets in our dataset from these 5 spammers, accounting for 1.32% of all 6678 tweets posted by these 200 users. On the other hand, we observed that there were some accounts that posted suspiciously frequently, and it could harm our analysis if they were spammers. Thus, we manually verified the top 1,000 accounts which posted most frequently in our dataset, and removed the identified spam accounts and their tweets. Not surprisingly, among the 1,000 accounts, there were 19 spammers in the top 100 accounts, 15 spammers in the following 100 accounts, and then this fraction kept diminishing, with only 3 spammers identified in each of the last two sets of 100 accounts. In total, we removed 68 spammers and 89,556 tweets from our dataset.

[1] https://dev.twitter.com/docs/api/1.1/get/statuses/sample
[2] https://pypi.python.org/pypi/chromium_compact_language_detector/0.2

6.2.2 Cursing Lexicon Coding

To create a lexicon of curse words for this study, we first collected existing curse word lists from the Internet used by native English speakers for cursing on social media. Besides the curse word list (NoSwearing, 2013) that has been used by existing studies (Sood et al., 2012; Xiang et al., 2012), we collected additional curse word lists from (Alvarez, 2013; BannedWordList, 2013; BanBuilder, 2013) to increase the coverage. After merging the above word lists, we found that a few non-curse words were also included, e.g., "sexy". Also, there are some non-English words, e.g., "buceta", which means "pussy" in Portuguese. Moreover, some words can be used in both cursing and non-cursing contexts: "gay" in "you are so gay" conveys cursing, but "gay" in "Bill Clinton urges Illinois to approve gay marriage bill" does not convey cursing. To achieve a high precision in identifying cursing expressions, we eliminated ambiguous words, e.g., "gay" and kept only the words that are most strongly attributed with a cursing connotation.

Specifically, to retain these curse words, we asked two college students who are native English speakers to independently annotate the collected words in the context of social media with the following labels: 1 - the word is mostly used for cursing, 2 - the word can be used for both cursing and non-cursing purposes, 3 - usually the word is not used for cursing, or 4 - I do not know its meaning. Cohen's Kappa between the labels chosen by the two students was 0.5582. In the end, we kept only 788 words that received label '1' from both students to emphasize high precision. Besides correctly spelled words, (e.g., *fuck*, *ass*), the lexicon also included different variations of curse words, e.g., *a55*, *@$$*, *$h1t*, *b!tch*, *bi+ch*, *c0ck*, *f*ck*, *l3itch*, *p*ssy*, and *dik*.

We call a tweet a *cursing tweet* if it contains at least one curse word. Twitter users may use different variations of the same word, so we first simply compare words in a tweet against all the curse words in the lexicon. If there is no match, we remove repeating letters in the words (e.g., fuckk $\rightarrow$ fuck) of a tweet and repeat the matching process. We also convert digits or symbols in a word to their original letters: e.g., 0 $\rightarrow$ o, 9 $\rightarrow$ g, ! $\rightarrow$ i.

Table 6.1: Statistics of overall tweets and cursing tweets per user

Statistics	Min	Max	Mean	Median	Std. Dev.
Overall Tweets	1.0	4124.0	3.56	2.0	8.00
Cursing Tweets	1.0	549.0	1.78	1.0	2.39

Moreover, based on our observations, the following symbols, '_', '%', '-', '.', '#', '\', '’', are frequently used to mask curse words: *f_ck*, *f%ck*, *f.ck*, *f#ck*, *f’ck* $\rightarrow$ *fuck*. We apply the edit distance approach similar to (Sood et al., 2012) to spot curse words with mask symbols. Namely, if the edit distance between a candidate word (*f_ck*) and a curse word (*fuck*) equals the number of mask symbols (1 in this case) in the candidate word, then it is a match. Table 6.1 provides an overview of the per-user count of the number of overall tweets and cursing tweets in our data collection.

To evaluate the accuracy of this lexicon-based method to spot cursing tweets, we drew a random sample of 1,000 tweets, and asked two annotators to manually label them as cursing or non-cursing independently. In the end, there were 118 tweets labeled as cursing tweets for which both annotators agreed on their labels, and the other 882 tweets were labeled as non-cursing ones. We then tested the lexicon-based spotting approach on this labeled dataset, and the results showed that this lexicon-based method achieved a precision of 98.84%, a recall of 72.03%, and an F1 score of 83.33%. As expected, this lexicon-based approach for cursing detection provides a high precision but a lower recall, which is mainly due to the variations in curse words (e.g., due to misspellings and abbreviations) and context sensitivity of cursing. Though we believe that, for this work, a high-precision is preferred and the recall of 72.03% is considered reasonable, more sophisticated classification methods that can further improve the recall remain an interesting topic for future work.

Table 6.2: Cursing frequency over different datasets: Cursing on Twitter is more frequent than that in the other two datasets – 0.80% of all words vs. 0.5% of all words, and 7.73% of all tweets vs. 3% of all utterances

	Mehl and Pennebaker (2003)	Subrahmanyam et al. (2006)	**Our work**
Subject	52 undergraduates	1,150 chatroom users	14 million Twitter users
Sample	4 days' tape recording	12,258 utterance	51 million tweets
Cursing Frequency	0.5% of all words	3% of all utterances	0.80% of all words 7.73% of all tweets

6.2.3 Cursing Frequency and Choice of Curse Words

Prior studies have found that 0.5% to 0.7% of all the words we speak in our daily lives are curse words (Jay, 1992; Mehl and Pennebaker, 2003). Turning to Internet chatrooms, Subrahmanyam et al. (2006) report that 3% of utterances contain curse words. Our comparison of cursing frequencies from different studies is shown in Table 6.2. Compared with existing studies, our estimate of cursing frequency was conducted for a significantly larger population: 14 million Twitter users and 51 million tweets. After removing punctuation marks and emoticons, we find that curse words occurred at the rate of 0.80% on Twitter, which is more than the rate (0.5%) in (Mehl and Pennebaker, 2003). About 7.73% of all the tweets in our collection contain curse words, namely, one out of 13 tweets contains curse words. If we consider one tweet as roughly one utterance, this rate is more than twice the rate (3%) in (Subrahmanyam et al., 2006).

Besides the cursing frequency, we are also interested in the question: Which curse words are most popular? We manually grouped different variations of curse words into their root forms, e.g., *@$$*, *a$$*, $\rightarrow$ *ass*. If a curse word is the combination of two or more words, and one of its component words is also a curse word, then it will be grouped into its cursing component word, e.g., *dumbass*, *dumbasses*, *@sshole*, *a$$h0!e*, *a55hole* $\rightarrow$ *ass*. All 788 curse words were grouped into 89 distinct groups based on the root curse words and

the frequencies of the top 20 words are shown in Figure 6.1. The most popular curse word is *fuck*, which covers 33.57% of all the curse word occurrences, followed by *shit* (15.45%), *ass* (14.66%), *bitch* (10.67%), *nigga* (10.30%), *hell* (3.91%), *whore* (1.84%), *dick* (1.74%), *piss* (1.55%), and *pussy* (1.24%).

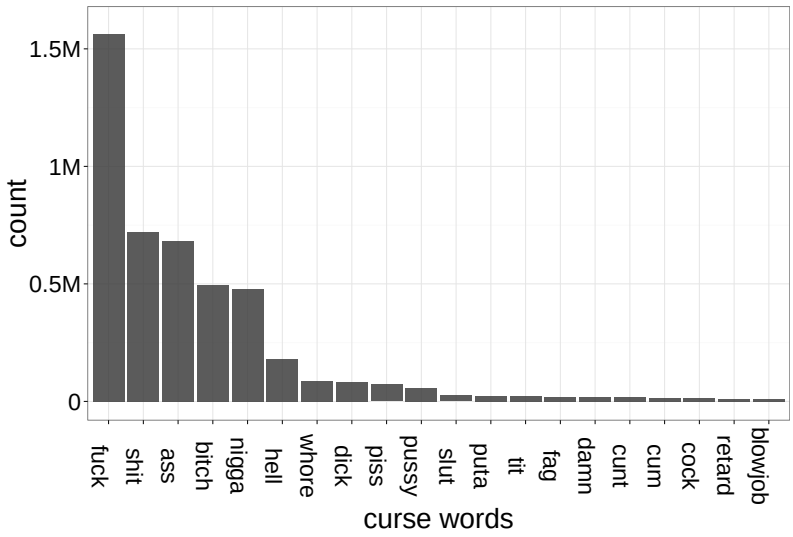

Figure 6.1: Counts of curse words: only top 20 curse words are shown due to space limitation.

Realizing that only a small subset of curse words occurs very frequently, we also draw the cumulative distribution of the top 20 curse words. We find that the top seven curse words – *fuck*, *shit*, *ass*, *bitch*, *nigga*, *hell* and *whore* cover 90.40% of all the curse word occurrences.

6.2.4 Cursing vs. Emotion

Psychology studies (Jay and Janschewitz, 2008) suggest that "the main purpose of cursing is to express emotions, especially anger and frustration." Thus, we aim to explore emotions expressed in cursing tweets and compare them with those in non-cursing tweets. We adapted the emotion identification approach from our prior work in Section 4.3 to automatically detect emotions expressed in tweets. The basic idea is to leverage ending emotion

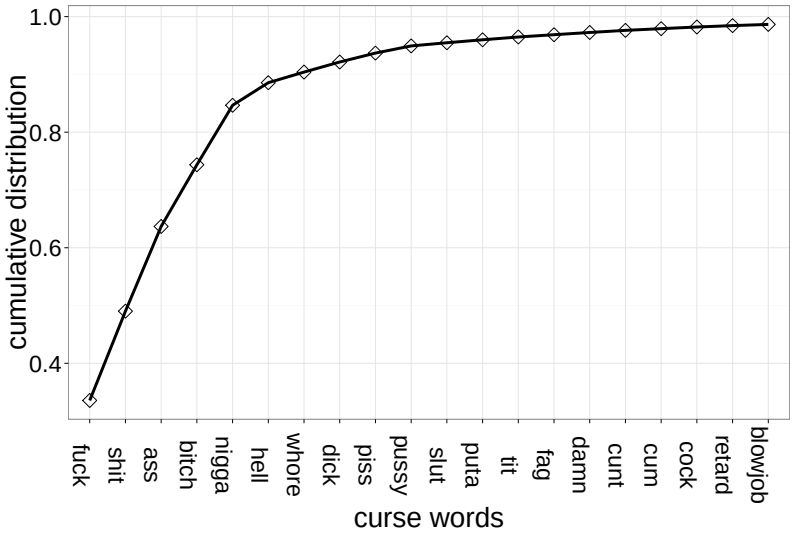

Figure 6.2: Cumulative distribution of curse words: The top 7 curse words cover 90.40% of all the curse word occurrences.

hashtags to automatically create labeled training data. For example, the tweet "And all I need is one fuckin sheet stamped! #rage" will be labeled with emotion *anger* and added into training data after removing the ending emotion hashtag "#rage". In this way, we collected a large number of self-labeled tweets covering seven emotions: *joy*, *sadness*, *anger*, *love*, *fear*, *thankfulness* and *surprise*. We collected about 2 million tweets for training, 250 thousand tweets for testing and nearly 250 thousand tweets for algorithm development, all of which were used in our experiment. We did not apply all the features used in Section 3.4.2. Instead, we applied a combination of unigram, bigram, and LIWC [3] features. LIWC features refer to the percentages of positive and negative emotion words according to the LIWC dictionary. This combination achieved a reasonably good accuracy, very close to the best performance achieved by incorporating more features, according to the feature engineering experiments in Table 3.4.

We trained seven binary classifiers for seven emotions, such that for each emotion e_i, the corresponding classifier C_{e_i} predicts the probability p_j of a tweet t_j expressing the emotion e_i: $p_j = C_{e_i}(t_j)$. Specifically, we trained a binary classifier C_{e_i} by selecting all

[3] http://www.liwc.net

the tweets of a specific emotion (e.g., *anger*) and randomly selecting the same number of tweets that do not express this emotion (the tweets may express other emotions such as *sadness*, and *love*). For a given tweet t_j, we applied all seven classifiers. If a classifier C_{e_i} provides the highest probability that t_j expresses the emotion e_i among all the classifiers, and this probability is greater than or equal to the predefined threshold τ, we conclude that the emotion e_i is expressed in t_j; otherwise, t_j is labeled as *other*.

$$\begin{cases} Emotion\ e_i & i = \arg\max_k \{C_{e_k}(t_j)\}\ and\ C_{e_i}(t_j) >= \tau \\ Other & Otherwise \end{cases} \tag{6.1}$$

Intuitively, the higher the value of τ, the higher the precision of identifying the seven emotions, at the expense to recall. To find a τ that provides high precision and reasonable recall, we tried a series of τ values in the development dataset: starting from 0, with an increment of 0.02, and ending at 1.0. We plot the precisions and recalls of individual emotion classifiers as well as the combined classifier in Figure 6.3. As we can see from the figure, with the increasing value of τ, the precision increases, while the recall decreases. Emotion classifiers that are on the upper right perform better than those on the lower left. Since we are interested in only the emotions that we can predict with a high precision, we skipped detecting emotion *surprise* and *fear*, for which the highest precision is less than 65%. We selected $\tau = 0.88$ for later emotion identification, with which the combined classifier achieves good a precision while retaining a reasonable recall among all the values of τ we have tested using the development dataset.

We then trained the classifiers on the training dataset and applied the combined classifiers to the testing dataset. The results are shown in Table 6.3. Among all the five emotion categories, the precision ranges between 56.04% (*thankfulness*) and 84.66% (*anger*), the recall ranges between 37.22% (*love*) and 57.01% (*thankfulness*), and the F1-score ranges between 45.86% and 68.11%. The combined classification achieves a micro-averaged pre-

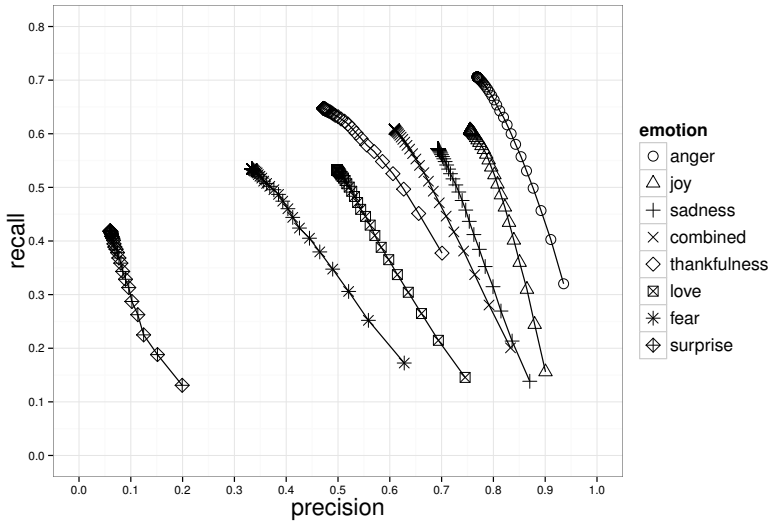

Figure 6.3: Performance of emotion identification on the development dataset

Table 6.3: Performance of emotion identification on the testing dataset. * micro-averaged metrics. (*Surprise* and *Fear* were dropped because we couldn't detect it with a reasonably high precision on the development dataset)

Emotions	Precision (%)	Recall (%)	F1-score (%)
anger	84.66	56.97	68.11
joy	82.77	44.81	58.14
sadness	76.05	39.34	51.86
love	59.72	37.22	45.86
thankfulness	56.04	57.01	56.53
combined	76.17*	46.07*	57.41*

cision of 76.17%, a micro-averaged recall of 46.07%, and a micro-averaged F1-score of 57.41%. This performance is quite reasonable considering that it is a multi-class classification problem.

Finally, we applied the combined classifier to the 51 million cursing tweets, and obtained the emotion distributions on both cursing and non-cursing tweets, which is shown in Figure 6.4. Not surprisingly, cursing is associated with negative emotions: 21.83% and 16.79% of the cursing tweets express the emotions sadness and anger, respectively. In contrast, 11.31% and 4.50% of the non-cursing tweets express sadness and anger emotions, respectively. This can be explained by the fact that curse words are usually used for vent-

ing out negative emotions, especially anger and sadness. However, we also find that 6.59% of cursing tweets express love. One reason is that curse words can be used to emphasize emotions, including positive ones such as love: e.g., "**fucking** love you." Another reason is that certain curse words are used between close friends as a playful interaction, e.g., close female friends call each other *whore*. To better understand how curse words are used to express emotions in tweets, we list some example cursing tweets in Table 6.4.

In addition, we also examined the frequency of cursing in each type of emotional tweet. As we expected, 23.82% of angry tweets and 13.93% of sad tweets contain curse words, which are much higher than the rate of curse words in other emotional tweets, such as love (4.16%), thankfulness (3.26%), and joy (2.5%), or the remaining tweets that are not labeled with any of these five emotions (6.39%). This again shows that curse words are often used to express negative emotions.

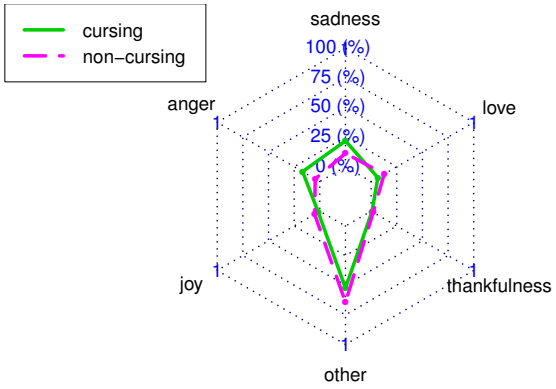

Figure 6.4: Emotion distributions in both cursing and non-cursing tweets. This shows that curse words are usually used for venting out negative emotions: 21.83% and 16.79% of the cursing tweets express the emotions sadness and anger, respectively; in contrast, 11.31% and 4.50% of the non-cursing tweets express sadness and anger emotions, respectively

Table 6.4: Example tweets in which curse words are used to express different emotions.

sadness	"Where da **fuq** is the sun at, this weather is so #depressing" "My life fell apart a long **ass** time ago.. So everythings normal i guess."
anger	"Soo **pissed** off" "People laugh when I say I work at McDonald's. And I say, **bitch** at least I have a job! At least I don't bother my parents asking them for $$$"
love	"Across the ocean, across the sea starting to hate the **fucking** distance between Justin Bieber and me." "@user you little **whore** TAKE ME WITH YOU" "Dear Marilyn Manson, I **fucking** love you and your music. The end."

6.2.5 Cursing vs. Time

A previous study (Kamvar and Harris, 2011) has shown a marked difference in emotions (e.g., stress, happiness) expressed between weekdays and weekends, or between morning and night. Similarly, we investigate the relationship between cursing and two types of time periods: times of a day and days of a week. For each tweet, Twitter provides a timestamp based on UTC timezone, indicating when the tweet was posted. However, it makes more sense to use local time when the tweet was posted, so we calculated the corresponding local timestamp for every tweet whose sender has a specified timezone in his/her profile. In Figure 6.5, the lines with triangles and crosses stand for the volumes of overall tweets and cursing tweets, and the line with circles stands for the ratio of cursing tweets to overall tweets. A flat segment of the line with circles suggests the cursing ratio is stable – the increment of cursing tweets keeps pace with that of overall tweets. A rising line segment with circles suggests that the increment of cursing tweets outpaces that of overall tweets. A falling line segment with circles suggests that the increment of cursing tweets is outpaced by that of overall tweets.

We have the following interesting observations from Figure 6.5. First, the pattern of overall tweet volume fits humans' diurnal activity schedule: it starts rising at 5 am when people get up at the beginning of a day. From then, it keeps rising, and reaches a small

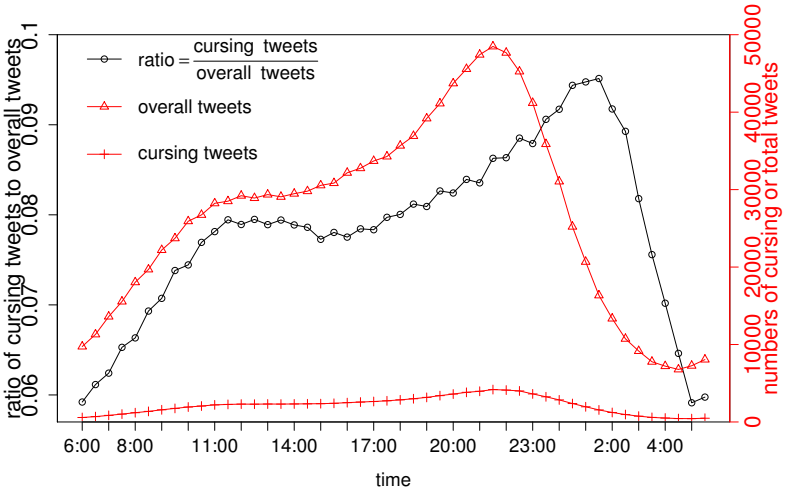

Figure 6.5: Cursing volume and ratio at different times of a day

peak around lunch time. It keeps rising until it reaches the peak of the day around 9 pm, after which people start preparing to go to sleep. Second, cursing is persistent: the black cursing ratio line with circles always stays above 0, suggesting that people curse all the time throughout the day. Third, the increment of cursing outpaces the increment of overall tweet volume during most of the day time: people curse more and more as they go through the day! In particular, there are two sharp rising slopes: 6 am - 11 am and 3 pm - 1:30 am. We speculate that Twitter users being in a good mood during lunch contributes to the flat ratio line segment between 11 am - 2 pm (lunch time). It seems that midnight to 1:30 am is the *high time* for cursing. After that, the volume of cursing tweets decreases faster than the overall tweets.

We now explore the popularity changes of the top seven curse words (refer to Figure 6.1) at different times of a day to gain more insights. We define the *relative frequency* for a curse word as its total number of occurrences in any tweet divided by the total number of tweets in a predefined time window. Three representative time windows are selected: 12 am - 2 am, 5 am - 7 am and 12 pm - 2 pm. We observe that the relative frequencies for almost all of the top seven curse words keep increasing from 5 am - 7 am to 12 pm - 2 pm and from 12 pm - 2 pm to 12 am - 2 am. On average, from 5 am - 7 am to 12 am

- 2 am, the relative frequencies of the top seven curse words have increased by 59.60%. In descending order of their relative increase of relative frequencies, the top seven curse words rank as follows: *ass* (86.33%), *nigga* (78.17%), *bitch* (61.03%), *shit* (56.90%), *fuck* (50.85%), *whore* (34.54%) and *hell* (23.69%).

To explore how people curse during different days of a week, we plot the ratio of cursing tweets to total tweets each day for four weeks, separately, in Figure 6.6. The general trend is that users start with relatively high cursing ratios on Mondays, Tuesdays, and Wednesdays, then the ratios keep decreasing on the following three days, and reaches the lowest point on Saturdays. Then they start rising up on Sundays. To see the general trend clearly, readers are referred to see the four-week average ratio in the plot. Although we observe this general pattern across four weeks, we are still unclear about the reason. We are interested in the popularity changes of the top seven words during different days of a week, similar to those at different times of a day. We select the following two time windows: Monday-Tuesday and Friday-Saturday. On average, from Friday-Saturday to Monday-Tuesday, the relative frequencies of the top seven curse words have increased by 10.36%. In descending order of their relative increase of relative frequencies, the top seven words rank as follows: *bitch* (15.15%), *shit* (13.55%), *nigga* (12.41%), *ass* (10.37%), *whore* (10.30%), *hell* (7.53%) and *fuck* (7.16%).

6.2.6 Cursing vs. Message Type

Tweets can be grouped into different message types and we are curious whether users curse differently in different types of tweets. Specifically, *retweet* refers to a tweet that is simply a re-posting of a tweet from another user. If a user receives a tweet from another user, and this user clicks on the *reply* button to write a new tweet to reply to this tweet, then this newly posted tweet is called a *reply*. If a user starts sending a tweet to another user, and this tweet is not a reply to any other tweet, we name it a *starter*. If a tweet mentions another

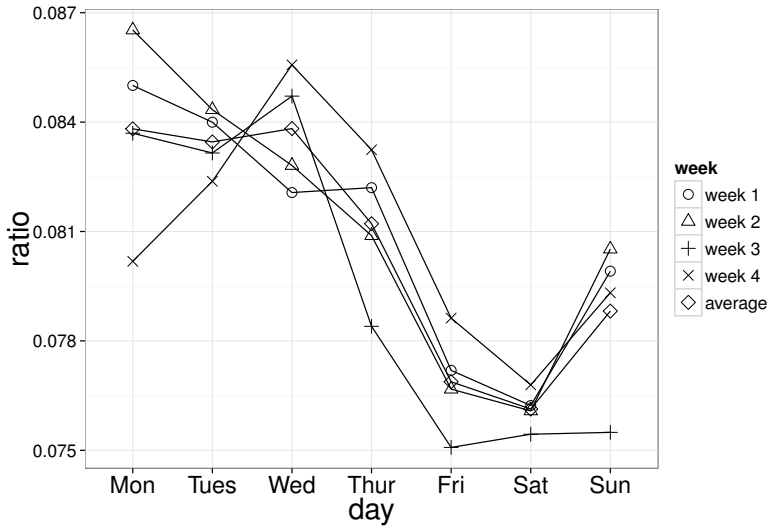

Figure 6.6: Cursing ratios in different days of a week

user, and it is neither a *reply* nor a *starter*, we call it a *mention*. If a tweet does not belong to any of the above categories, it is an *update*.

We plot the ratio of cursing tweets in each message category in Figure 6.7, where the black horizontal line stands for the average ratio of cursing tweets to all the tweets. It is interesting to note that although we see quite a bit of cursing messages on Twitter in general, when the messages are sent to other users, the cursing ratios are below the average ratio. The ratio of cursing tweets in *starters* is 3.93%, which is only 51.01% of the average cursing ratio. This suggests that users perform self-censorship to some extent when they directly write to other users. When they post updates about themselves or simply mention other users' names, they do not pay as much attention to the use of curse words. Another interesting observation is that the highest cursing ratio occurs in *retweets*. Sood et al. (2012) find that "*profane comments are more popular or more widely read than non-profane comments*" by receiving thumb ups and downs in Yahoo! Buzz. We are interested in assessing whether the use of curse words can help draw other users' attention so as to be retweeted. However, Pearson's product-moment correlation analysis between whether a tweet has curse words and the number of times it is retweeted suggests a negligible correlation: $r = -0.00154, p < 2.2e-16$. We perform the same analysis on whether a

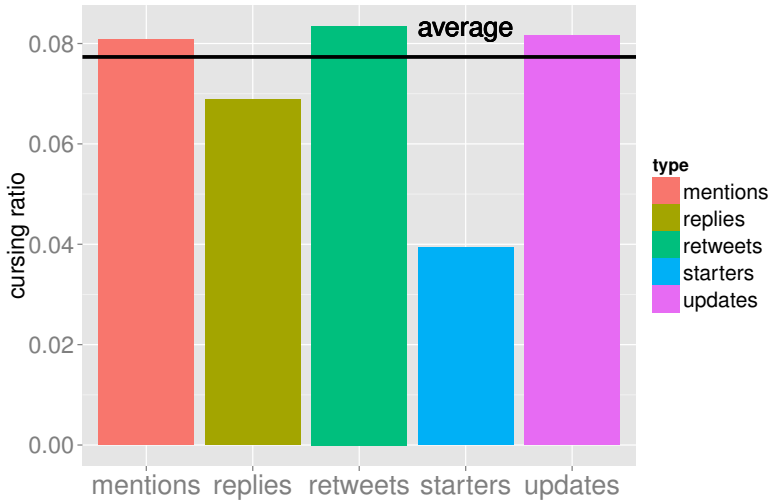

Figure 6.7: Cursing ratios in different types of messages

tweet has curse words and whether it is retweeted, and find a stronger but still very weak positive correlation: $r = 0.03366, p < 2.2e - 16$. Similarly, a negligible correlation is also observed between whether a tweet has curse words and whether it has been favorited: $r = 0.005436, p < 2.2e - 16$.

6.2.7 Cursing vs. Location

Location also affects the way people curse in the physical world. Cameron (1969) find that people curse more at parties than they do at work places. Jay (1992) has a similar discovery obtained by investigating cursing frequency at different campus locations: people tend to curse more in relaxed environments (e.g., college dorms vs. Dean's office). Compared with physical world conversations, tweets are posted in digital world: a user can use curse words without being noticed by surrounding people. Do physical locations still affect Twitter users' cursing frequency? Luckily, a Twitter user's location can be inferred via geo-enabled tweet feature. This feature provides the latitude and longitude of the user's location, along with the usual tweet content.

Given a pair of latitude and longitude, Foursquare's venue search API [4] returns a list of nearby venues, as well as the distances. If the distance from the user's location to the nearest venue is less than 50 meters, we assume the user posted the tweet from that venue. For every venue, we retrieve its immediate category and upper category from Foursquare, e.g., a venue is under Asian Restaurant (immediate category), and Asian Restaurant is under Food (upper category). We made very few changes to the Foursquare category hierarchy to reduce ambiguous information, e.g., we deleted *other places* from the hierarchy, because we have no idea what the name suggests. We also removed categories if they have very sparse tweets. Table 6.5 shows the different categories of venues, the raw number of cursing tweets and the ratio of cursing tweets to all the tweets sent from venues of the same category.

We have the following observations: a) The pattern of more swearing in more relaxed environment still holds, e.g., cursing ratios in a descending order are: Residence (7.08%) $>$ Shop & Service (6.41%) $>$ Nightlife Spot (6.37%) $>$ Entertainment & Recreation (5.71%) $>$ Professional Places (5.64%) $>$ Travel & Transport (5.34%). However, the gaps are much less than those in the physical world, partly due to the fact that communications happen in digital world. b) Two exceptions, College Academic Place and High School, have very high cursing rates. This suggests that young high school and college students tend to use more curse words, even in educational places. c) We speculate that users are usually in a good mood while out in the nature, and that is why its cursing ratio is the lowest (4.97%) among all the venues.

6.2.8 Cursing vs. Gender

Another interesting question about cursing is "*who says curse words to whom*". We first explore the gender factor in this section, and then discuss the effect of social rank in the

[4] https://developer.foursquare.com/docs/venues/search

Table 6.5: Cursing ratios from different places. Field: lakes, beach, mountain, etc.; Travel & Transport: train, plane, ferry, etc.; Professional Places: police station, city hall, office, etc.; College Academic Place: law school, engineering building, math building, etc.; Residence: home, residential building, hotel, etc.

Venues	Cursing Tweets (#)	Cursing Tweets (%)
Field (Nature)	380	**4.97**
Travel & Transport	621	5.34
Food	2814	5.35
Professional Places	2020	5.64
Entertainment & Recreation	1305	5.71
Arts	195	5.77
Nightlife Spot	1063	6.37
Shop & Service	3036	6.41
College Academic Place	1155	6.45
Residence	2198	7.08
High School	339	**9.36**

next section. Prior studies have found that gender affects the cursing frequency and the choice of curse words; in addition, people curse more in same-gender contexts than in mixed-gender contexts (Mehl and Pennebaker, 2003; McEnery, 2006; Jay and Janschewitz, 2008; Pilotti et al., 2012). We explore whether these hypotheses still hold when people send messages to each other on Twitter. In order to study the gender difference, we first applied an algorithm to recognize the gender of users in our data collection. A person's gender can be revealed by his/her first name: Linda, Lisa, Betty, etc. are usually females' names; John, Paul, William, etc. are male names. US Census Bureau [5] provides 1,219 most popular male names and 4,275 most popular female names. We calculated the "maleness" or "femaleness" by dividing the number of male/female people using this name by the overall size of the population. If a name has a high female percentage and a low male percentage, e.g., Mary: Male (0.009%), Female (2.629%), then the corresponding person is mostly female. If the female and male percentages of a name are close, e.g., Morgan: Male 1.8%, Female: 2.2%, it suggests that this name is usually used for both genders. If a name is missing from the male (resp., female) name list, we take it as a female (resp., male).

[5] https://www.census.gov/genealogy/www/data/1990surnames/names_files.html

Table 6.6: Cross-gender cursing statistics. Statistics of each row are drawn on randomly sampled 100K tweets. Reported *cursing ratio* in each row is the percentage of cursing tweets out of all the tweets within each corresponding group.

Sender	Recipient	Cursing Tweets (#)	Cursing Ratio (%)
F	M	3,808	**3.81**
F	F	3,977	3.98
M	F	4,192	4.19
M	M	5,483	**5.48**

Our algorithm will label a user as female when the female percentage divided by the male percentage is greater than or equal to four; if the male percentage divided by the female percentage is greater than or equal to four, the user will be labeled as male; otherwise, the user will be labeled as unknown.

Overall, this algorithm identified 4,639,204 females and 3,826,701 males in our Twitter user collection. Recall that previously we grouped tweets into five categories: *mention*, *reply*, *retweet*, *starter*, and *update*. Here we consider only *reply* and *starter*, since they represent targeted messages between Twitter users with an explicit message sender (*who*) and recipient (*to whom*) specified. These messages are further divided into four groups based on gender – *female to female*, *male to female*, *female to male*, and *male to male*. To make results comparable, we randomly sampled 100K tweets from each of these four groups and the statistics are shown in Table 6.6.

Comparing the same-gender contexts (*F to F* and *M to M*) with the mixed-gender contexts (*F to M* and *M to F*) in Table 6.6, we observe that people are more likely to use curse words within the same-gender context, and this tendency is more obvious when the message senders are males (5.48% vs. 4.19%). This is consistent with the findings in prior studies (Jay and Janschewitz, 2008; Pilotti et al., 2012) on offline communications. Moreover, Male-to-Male communication has the highest cursing ratio: 5.48%, while Female-to-Male has the lowest cursing ratio: 3.81%.

Table 6.7: The frequency of curse words out of 100K tweets posted or received by males and females. *** p≤0.001, ** p≤0.01

Word	F→F	F→M	M→F	M→M	χ^2
fuck	1236	1284	1359	**2069**	308.89***
shit	670	661	831	**1159**	195.61***
nigga	119	171	201	**338**	126.59***
bitch	**475**	273	281	298	83.46***
hell	334	315	349	**532**	79.43***
dick	54	67	80	**137**	47.49***
cunt	22	24	26	**60**	29.70***
fag	20	29	26	**58**	25.83***
pussy	25	25	33	**60**	23.13***
slut	**50**	30	24	17	19.99***
ass	**1091**	1030	970	922	16.07**
bastard	9	14	18	**32**	16.04**
piss	95	91	107	**143**	15.41**
cock	14	15	25	**37**	15.15**
whore	**92**	68	68	50	12.82**

Regarding the preference of curse words, out of the randomly sampled 100K tweets for each of the four groups (see Table 6.7), we also find clear difference between females and males. There are a set of words that are used significantly more often by males than by females, such as: *fuck*, *shit*, and *nigga*. Some other words are significantly overused by females, such as *bitch* and *slut*. It is also interesting to observe that such differences are more apparent between two same-gender contexts – *F to F* vs. *M to M*. This suggests that the genders of both "*who*" and "*whom*" matter in the choice of curse words.

6.2.9 Cursing vs. Social Rank

We now look into the relationship between social rank and cursing behavior. Within a society, it is expected that the higher the social rank of a person, the less cursing the person performs (Jay and Janschewitz, 2008). We used the number of followers on Twitter as an approximation to social rank in the digital world. We sorted both senders and recipients

Table 6.8: Cursing Ratio vs. Social Ranking (followers) for both senders and recipients. μ population mean, σ standard deviation. For senders and recipients, we show statistics regarding their posted and received tweets, respectively

User Group	Sender			
	Cursing Tweet (#)	Cursing Ratio(%)	$\mu_{followers}$	$\sigma_{followers}$
top 1%	146,035	5.98	67,810	408,228.8
1% - 10%	847,467	**8.78**	1,923.00	1,481.60
10% - 40%	1,744,258	**8.75**	400.10	148.20
40% - 90%	1,116,645	6.62	101.60	60.58
90% - 100%	77,523	4.00	2.30	2.91
User Group	Recipient			
	Cursing Tweet (#)	Cursing Ratio(%)	$\mu_{followers}$	$\sigma_{followers}$
top 1%	49,069	3.91	155,000	650,825.3
1% - 10%	101,983	6.11	3,764.00	3,744.18
10% - 40%	289,035	**7.96**	565.70	219.82
40% - 90%	258,984	6.26	172.20	71.85
90% - 100%	28,348	4.56	29.91	15.77

based on the descending order of their number of followers, and then divided them into five groups: top 1% (who have the highest numbers of followers), then followed by 1% - 10%, 10% - 40%, 40% - 90%, 90% - 100%. In Table 6.8, we show the raw numbers of posted/received cursing tweets, the ratio of posted/received cursing tweets out of overall tweets, and the mean and standard deviation of followers that the group of users have within each sender/recipient group.

The top 1% of senders do curse, but it is less than what we expected. We also observe bell-shaped distributions in cursing ratios for both senders and recipients: the middle sender groups (1%-10% and 10%-40%) curse the most, while the middle recipient group (10%-40%) receive tweets with the highest cursing ratio. Senders from the bottom group, who may have recently joined Twitter, and have very few followers (mean: 2.3), curse the least among all sender groups. Turning to recipients, the cursing ratio among tweets received by the top 1% group, is the lowest across all recipient groups: these popular users receive a lot of friendly messages from their fans, e.g., "@Harry_Styles follow me

Table 6.9: The frequency of curse words out of 100K tweets based on the social rank (follower counts) of senders. χ^2 results are based on the comparison of frequencies of each word across different sender groups. *** p≤0.001 for all the values in this column

Word	top 1%	1-10%	10-40%	40-90%	90-100%	χ^2 * **
fuck	2621	3306	**3399**	2814	1265	1093.81
ass	744	**1624**	1607	1027	675	738.11
nigga	563	**1354**	1131	564	696	603.74
shit	986	1588	**1614**	1221	668	534.82
bitch	779	**1224**	1095	763	596	301.16
cock	30	24	19	22	**129**	199.26
blowjob	24	16	15	16	**89**	128.56
dick	178	**203**	169	130	64	78.92
piss	98	119	**159**	148	54	60.98
whore	167	**205**	183	136	95	46.85
pussy	151	**160**	117	76	101	40.18
hell	286	358	**375**	357	253	34.73
slut	**71**	41	50	54	25	23.79

Table 6.10: The frequency of curse words out of 100K tweets based on the social rank (follower counts) of recipients. χ^2 results are based on the comparison of frequencies of each word across different recipient groups. *** p≤0.001, ** p≤0.01 * p≤0.05

Word	top 1%	1-10%	10-40%	40-90%	90-100%	χ^2
ass	618	1521	**2284**	1590	1034	1119.08***
nigga	243	674	**892**	471	266	599.30***
shit	628	1089	**1428**	1090	774	387.95***
fuck	1556	1875	**2330**	2023	1507	250.08***
bitch	350	533	**643**	458	346	136.78***
hell	244	431	**522**	434	370	105.54***
pussy	62	**88**	71	52	39	22.20***
whore	92	128	**145**	102	92	20.08***
slut	**56**	27	32	35	24	18.24**
dick	124	133	**134**	110	84	14.80**
piss	65	80	96	**107**	93	11.82*

babe<3", "@NiallOfficial I can't sleep :("

Besides the cursing ratios in tweets that are posted/received by different user groups, we are also interested in the curse word choices across all groups. To make the results comparable, we randomly sampled 100K posted tweets from each sender group and counted the corresponding frequencies of curse words in Table 6.9. We did the same to all the tweets received by different recipient groups in Table 6.10. We observed that the same word can be used at different rates across groups: the 10-40% sender group used *fuck* 3,399 times out of 100K posted tweets, while the 90-100% sender group used it only 1,265 times; the 10-40% recipient group received *ass* 2,284 times out of 100K received tweets, while the top 1% recipient group only received it 618 times. We found that, for the same word, its post/received volumes usually achieve the highest frequencies in the 1-10% and 10-40% for sender groups, and 10-40% for recipient groups with a few exceptions. The reason why top 1% sender group used *slut* word the most is because there are a few popular Twitter accounts that posted funny tweets about the word *slut*. We also found some porn accounts in 90-100% sender group that aggressively posted porn links, which explains the peaks for the words *cock* and *blowjob* in this group. The reason why top 1% recipient group received more tweets containing *slut* is because some fans like to call celebrities *slut* regardless of their gender for fun, e.g., "@taylorswift13 slut", "@Harry_Styles slut drop on my follow button :)))))))"

6.3 Limitations

This study is limited in several ways. Firstly, our exploration is based on a random sampling of tweets posted by Twitter users, and our results may be biased towards these users, who may not statistically represent users in other social media websites or mirror overall population in the real world. Thus our findings may turn out to be different on other datasets or social media platforms. Though the findings may not be generalized beyond

Twitter, the analysis framework can be applied to cross-platform studies that we would like to pursue in our future work. Secondly, with such a large amount of data, it is impossible to manually label all the tweets/users. Because of this, we employed automatic approaches to labeling, profanity, emotion, gender, etc. Though these techniques have been used in many studies, they are not perfect. It is important to note that we always choose precision over recall when designing these techniques. Further improving any of these approaches would be an interesting topic for future research. Moreover, in a few tasks, we relied on self-reported data, such as users' names and geo-locations. Using self-reported data may lead to bias toward users who opt to share such information. Further, we do not segment Twitter user accounts into different types, e.g., celebrity, media, organization or regular personal accounts. Users from these different categories may curse in varied manner, and it would be interesting to examine the differences. Other topics as for extension of this study are the comparative study of native speakers and non-native speakers in using curse words, and the explorations of cursing behavior in languages other than English. Finally, this work is mostly descriptive, which provides insights on the *what* aspect of the cursing phenomenon on Twitter. In order to achieve a deeper understanding on *why*, e.g., why people choose particular curse words they use, and why people's cursing behaviors depend on certain contextual variables, user surveys and qualitative analysis will be needed.

6.4 Conclusions and Future Work

In this chapter, we investigated cursing, an emotion-rich behavior that remains largely unexplored in social media. We studied the use of curse words in the context of Twitter based on the analysis of randomly collected 51 million tweets and about 14 million users. In particular, we explored four questions that have been identified as important by the prior swearing studies in the areas of psychology, sociology, and linguistics.

Regarding the question of *ubiquity* of cursing on Twitter, we examined the frequency

of cursing and people's preference in the use of specific curse words. We found that the curse words occurred at the rate of 0.80% on Twitter, and 7.73% of all the tweets in our dataset contained curse words. We also found that seven most frequently used curse words accounted for more than 90% of all the cursing occurrences. The second question we studied is the *utility* of cursing, especially the use of cursing to express emotions. Based on our research in previous chapters, we built a classifier that identified five different emotions from tweets – *anger*, *joy*, *sadness*, *love*, and *thankfulness*. Based on the classification results, we found that cursing on Twitter was most closely associated with two negative emotions: *sadness* and *anger*. However, curse words could also be used to emphasize positive emotions such as *joy* or *love*.

Prior studies suggest that cursing is sensitive to various contextual variables. We focused on examining three contextual variables regarding *when*, *where*, and *how* the cursing occurs. We found that the pattern of overall tweet volume matches people's diurnal activity schedule, and people curse more and more after they get up in the morning till the hours for sleep arrive at the night. Our study of the relation between cursing and message types suggests that users perform self-censorship when they write directly to other users. We found that users do curse more in relaxed environments, but the differences across different environments are very small, partly due to the fact that Twitter messages are posted in a virtual and digital world.

The last question we tried to investigate is concerned with *who says curse words to whom*. We examined the gender and social rank factors and how they might affect people's cursing behaviors on Twitter. Our results support the findings from prior studies that gender and social rank relate to people's propensity to curse and the choice of curse words. Specifically, men curse more than women, men overuse some curse words different from what women use and vice versa, and both men and women are more likely to curse in the same-gender contexts. Turning to social rank, high rank users do curse less than most low rank users; the ratios of using/receiving curse words achieve the highest numbers in the

1%-10%, 10%-40% sender groups, and the 10%-40% recipient group.

The above-mentioned study received wide media coverage such as: time.com (Steinmetz, 2014), Fast Company's Co.Exist (Leber, 2014), and Gizmodo (Aguilar, 2014). As future work, we want to explore the personal well-being of Twitter users. For example, how do people's emotions change over the times of a day or days of a week? Which things keep people happy? Can we spot users who are more vulnerable to negative emotions, e.g., depression, anger? Can we combine different emotion signals into one emotion index number that can be used to track people's overall well-being?

Conclusions

In this chapter, we summarize the findings of this dissertation and talk about some interesting future research directions.

7.1 Summary

Emotions are both prevalent in and essential to most aspects of our lives. With the rapid growth of emotion-rich textual content, such as microblog posts, Facebook posts, blogs posts, and forum discussions, such content can be used to unobtrusively identify and track people's emotions expressed in text, which has great applications in suicide prevention, work performance, personal happiness, personal retrospection, and customer satisfaction. Therefore, there is a great need to develop algorithms and techniques to identify people's emotions from text.

In this dissertation, we conducted in-depth research on the problem of identifying people's emotions from text. By casting this problem as a multi-class classification problem, we first analyzed the contributions of a variety of different features (n-gram, knowledge-based, syntactic, context, information-specific and instruction-specific features) on two datasets: suicide notes and Twitter data. After realizing that the supervised classifier was not effective at gleaning features from minority emotion classes with sparse instances on suicide notes, we proposed an algorithm to automatically spot beneficial features to construct a rule-based classifier to complement the supervised classifier.

A large training dataset for emotion identification can cover different emotional moments in people's daily lives. However, most of the existing labeled emotion datasets are relatively small, because it is time-consuming and error-prone to annotate a sentence with the most appropriate label out of multiple emotion labels. To solve this problem, we proposed an approach to automatically collect self-labeled emotional tweets by applying emotion hashtags to filter big 'Twitter data.' After applying simple filtering heuristics to the collected data, we effectively improved its label quality. Applying the proposed approach, we created one of the largest labeled datasets for emotion identification: about 2.5 tweets covering 7 emotion categories, which provides comprehensive coverage of various emotional situations in our daily lives. Experimental results show that the size of the training data plays an important role in increasing the classifier performance.

After collecting a large number of self-labeled tweets, we studied how to leverage these tweets to improve emotion identification in target domains where labeled instances are relative small, which is one of the first attempts to tackle the domain adaptation problem for emotion identification. We proposed a bootstrapping framework that iteratively selects and adds informative tweets to the target domain training data. We proposed an informativeness scoring function for source instances using criteria based on consistency, diversity, and similarity. Experimental results show that the proposed framework and informativeness scoring function improve over the baseline approaches in four different domains.

As an application, we adapted the proposed algorithm for emotion identification in Chapter 3 to study an emotion-rich behavior in social media, cursing, and received wide media coverage. We identified five types of emotions from both cursing tweets and non-cursing tweets with reasonably good precisions, and experimental results empirically confirmed the findings by existing psychology studies: cursing is often used to vent out negative emotions. We also explored a set of crucial questions for understanding cursing, including: ubiquity, utility, contextual variables and people factors.

7.2 Future Work

The above-described work is a step down the road of ultimately applying emotion identification to improve our daily lives. There are many research directions that worth exploring in the future.

Multi-label Emotion Classification: Currently, we are under the assumption that there is at most one emotion expressed per sentence/tweet. However, this assumption may not always be held and some sentences may have multiple emotion labels, because people may express a mixed of emotions in one sentence. For example, some sentences in the suicide notes dataset in chapter 3 were annotated with multiple emotions. Thus, it will be valuable to identify more than one emotion from a sentence as future work to provide more accurate results.

Personalized Emotion Identification: In the situation of having half a glass of water, pessimistic people may feel sadness because it is half empty, while optimistic people may feel joy because it is half full. Emotions are very subjective and different people may have different emotion responses towards the same situation. For example, in Table 5.1, tweet #2 and tweet #3 describe a very similar situation, but they were annotated with different emotions (sadness and anger) by their authors. We can collect a user's history posts in social media and leverage them to personalize emotion identification for this specific user.

Context-aware Emotion Identification: The context of a situation can help us better understand the situation and therefore improve the emotion identification performance. Specifically, social media provides rich meta-data that can be used to glean information about the context. For example, what is the user's location? What time is it? Is it a weekday or weekend? Is it a holiday? Is it a festival? What is the weather like? What is the age of the user? What is the gender? How many followers does the user have? If the user is chatting with another user, are they old friends or this is the first time they talk to each other? By leveraging such information, we can draw a more detailed context of the situation or event described in a post. Thus, it is very beneficial to glean as much context

as possible to improve emotion identification.

Large-scale Emotion Classification: In Chapter 4, we have shown that millions of self-labeled tweets can be collected by using the emotion hashtags. We have also shown that the accuracy of emotion identification gets improved as more training data is employed. However, with larger amount of training data, it takes a lot of time to train a model on a single machine. Recently, distributed machine learning, such as: Spark MLlib (Meng et al., 2015), has been developed to handle a large amount of training data. It is very useful to leverage distributed machine learning to make full and fast use of the large number of self-labeled emotional tweets.

Personal Happiness Index: As an application, it is very helpful to apply emotion identification to people's posts in social media and aggregate the expressed emotions into a happiness index score. For a user, we can plot the happiness index chart for personal retrospection. For example, in which month is the user the happiest? Is the user happier this year than last year? If not, what happened? Is there a dramatic shift of the happiness index?

Bibliography

Mario Aguilar. Science reveals everything you've ever wondered about cursing on twitter. http://gizmodo.com/science-reveals-everything-youve-ever-wondered-about-c-1525868135, February 2014.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 579–586. Association for Computational Linguistics, 2005.

Hazim Almuhimedi, Shomir Wilson, Bin Liu, Norman Sadeh, and Alessandro Acquisti. Tweets are forever: a large-scale quantitative analysis of deleted tweets. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 897–908. ACM, 2013.

Alejandro U. Alvarez. Bad words list, March 2013. http://urbanoalvarez.es/blog/2008/04/04/bad-words-list/.

Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. In *Text, Speech and Dialogue*, pages 196–205. Springer, 2007.

Saima Aman and Stan Szpakowicz. Using roget's thesaurus for fine-grained emotion recognition. In *IJCNLP*, pages 312–318, 2008.

Alina Andreevskaia and Sabine Bergler. Clac and clac-nb: Knowledge-based and corpus-based approaches to sentiment tagging. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 117–120. Association for Computational Linguistics, 2007.

Associated Press. Watch your mouth! Americans see profanity getting worse, poll finds, March 2006. http://www.nbcnews.com/id/12063093/from/RS.4/?GT1=7938#.UaKaG2cSpZo.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Conference on Empirical Methods in Natural Language Processing*, pages 355–362. ACL, 2011.

Jin Yeong Bak, Suin Kim, and Alice Oh. Self-disclosure and relationship strength in twitter conversations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 60–64. Association for Computational Linguistics, 2012.

BanBuilder. Banbuilder - PHP curse word function and word list for application developers, moderators, etc., March 2013. http://banbuilder.com/.

Anil Bandhakavi, Nirmalie Wiratunga, Deepak P, and Stewart Massie. Generating a word-emotion lexicon from #emotional tweets. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 12–21, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.

BannedWordList. BannedWordList.com - a resource for web administrators, March 2013. http://www.bannedwordlist.com/.

Shenghua Bao, Shengliang Xu, Li Zhang, Rong Yan, Zhong Su, Dingyi Han, and Yong Yu. Mining social emotions from affective text. *Knowledge and Data Engineering, IEEE Transactions on*, 24(9):1658–1670, 2012.

Plank Barbara. *Domain adaptation for parsing*. PhD thesis, University of Groningen, 2011.

John Rupert Lyon-Bowes Bernard. *The Macquarie Thesaurus: The Book of Words*. Macquarie Library, 1984.

Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. Moa: Massive online analysis. *The Journal of Machine Learning Research*, 11:1601–1604, 2010.

J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128. Association for Computational Linguistics, 2006.

John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*, volume 7, pages 440–447, 2007.

Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *International AAAI Conference on Weblogs and Social Media*, 2011.

M. M. Bradley and P. J. Lang. Affective norms for english words (ANEW): Stimuli, instruction manual, and affective ratings. Technical report, Center for Research in Psychophysiology, University of Florida, Gainesville, Florida, 1999.

Rob B Briner. The neglect and importance of emotion at work. *European Journal of Work and Organizational Psychology*, 8(3):323–346, 1999.

Paul Cameron. Frequency and kinds of words in various social settings, or what the hell's going on? *The Pacific Sociological Review*, 12(2):101–104, 1969.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

François-Régis Chaumartin. Upar7: A knowledge-based system for headline sentiment tagging. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 422–425. Association for Computational Linguistics, 2007.

Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

Keke Chen, Rongqing Lu, CK Wong, Gordon Sun, Larry Heck, and Belle Tseng. Trada: tree based ranking function adaptation. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1143–1152. ACM, 2008.

Lu Chen, Wenbo Wang, Meenakshi Nagarajan, Shaojun Wang, and Amit Sheth. Extracting diverse sentiment expressions with target-dependent polarity from twitter. In *Proceedings of ICWSM*, 2012a.

Lu Chen, Wenbo Wang, and Amit P Sheth. Are twitter users equal in predicting elections? a study of user groups in predicting 2012 us republican presidential primaries. In *Social informatics*, pages 379–392. Springer, 2012b.

Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. Emotion cause detection with linguistic constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 179–187. Association for Computational Linguistics, 2010.

Munmun De Choudhury, Scott Counts, and Michael Gamon. Not all moods are created equal! exploring human emotional states in social media. In *International AAAI Conference on Weblogs and Social Media*, 2012.

Munmun De Choudhury, Scott Counts, and Eric Horvitz. Major life changes and behavioral markers in social media: case of childbirth. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1431–1442. ACM, 2013a.

Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. Predicting depression via social media. In *International AAAI Conference on Weblogs and Social Media*, 2013b.

Glen Coppersmith, Craig Harman, and Mark Dredze. Measuring post traumatic stress disorder in twitter. In *International AAAI Conference on Weblogs and Social Media*, 2014.

Russell Cropanzano and Thomas A Wright. When a "happy" worker is really a "productive" worker: A review and further refinement of the happy-productive worker thesis. *Consulting Psychology Journal: Practice and Research*, 53(3):182, 2001.

W. Dai, Q. Yang, G.R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM, 2007.

Taner Danisman and Adil Alpkocak. Feeler: Emotion classification of text using vector space model. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, volume 1, page 53, 2008.

H. Daumé. Frustratingly easy domain adaptation. In *Annual meeting-association for computational linguistics*, volume 45, page 256, 2007.

Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, pages 101–126, 2006.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computa-*

tional Linguistics: Posters, pages 241–249. Association for Computational Linguistics, 2010.

Peter Sheridan Dodds, Kameron Decker Harris, Isabel M Kloumann, Catherine A Bliss, and Christopher M Danforth. Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. *PloS one*, 6(12):e26752, 2011.

Raymond J Dolan. Emotion, cognition, and behavior. *Science*, 298(5596):1191–1194, 2002.

Matthias Eck, Stephan Vogel, and Alex Waibel. Language model adaptation for statistical machine translation based on information retrieval. In *The International Conference on Language Resources and Evaluation*, 2004.

Paul Ekman, Wallace V Friesen, and Phoebe Ellsworth. *Emotion in the human face: Guidelines for research and an integration of findings*. Pergamon Press, 1972.

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

Jenny Rose Finkel and Christopher D Manning. Hierarchical bayesian domain adaptation. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610. ACL, 2009.

George Foster, Cyril Goutte, and Roland Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 451–459. ACL, 2010.

Barbara L Fredrickson. The role of positive emotions in positive psychology: The broaden-and-build theory of positive emotions. *American psychologist*, 56(3):218, 2001.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proceedings of HLT:short papers*, HLT '11, pages 42–47, Stroudsburg, PA, USA, 2011. ACL.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *International Conference on Machine Learning*, pages 513–520, 2011.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of EAMT*, volume 2005, pages 133–142, 2005.

Olivier Janssens, Maarten Slembrouck, Steven Verstockt, Sofie Van Hoecke, and Rik Van de Walle. Real-time emotion classification of tweets. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1430–1431. ACM, 2013.

Mario Jarmasz and Stan Szpakowicz. The design and implementation of an electronic lexical knowledge base. In *Advances in Artificial Intelligence: 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2001 Ottawa, Canada, June 7-9, 2001 Proceedings*, volume 14, page 325. Springer, 2001.

Timothy Jay. *Cursing in America: A Psycholinguistic Study of Dirty Language in the*

Courts, in the Movies, in the Schoolyards, and on the Streets. John Benjamins Publishing Co, 1992.

Timothy Jay. *Why we curse: A neuro-psycho-social theory of speech*. John Benjamins Publishing, 2000.

Timothy Jay. Do offensive words harm people? *Psychology, public policy, and law*, 15(2): 81, 2009a.

Timothy Jay. The utility and ubiquity of taboo words. *Perspectives on Psychological Science*, 4(2):153–161, 2009b.

Timothy Jay and Kristin Janschewitz. The pragmatics of swearing. *Journal of Politeness Research. Language, Behaviour, Culture*, 4(2):267–288, 2008.

Timothy B Jay. Sex roles and dirty word usage: a review of the literature and a reply to haas. *Psychological Bulletin*, 88(3):614–621, 1980.

J. Jiang and C.X. Zhai. Instance weighting for domain adaptation in nlp. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 264, 2007.

Sepandar D Kamvar and Jonathan Harris. We feel fine and searching the emotional web. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 117–126. ACM, 2011.

Phil Katz, Matthew Singleton, and Richard Wicentowski. Swat-mp: the semeval-2007 systems for task 5 and task 14. In *Proceedings of the 4th international workshop on semantic evaluations*, pages 308–313. Association for Computational Linguistics, 2007.

Renato Kempter, Valentina Sintsova, Claudiu Musat, and Pearl Pu. Emotionwatch: Visualizing fine-grained emotions in event-related tweets. 2014.

Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! *International AAAI Conference on Weblogs and Social Media*, 11:538–541, 2011.

Adam DI Kramer. An unobtrusive behavioral model of gross national happiness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 287–290. ACM, 2010.

Adam DI Kramer, Jamie E Guillory, and Jeffrey T Hancock. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24):8788–8790, 2014.

Jessica Leber. 140 characters of f*ck, sh!t, and @ss: How we swear on twitter. http://www.fastcoexist.com/3026596/140-characters-of-fck-sht-and-ss-how-we-swear-on-twitter, February 2014.

Gilly Leshed and Joseph'Jofish' Kaye. Understanding how bloggers feel: recognizing affect in blog posts. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1019–1024. ACM, 2006.

Kevin Hsin-Yih Lin, Changhua Yang, and Hsin-Hsi Chen. What emotions do news articles trigger in their readers? In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 733–734. ACM, 2007.

Hugo Liu, Henry Lieberman, and Ted Selker. A model of textual affect sensing using real-

world knowledge. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 125–132. ACM, 2003.

Yajuan Lü, Jin Huang, and Qun Liu. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 343–350. Citeseer, 2007.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

Justin Martineau, Lu Chen, Doreen Cheng, and Amit Sheth. Active learning with efficient feature weighting methods for improving data quality and classification accuracy. In *Proceedings of ACL*, pages 1104–1112, Baltimore, Maryland, June 2014. ACL.

Tony McEnery. *Swearing in English: Bad language, purity and power from 1586 to the present*, volume 1. Psychology Press, 2006.

Matthias R Mehl and James W Pennebaker. The sounds of social life: a psychometric analysis of students' daily social environments and natural conversations. *Journal of personality and social psychology*, 84(4):857, 2003.

Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. Mllib: Machine learning in apache spark. *arXiv preprint arXiv:1505.06807*, 2015.

Merriam-Webster Online Dictionary. Emotion. http://www.merriam-webster.com/dictionary/emotion, April 2014.

R. Mihalcea and H. Liu. A corpus-based approach to finding happiness. In *Proceedings of the AAAI Spring Symposium on Computational Approaches to Weblogs*, page 19, 2006.

Rada Mihalcea and Carlo Strapparava. Lyrics, music, and emotions. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 590–599, 2012.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Gilad Mishne. Experiments with mood classification in blog posts. In *Style2005 - the 1st Workshop on Stylistic Analysis Of Text For Information Access, at SIGIR 2005*. SIGIR, ACM, 2005.

Saif 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*, pages 246–255. Association for Computational Linguistics, 2012.

Saif M Mohammad and Peter 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*, pages 26–34. Association for Computational Linguistics, 2010.

Saif M Mohammad, Xiaodan Zhu, and Joel Martin. Semantic role labeling of emotions in tweets. In *Proc. of WASSA*, pages 32–41, 2014.

Meredith Ringel Morris, Scott Counts, Asta Roseway, Aaron Hoff, and Julia Schwarz. Tweeting is believing?: understanding microblog credibility perceptions. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 441–450. ACM, 2012.

Ririn Febrima Nasution and Rusdi Noor Rosa. Swearwords found in chat room yahoo messenger. *E-Journal English Language and Literature*, 1(1), 2012.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. Recognition of affect, judgment, and appreciation in text. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 806–814. Association for Computational Linguistics, 2010.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. Affect analysis model: novel rule-based approach to affect sensing from text. *Natural Language Engineering*, 17(1):95–135, 2011.

Xiaochuan Ni, Gui-Rong Xue, Xiao Ling, Yong Yu, and Qiang Yang. Exploring in the weblog space by detecting informative and affective articles. In *Proceedings of the 16th international conference on World Wide Web*, pages 281–290. ACM, 2007.

NoSwearing. List of Bad Words, March 2013. http://www.noswearing.com/dictionary/.

Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, Qiang Yang, et al. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

S.J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

S.J. Pan, X. Ni, J.T. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM, 2010.

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

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

Maja Pantic, Alex Pentland, Anton Nijholt, and Thomas S Huang. Human computing and machine understanding of human behavior: a survey. In *Artifical Intelligence for Human Computing*, pages 47–71. Springer, 2007.

Parents Television Council. Habitat for Profanity, July 2010. http://www.parentstv.org/PTC/publications/reports/2010ProfanityStudy/study.pdf.

Sungkyu Park, Inyeop Kim, Sang Won Lee, Jaehyun Yoo, Bumseok Jeong, and Meeyoung Cha. Manifestation of depression and loneliness on social networks: A case study of young adults on facebook. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 557–570. ACM, 2015.

James W. Pennebaker, Roger J Booth, and Martha E. Francis. Liwc: Linguistic inquiry and word count. http://www.liwc.net/, April 2014.

Jessica Perrie, Aminul Islam, Evangelos Milios, and Vlado Keselj. Using google n-grams to expand word-emotion association lexicon. In *Computational Linguistics and Intelligent Text Processing*, pages 137–148. Springer, 2013.

John P Pestian, Pawel Matykiewicz, Michelle Linn-Gust, Brett South, Ozlem Uzuner, Jan Wiebe, K Bretonnel Cohen, John Hurdle, and Christopher Brew. Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*, 5(Suppl 1):3, 2012.

Pew Internet. How American teens navigate the new world of "digital citizenship", November 2011. http://pewinternet.org/Reports/2011/Teens-and-social-media/Summary/Findings.aspx.

Pew Internet. Pew Internet: Social Networking (full detail), February 2013. http://pewinternet.org/Commentary/2012/March/Pew-Internet-Social-Networking-full-detail.aspx.

Pew Research Center. Social networking fact sheet. http://www.pewinternet.org/fact-sheets/social-networking-fact-sheet/, January 2014.

Maura Pilotti, Jennifer Almand, Salif Mahamane, and Melanie Martinez. Taboo words in expressive language: Do sex and primary language matter? *American International Journal of Contemporary Research*, 2(2):17–26, 2012.

Profanity - Wikipedia. Profanity - Wikipedia, the free encyclopedia, March 2013. http://en.wikipedia.org/wiki/Profanity.

Matthew Purver and Stuart Battersby. Experimenting with distant supervision for emotion classification. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491. ACL, 2012.

Daniele Quercia, Jonathan Ellis, Licia Capra, and Jon Crowcroft. Tracking gross community happiness from tweets. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 965–968. ACM, 2012.

Kirk Roberts, Michael A Roach, Joseph Johnson, Josh Guthrie, and Sanda M Harabagiu. Empatweet: Annotating and detecting emotions on twitter. In *The International Conference on Language Resources and Evaluation*, pages 3806–3813, 2012.

Magnus Sahlgren, Jussi Karlgren, and Gunnar Eriksson. Sics: Valence annotation based

on seeds in word space. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 296–299. Association for Computational Linguistics, 2007.

Hansen Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Richard E Lucas, Megha Agrawal, Gregory J Park, Shrinidhi K Lakshmikanth, Sneha Jha, Martin EP Seligman, et al. Characterizing geographic variation in well-being using tweets. In *International AAAI Conference on Weblogs and Social Media*, 2013.

Phillip Shaver, Judith Schwartz, Donald Kirson, and Cary O'connor. Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology*, 52(6):1061, 1987.

Amit Sheth, Ashutosh Jadhav, Pavan Kapanipathi, Lu Chen, Hemant Purohit, Gary Alan Smith, and Wenbo Wang. Twitris: A system for collective social intelligence. In *Encyclopedia of Social Network Analysis and Mining*, pages 2240–2253. Springer, 2014.

Sunghwan Sohn, Manabu Torii, Dingcheng Li, Kavishwar Wagholikar, Stephen Wu, and Hongfang Liu. A hybrid approach to sentiment sentence classification in suicide notes. *Biomedical informatics insights*, 5(Suppl 1):43, 2012.

Sara Sood, Judd Antin, and Elizabeth Churchill. Profanity use in online communities. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 1481–1490. ACM, 2012.

Barry M Staw, Robert I Sutton, and Lisa H Pelled. Employee positive emotion and favorable outcomes at the workplace. *Organization Science*, 5(1):51–71, 1994.

Katy Steinmetz. #cursing study: 10 lessons about how we use swear words on twitter. http://time.com/8760/cursing-study-10-lessons-about-how-we-use-swear-words-on-twitter/, February 2014.

Richard Stephens, John Atkins, and Andrew Kingston. Swearing as a response to pain. *Neuroreport*, 20(12):1056–1060, 2009.

Philip J Stone, Dexter C Dunphy, and Marshall S Smith. *The General Inquirer: A Computer Approach to Content Analysis.* MIT press, 1966.

Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics, 2007.

Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1556–1560. ACM, 2008.

Carlo Strapparava and Alessandro Valitutti. Wordnet affect: an affective extension of wordnet. In *The International Conference on Language Resources and Evaluation*, volume 4, pages 1083–1086, 2004.

Kaveri Subrahmanyam, David Smahel, and Patricia Greenfield. Connecting developmental constructions to the internet: identity presentation and sexual exploration in online teen chat rooms. *Developmental psychology*, 42(3):395, 2006.

Jared Suttles and Nancy Ide. Distant supervision for emotion classification with discrete binary values. In *Computational Linguistics and Intelligent Text Processing*, pages 121–136. Springer, 2013.

Swiss Center for Affective Sciences. Research material affective sciences. http://www.affective-sciences.org/researchmaterial, April 2014.

Mike Thelwall. Fk yea i swear: Cursing and gender in a corpus of myspace pages. *Corpora*, 3(1):83–107, 2008.

Ryoko Tokuhisa, Kentaro Inui, and Yuji Matsumoto. Emotion classification using massive examples extracted from the web. In *Proceedings of the 22nd International Conference*

on Computational Linguistics-Volume 1, pages 881–888. Association for Computational Linguistics, 2008.

Peter 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*, pages 417–424. Association for Computational Linguistics, 2002.

Svitlana Volkova, William B Dolan, and Theresa Wilson. Clex: a lexicon for exploring color, concept and emotion associations in language. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 306–314. Association for Computational Linguistics, 2012.

Steven Walden and Qaalfa Dibeehi. The predictive power of emotions. http://www.beyondphilosophy.com/blog/the-predictive-power-of-emotions, October 2012.

Wenbo Wang, Lu Chen, Ming Tan, Shaojun Wang, and Amit P Sheth. Discovering fine-grained sentiment in suicide notes. *Biomedical informatics insights*, 5(Suppl 1):137, 2012a.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. Harnessing twitter "big data" for automatic emotion identification. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 587–592. IEEE, 2012b.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. Cursing in english on twitter. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 415–425. ACM, 2014.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and

dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207, 2013.

David Watson, Lee A Clark, and Auke Tellegen. Development and validation of brief measures of positive and negative affect: the panas scales. *Journal of personality and social psychology*, 54(6):1063, 1988.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1980–1984. ACM, 2012.

G. Xu, X. Meng, and H. Wang. Build chinese emotion lexicons using a graph-based algorithm and multiple resources. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1209–1217. Association for Computational Linguistics, 2010.

Jun-Ming Xu, Xiaojin Zhu, and Amy Bellmore. Fast learning for sentiment analysis on bullying. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, page 10. ACM, 2012a.

R. Xu, J. Xu, and X. Wang. Instance level transfer learning for cross lingual opinion analysis. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 182–188. Association for Computational Linguistics, 2011.

Yan Xu, Yue Wang, Jiahua Liu, Zhuowen Tu, Jian-Tao Sun, Junichi Tsujii, and Eric Chang. Suicide note sentiment classification: a supervised approach augmented by web data. *Biomedical informatics insights*, 5(Suppl 1):31, 2012b.

Changhua Yang, Kevin H Lin, and Hsin-Hsi Chen. Emotion classification using web blog corpora. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 275–278. IEEE, 2007a.

Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. Building emotion lexicon from weblog corpora. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 133–136. Association for Computational Linguistics, 2007b.

Dan Yang and Won-Sook Lee. Music emotion identification from lyrics. In *11th IEEE International Symposium on Multimedia, 2009. ISM'09.*, pages 624–629, 2009.

Hui Yang, Alistair Willis, Anne De Roeck, and Bashar Nuseibeh. A hybrid model for automatic emotion recognition in suicide notes. *Biomedical informatics insights*, 5(Suppl 1):17, 2012.

Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *International Conference on Machine Learning*, volume 97, pages 412–420, 1997.

Xu Zhe and AC Boucouvalas. Text-to-emotion engine for real time internet communication. In *Proceedings of International Symposium on Communication Systems, Networks and DSPs*, pages 164–168, 2002.