Emotion Mining from Text

by

Ameneh Gholipour Shahraki

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

Master of Science

Department of Computing Science
University of Alberta

Abstract

Emotion mining is the science of detecting, analyzing, and evaluating humans' feelings towards different events, issues, services, or any other interest. One of its specific directions is text emotion mining, that refers to analyzing people's emotions based on observations of their writings. Text emotion mining is an interdisciplinary topic of interest and has many applications including helping customer care services, recommending music or movies to computer users, helping in selecting e-learning materials, filtering results of searches by emotion, and diagnosing depression or suicidal tendency.

In this work, we study the problem of text emotion classification. First, we collect and cleanse a corpus of Twitter messages that convey at least one of the emotions: anger, fear, joy, love, sadness, surprise, disgust, guilt, and thankfulness. Then, we propose several lexical and learning based methods to classify the emotion of test tweets and study the effect of different feature sets, dimension reduction techniques, different learning algorithms and configurations, and also try to address the problem of sparsity of the input data. Our experimental results show that a set of Naïve Bayes classifiers, each corresponding to one emotion, using unigrams as features, is the best-performing method for the task. Moreover, we address the problem of multi-label emotion classification of texts, that is concerned with tweets that represent more than one emotion. In this case, again the Naïve Bayes method outperforms the others.

In order to compare the efficiency of our algorithms, we test them also on a couple of other datasets, one of which is collected from Twitter, and the other contains a set of formally written texts. Our Naïve Bayes approach achieves higher accuracy, compared with state-of-the-art methods working on these corpora.

Acknowledgements

I would like to thank my supervisor, Professor Osmar R. Zaiane for his invaluable guidance, encouragement and patience throughout this thesis. I cannot imagine this project done without his incredible support.

I would also like to thank my parents for being supportive and making my life easy even from long distances. Lastly, I thank my beloved husband, Ali, whom I can always count on.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Emotion mining, one of the fields in affective computing, refers to all areas of detecting, analyzing, and evaluating humans' feelings towards different events, issues, services, or any other interest. More precisely, this field aims to mine emotions based on observations of people's actions that can be captured using their writings, facial expressions, speech, music, movements, etc. Analysis of emotions from each of these media is a specific field of study. Here we focus only on text emotion mining. For further information regarding other types of sentiment analysis, one can refer to [1, 2, 3, 4, 5]

Textual emotion mining is a general concept that includes more fine-grained tasks listed below:

- **Emotion Detection:** The task of detecting if a text conveys any type of emotion or not. It can be seen as a binary classification problem, where the possible labels are "yes" and "no".
- **Emotion Polarity Classification:** The task of determining the polarity of the existing emotion in a text, assuming that it has some. This is also a binary classification problem, but here the possible labels are "positive" and "negative".
- **Emotion Classification:** The task of fine-grained classification of existing emotion(s) conveyed by a text into one (or more) of a set of predefined emotions.

As can be inferred from the definitions, we discriminate the words "detection" and "classification", while in existing literature they are mostly used interchangeably referring to the same meaning. In fact, the answer to a detection problem is either yes or no, meaning that there exists an expression of emotion in the text or not. However, the answer to a classification problem is the exact type of emotion (*joy, sadness,* etc.) expressed in the target text.

Emotion mining from text is still in its infancy and yet has a long way to proceed. It is an interesting topic in many disciplines such as neuroscience, cognitive sciences, and psychology. Only recently, has it attracted attention in computer science. Developing systems that can detect emotions from text has many potential applications.

1. In customer care services, emotion mining can help marketers gain information about how satisfied their customers are and what aspects of their service should be improved or revised to strengthen the relationship to their end users [6]. Users' emotions can additionally be used for sale predictions of a particular product.

2. In e-learning applications, the Intelligent Tutoring System (ITS) can decide on teaching materials, based on user's feelings and mental state.

3. In Human Computer Interaction (HCI), the computer can monitor user's emotions to suggest suitable music or movies [7].

4. Having the technology of identifying emotions enables new textual access approaches such as allowing users to filter results of a search by emotion.

5. Output of an emotion mining system can serve as input to other systems. For instance, in [8], authors use the emotions detected in the text for author profiling, specifically identifying the writer's age and gender.

6. Last but not least, psychologists can infer patients' emotions and predict their state of mind accordingly. On a longer period of time, they are able to detect if a patient faces depression or stress [9] or even thinks about committing suicide which is extremely useful since he/she can be referred to counseling services before the suicide happens [10].

On the other hand, with the explosive growth of Web 2.0 technology, different media are available for people to express themselves and their feelings. This has added another aspect to the area. Research exists on detecting emotions from text, facial expressions, images, speeches, paintings, songs, and other sorts of media [11, 12]. Of these, facial expressions and voice recorded speeches contain the most dominant clues and have widely been studied. There are also studies on the combination of different types of information such as features from text and image including the work of Y. Zhang et al. [13]. In this study, we focus only on text, that cannot take advantage of the information conveyed via facial or audio channels. Personal notes, emails, news headlines, blogs, tales, novels, and

chat messages are some types of text that can convey emotions. Particularly, popular social networking websites such as Twitter, Facebook, and MySpace are common places to share one's feelings.

In this research, we address the problem of text-based emotion mining, putting more emphasis on the emotion classification subtask. To this end we target Twitter message. Twitter is an online microblogging media that allows users to post short messages, called *tweet*, limited to 140 characters. Text-based emotion classification is challenging in many aspects such as:

1. Number of representative features of a text are directly related to the length of that text. The shorter the message is, the more sparse its feature vector would be. In the case of tweets, this problem would be extreme.

2. To describe a text by a set of features, the most representative features that one can think of are based on the words used in that text. However, there are many emotional texts that do not have any explicit emotional words. "*Price of IBM stock that I bought yesterday rose.*" is an example in which none of the words are emotional on their own, but they together make a positive feeling when arranged in this sentence. Spotting such implicit emotions is much harder than explicit ones.

3. One of the key characteristics of each language is its changing and evolving nature. This is even more conspicuous in the language used in online conversations and messages. Thus, even if we had a fully functional system for classifying today's texts, it would possibly be less effective over the years.

4. For the special case of tweets, being short, informal, having misspellings, special symbols such as emoticons and emojis, short forms of words, hashtags, and abbreviations are properties that discriminate them from normal texts and add to the complexity of the task. Although emoticons in particular may help classifying the emotion(s) of a text, they still are considered as anomalies in texts because they are not normal words and are not included in dictionaries, so they should be treated with special care in order not to be removed in preprocessing steps.

1.1 Thesis statement

From a coarse-grained point of view we tend to address the problem of emotion classification of texts. From a more fine-grained point of view, we would like to answer the following

research questions:

1. Could we build a system that can automatically detect the emotion(s) conveyed by a piece of text?

2. What are informative features that can be extracted from texts to help detect their emotion(s)?

3. If there is an automatic emotion classifier, could it be generic enough to handle different types of text, in terms of formality of the language?

1.2 Thesis contribution

In this work, we contribute to addressing emotion classification task in several directions:

1. A set of widely acceptable emotion theories are studied, a complete categorization, survey, and analyses of previous research in emotion mining is suggested, and useful resources in this field such as lexicons and datasets are introduced.

2. A corpus of 27,000 emotional tweets containing a balanced number of samples from 9 basic emotions is assembled.

3. A lexicon for emotion mining research is developed based on our Twitter corpus. It contains about 24,000 words, each associated with a vector of weights corresponding to 9 basic emotions.

4. Several lexical and learning-based methods are proposed for classifying emotions on Twitter corpus as well as some other existing datasets, resulting in outperforming some of the state-of-the-art works done on these datasets.

1.3 Thesis organization

This manuscript is organized as follows: in Chapter 2, a thorough survey on emotion mining methodologies is given. In addition, useful resources such as lexicons and datasets that can be used in this field are introduced. In Chapter 3, our Twitter corpus, called *Cleaned Balanced Emotional Tweets (CBET)* and its collection and cleaning process is explained in details. Chapter 4 is dedicated to discussing the proposed emotion classification methods, explainig the designed experiments on *CBET*, and illustrating and analyzing the results. In Chapter 5, we test our methods on two other datasets. One of them, called *TEC*, is gathered

from Twitter and hence is more similar to ours. The other one, called *ISEAR*, contains some formally written texts. The purpose is to test the robustness of our methods against different types of texts. Finally, in Chapter 6, future work in this field is explored and the discussion is summarized and concluded.

Chapter 2

Background and Related Work

In any emotion related research, the first question to be answered is "what defines emotion". This section is started by introducing some theories that define emotion and suggest some sets of basic emotions. Then some useful resources are introduced and some text-based emotion identification works are reviewed.

2.1 Review of Emotion Theories

In this section we introduce some theories that define emotion and suggest some sets of basic emotions. While most research on emotions in computer science use the terms *emotion*, *feeling*, *mood*, and *affect* interchangeably, these terms do not share the same exact meaning. According to E. Fox [14], in affective neuroscience, the terms are defined as follows::

- **Emotion:** Discrete and consistent responses to internal or external events which have a particular significance for the organism. Emotion has short term duration.
- **Feeling:** a subjective representation of emotions, private to the individual experiencing them. Similar to emotion, it has short term duration.
- **Mood:** a diffuse affective state that compared to emotion is usually less intense but with longer duration.
- **Affect:** an encompassing term, used to describe the topics of emotion, feelings, and moods together. It often has long term duration.

Even with having clear definitions of these terms, there are still some controversial issues regarding whether some particular human states are classified as an emotion or not. For instance, *thankfulness* and *gratitude* is considered as an emotion by some theorists

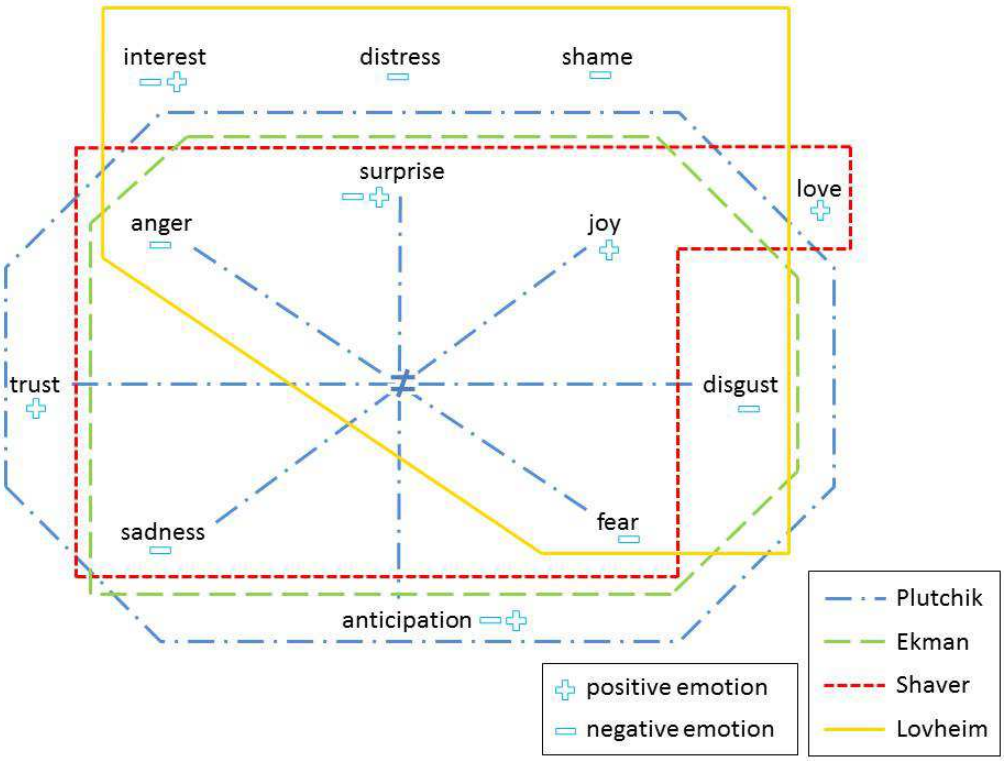

Figure 2.1: The illustration of 4 frequently used emotion models

while others believe actions such as greeting, thanking, and congratulating are just some *communicative functions*.

Scientific studies on classification of human emotions date back to 1960s. There are two prevalent theories in this field. The first one, *discrete emotion theory*, states that different emotions arise from separate neural systems. Conversely, *dimensional model* states that a common and interconnected neurophysiological system is responsible for all affective states. This model defines emotions according to one or more dimensions where usually one of them relates to intensity of emotions.

Basic emotions refer to those that do not have any other emotion as constituent parts. In addition, they can be recognized by humans all over the world regardless of their race, culture, and language. Many theorists on both sides have proposed sets of emotions that tend to be basic ones. Table 2.1 shows some of the more frequently used models of basic emotions. Ekman, one of the earliest emotion theorists, suggested that those certain emotions that are universally recognized form the set of basic emotions. He later expanded his set of emotions by adding 12 new positive and negative emotions [15]. The dimensional model of R. Plutchik and H. Kellerman [16] arranges emotions in four bipolar axes: *joy* vs. *sadness*;

Table 2.1: Different models of basic emotions proposed by theorists

Theorist	Year	Basic Emotions	Type
Ekman	1972	anger, disgust, fear, joy, sadness, surprise	discrete
Plutchik	1986	anger, anticipation, disgust, fear, joy, sadness, surprise, trust	dimensional
Shaver	1987	anger, fear, joy, love, sadness, surprise	discrete
Lovheim	2011	anger, disgust, distress, fear, joy, interest, shame, surprise	dimensional

anger vs. *fear*; *trust* vs. *disgust*; and *surprise* vs. *anticipation*. The fact that some of these emotions are actually opposite of each other is trivial in cases like *joy* vs. *sadness* but it is less intuitive in other cases, such as *anger* vs. *fear*. P. Shaver et al. [17] model emotions in a tree structure such that basic emotions are the main branches and each branch has its own categorization. H. Lövheim also suggests a dimensional model; however, his model is different from Plutchik's [18]. He believes that three hormones of serotonin, dopamine, and noradrenaline form three dimensions of a cube, where each basic emotion is placed on one of the corners.

Figure 2.1 illustrates the 4 explained models together so that one can compare them. The Plutchik's bipolar division of emotions is shown using the sign $\neq$. The positiveness and/or negativeness of emotions are also shown using the signs $+$ and $-$ respectively. Emotions such as *interest, surprise,* and *anticipation* can be both positive and negative, depending on the situation they are felt. C. O. Alm and R. Sproat [19] even divide *surprise* to two separate emotions of *positively surprise* and *negatively surprise*. Table 2.2 shows another illustrations of commonality of these emotion models. According to both Figure 2.1 and Table 2.2, *anger, fear, joy,* and *surprise* are common in all models, but there is no agreement on the rest. One interesting point in all models is that the number of negative emotions outweighs the number of positive ones. While psychologists do not agree on what model describes more accurately the set of basic emotions, the model suggested by P. Ekman et al. [20], with 6 emotions, is the most widely used in computer science research.

2.2 Review of Emotion Mining Methodologies

In 1992, J. B. Walther introduced the *Social Information Processing (SIP)* theory which states that in order to convey relational information in computer-mediated communications, people use verbal clues instead of nonverbal clues that would have been used in face-to-face environments [21]. Theorists of this theory later validated their hypothesis in [22] by

Table 2.2: Commonality of emotion models

Emotion	Ekman	Plutchik	Shaver and Parrott	Lovheim
Anger	✓	✓	✓	✓
Anticipation		✓		
Disgust	✓	✓		✓
Distress				✓
Fear	✓	✓	✓	✓
Interest				✓
Joy	✓	✓	✓	✓
Love			✓	
Sadness	✓	✓	✓	
Shame				✓
Surprise	✓	✓	✓	✓
Trust		✓		

conducting an experimental study and showed that affinity is expressed equally effectively in both face-to-face and textual styles. In addition, verbal clues carried a larger portion of relational information in communications via a computer medium. This simple theory can be a proof for validity of textual emotion mining research topic.

Automatic detection of emotions can be categorized from different aspects. From the perspective of granularity, it can be done on the document level or sentence level. At the document level, the whole document, whether short or long, is the atomic unit of input to the problem and what matters is the emotion of the whole document, whereas at the sentence level, each sentence is expected to have an emotion independently. As noted in [23], a challenge in labeling at the sentence level is the influence of surrounding context on the sentence. For example, depending on what context it is used, the sentence "*I can't really describe my situation better than this.*" can convey *sadness, joy,* or even other emotions.

In a text environment, emotion analysis can be from the writer's perspective or the reader's perspective. The former refers to emotions that the author had when he/she was writing the message, while the latter refers to a user's affective response to being exposed to feelings evoked by an emotional text. Readers can further be divided into two groups: an individual reader or a group or society of readers, sometimes referred to as *social emotion detection*. Both the writer and reader can feel the same emotion in some cases; however, it is not a general rule. The reader's point of view has attracted less attention in the literature, nevertheless, it has many applications including helping authors to predict how their work will influence the audience or helping readers to retrieve documents that have content relevant to their desired emotion [24]. Examples of social emotion detection can be found in

[25] and [26].

In some configurations, each sample (a document or a sentence) is assumed to have one single emotion while some times the text can be multi-emotional which means it can contain several emotions at the same time. An example of this situation is the short document "I was happy that it was my birthday yesterday, I was anticipating my family to throw me a party, however that nobody remembered it made me sad" which shows *joy, anticipation,* and *sadness* simultaneously.

There are two general approaches to the problem of textual emotion mining. The first one, called *lexical-based* method, exploits a lexicon of words to decide about emotions of each or a group of words in a text and then aggregates those information to predict the total emotion of the whole document. This category of approaches is sometimes referred to as *keyword-based* methods too. The second one, called *learning-based* method, applies some machine learning algorithms on a set of training data, in order to be able to predict the emotion of unseen test data. A lexicon may still be used to help doing feature selection or extraction.

Learning-based methods can further be divided into two categories. In the *supervised* learning, the algorithm is provided with labeled training data, where label in our problem refers to the emotion of each sample. In *unsupervised* learning, the data does not have to be labeled. These algorithms try to learn how to classify the data without relying on labels. Supervised methods often show better results, nevertheless they require manual labeling of data which is an expensive process in terms of both time and money. There is also a hybrid method, called *semi-supervised* learning, which combines the idea of supervised and unsupervised methods. This means that a little portion of training data is labeled while the majority still stays unlabeled.

There exist some comprehensive surveys on sentiment analysis by B. Pang and L. Lee [27] and B. Liu [28] where the latter is expanded very recently in [29]. While methods and techniques discussed in these papers can be applied to the field of emotion mining as well, none of them have specific coverage of this task. There are also some surveys focusing on emotion mining such as the works by E. Kao et al. [30] and M. C. Jain and V. Y. Kulkarni [31], but they are rather incomplete. These facts motivate us to cover the state-of-the-art methods and resources developed for emotion mining as to be a complementary for existing surveys. Since most of the body of research on emotion mining is dedicated to emotion classification, we put more emphasis on this division too; however, it should be noted that some other directions of this field are also being investigated. For example,

emotion cause detection is the study of mining factors for eliciting some kinds of emotions and is addressed in [32].

One of the works to characterize how users express emotions in text-based systems is [33]. Their study on 40 undergraduate male and female students showed that both genders agree more with their conversation partner when they want to convey a positive attitude. They also use 5 times less negative affect terms and use more punctuation marks. On the other hand, those partners who receive the emotional texts, judge mostly based on negations and exclamation points. These findings are in line with what *SIP theory* suggests. This study contributes to automatic extraction of emotions from text in that it provides an insight into the strategies that people employ to convey their emotions.

E. Kao et al. introduce one of the earliest surveys on textual emotion mining [30]. They classify works into lexical-based (or keyword-based), learning-based, and hybrid methods where hybrids combine detecting keywords, learning patterns, and using other supplementary information. They then suggest a system in which keywords are extracted using a semantic analysis and an ontology is designed with emotion theory of appraisal. These two are combined in a case-based reasoning architecture.

M. C. Jain and V. Y. Kulkarni [31] give a short survey on emotion mining research but their review lacks a rational categorization of works. They introduce some Information Retrieval (IR) models that can be used in text research and suggest a system, called *TexEmo*, which essentially uses a bag of words with TF-IDF weighting as features and trains an SVM classifier on them. They do not report any results out of this system.

S. M. Kim et al. [34] follow lexical-based approaches to evaluate the merit of *discrete emotion theory* and *dimensional model*, discussed in Section 2.1. To build a classifier based on the theory of discrete emotions, they use Wordnet Affect lexicon as well as three dimension reduction techniques, namely Latent Semantic Analysis (LSA), Probabilistic LSA (PLSA), and Non-negative Matrix Factorization (NMF). To build a dimensional classifier, they use a normative database of English affective words, called *Affective Norm for English Words (ANEW)* in which each word is rated on three dimensions of valence, arousal, and dominance. According to their results on SemEval 2007, ISEAR, and fairy tales datasets, that all will be introduced in Section 2.4, performance of methods varies on each emotion and there is no method that performs better than others on all emotions that are under discussion.

C. O. Alm et al. [35] try to identify emotional passages and determine their valence (positive vs. negative). From their dataset of children's fairy tales, they extract 30 features

out of text including direct speech (if the sentence is a whole quote), punctuation marks, complete upper-case words, sentence length, range of story progress, and POS. Next, a linear classifier, called Sparse Network of Winnows (SNoW), is applied on the data. Although their classification results are unsuccessful, their dataset is reputed and widely used in the field of emotion mining.

A. Neviarouskaya et al. [23] construct a rule-based system for emotion recognition, named *Affect Analysis Model (AAM)*. They create an affect database that contains emoticons, acronyms, abbreviations, affect words, interjections, and modifiers. Each entry is manually labeled with an emotion and an intensity showing the degree of its affective state. This database is then used in a five-stage system: symbolic cue analysis, syntactical structure analysis, word level analysis, phrase level analysis, and finally sentence level analysis. Each stage consists of a set of rules that help identify the emotion relied in the text. An example rule is as: "In a compound sentence that independent clauses are connected with comma, 'and', or 'so', the output emotion is equal to the emotion of the clause with maximum intensity." In a later work, they added the ability to process sentences of different complexity [36]. To do so, they decompose a sentence in pieces that correspond to lexical units and then apply some extra rules to infer the total emotion of the text based on the emotions of its parts. AAM is claimed to handle informal messages and is tested on a dataset of diary-like blog posts; however, it still has a long way to prove this for other data. In addition, it cannot distinguish between different meanings of words regarding the context and does not take into account the expression-modifiers such as "*to death*" in the example "*I love my ipad to death*".

F. R. Chaumartin [37] proposes another rule-based system, called *UPAR7*, specifically for SemEval 2007 dataset. They use the Stanford syntactic parser to build the dependency graph for each news headline. Next, they enrich Wordnet Affect and SentiWordnet lexicons in order to use them for rating each word separately and then try to rate the main subject of the whole headline sentence, considering contrasts, accentuations, negations, modals, etc. UPAR7 ranked as one of the top systems that competed in SemEval 2007, shared task of affective computing.

C. Strapparava and R. Mihalcea in [38] predict emotions of news headlines in an unsupervised manner from SemEval 2007 dataset. In one experiment, they use Latent Semantic Analysis (LSA) technique as a semantic similarity mechanism. Each document can be represented in an LSA space by summing up the normalized LSA vectors of all the terms contained in it. In another experiment, they train a Naïve Bayes classifier on a collection

of LiveJournal blogs as training set and use this classifier to label their news data. Their results are acceptable compared to three other algorithms that participated in SemEval 2007 workshop.

T. Danisman and A. Alpkocak [39] use a Vector Space Model (VSM) classifier in which each document is represented as a vector and each axis corresponds to a unigram word. The value of a word in a vector (a document) is calculated using TF-IDF. VSM is relying on two simplifying assumptions that documents with the same emotion form a contiguous region and a region of one emotion does not overlap with the others'. Having this model, on classification time, the test document is converted to a vector and the cosine angle between this vector and all other vectors in the model determines the similarity. They show that VSM outperforms SVM and Naïve Bayes classifiers on SemEval 2007 dataset.

N. Gupta et al. [6] use an algorithm from the boosting family, namely *Boostexter* that is initially proposed in [40]. Each base classifier in Boostexter assigns a confidence value in addition to its prediction for each instance. For a test instance, the final classifier outputs the sum of all confidences of all classifiers per class. They also show the effectiveness of using a set of so called *salient features* that are essentially some linguistic clues from a dataset of customers' emails to customer service department of some companies. These salient features include negative emotions, negative opinions and other expressions specific to the domain of customer care such as threat to take their business elsewhere, etc. According to their results, adding salient features to traditional n-gram features improves the performance significantly.

Following a psychological-based approach, D. T. Ho and T. H. Cao [41] use a high-order Hidden Markov Model (HMM) to address the emotion classification problem on the ISEAR dataset. They believe that emotion is the result of a sequence of mental states, so their idea is to transform the input text into a sequence of events that cause mental states and then automatically generate an HMM to model the process that this sequence of events causes the emotion. Over 4 emotions of *anger, fear, joy,* and *sadness*, where *anger* includes both *anger* and *disgust*, they get the F1 value equal to 35.5% in their best setting. In Chapter 5, we will use this work to compare with our results.

As stated in Section 2.1, *mood* is a less intense state compared to emotion but has long term effects. Mood classification, thus, is very similar to emotion classification and is partially addressed in the literature such as G. Mishne's work [42]. The problem in [42] is to classify blog posts into one out of 40 moods including *excited, sleepy, confused, crazy,* etc. The author focuses mostly on feature selection by investigating the effectiveness of

length-related and semantic-oriented features, frequencies of POS tags, Pointwise Mutual Information (PMI) for each word and mood, and emphasized words. They believe that due to the subjective nature of mood categories and annotations in the corpus, good results are not achieved.

Table 2.3: Summary of current emotion mining methods

Name	**Dataset**	**Emotions**	**Multi-label**	**Method**
C. Alm et al., 2005	fairy tales	categorizing anger, disgust, fear, joy, sadness, positive surprise, and negative surprise into positive, negative, and neutral	No	Sparse Network of Winnows
G. Mishne, 2005	LiveJournal	40 moods	No	Support Vector Machine
A. Neviarouskaya et al., 2007	160 sentences from online blog posts	anger, disgust, fear, guilt, interest, joy, sadness, shame, surprise	No	Rule-based
F. R. Chaumartin, 2007	SemEval 2007	anger, disgust, fear, joy, sadness, surprise	No	Rule-based
C. Strapparava and R. Mihalcea, 2008	SemEval 2007	anger, disgust, fear, joy, sadness, surprise	No	(1) unsupervised: knowledge-based, (2) supervised: Naive Bayes
T. Danisman and A. Alpkocak, 2008	SemEval 2007	anger, disgust, fear, joy, sadness	No	Vector Space Model
A. Neviarouskaya et al., 2009	diary-like blog posts	anger, disgust, fear, guilt, interest, joy, sadness, shame, surprise	No	Rule-based
P. K. Bhowmick, 2009	Indian news headlines	disgust, fear, happiness, sadness	Yes	ensemble of Label Powerset classifiers
S. Kim et al., 2010	SemEval 2007, ISEAR, and Fairy tales	anger, fear, joy, sadness	No	unsupervised: lexical-based

Table 2.3: Summary of current emotion mining methods

Name	Dataset	Emotions	Multi-label	Method
D. T. Ho and T. H. Cao, 2012	ISEAR	anger (including disgust), fear, joy, and sadness	No	Hidden Markov Model
K. Luyckx et al. 2012	600 suicide notes for track 2 of the 2011 medical NLP challenge	instructions, hopelessness, love, information, guilt, blame, thankfulness, anger, sorrow, hopefulness, fear, happiness_peacefulness, pride, abuse, forgiveness	Yes	Support Vector Machine
N. Gupta et al., 2013	set of 1077 customers' emails	factual, emotional	No	Boosting
M. C. Jain and V. Y. Kulkarni, 2014	—	anger, disgust, fear, joy, sadness, surprise	No	Support Vector Machine

2.2.1 Multi-label emotion mining research

In machine learning, multi-label classification algorithms are traditionally categorized into two classes: algorithm adaptation methods and problem transformation methods. The idea of the first approach is to adapt the existing single-label classification algorithm to enable it to classify multi-labeled data. In the second approach, using some transformation techniques, the multi-labeled data are transformed into another problem space in which they have a single label and then a single-label classifier is applied on them [43]. In what follows, some of multi-label emotion classifiers are introduced.

Given that there are k different single labels, P. K. Bhowmick in [43] uses an ensemble-based approach, called *random k-label sets classifier (RAKEL)* which basically consists of an ensemble of *Label Powerset (LP)* classifiers. Each LP learns one single classifier with $k^{'}$ possible labels where $k^{'} \leqslant k$ and is trained using a different small random subset of all emotions. A test instance is classified by combining votes from individual LP classifiers such that it is labeled with an emotion if the average vote of all classifiers is greater than a user specified threshold. This work is an example of algorithm adaptation methods. Additionally, they explore the effectiveness of different feature sets such as polarity of subject, object, and verbs in sentences and semantic frame features using Berkeley FrameNet

lexicon [44]. Results of their experiments on a dataset of Indian news headlines reveal that the combination of polarity and semantic features is the best choice for a multi-label environment.

[10] is another work on multi-label classification of emotional texts. They focus on a dataset of notes written by people who have committed suicide, provided for track 2 of medical NLP shared task, 2011 [45]. The task is to predict label(s) of a note among 15 possible emotions, such as *hopelessness, love, pride, thankfulness,* etc. We think that it is doubtful to consider that some of these labels such as *instructions, information,* etc. are really emotions. First of all, they split all multi-labeled notes to single-labeled fragments manually. Next, an SVM with Radial Basis Function (RBF) is trained on these single-labeled data. Finally, a threshold is set for SVM's probability estimated for each emotion, if the probability exceeds the threshold, then that emotion is assigned to the sentence. Their method has improved the recall compared to a baseline method with the cost of degrading the precision.

Table 2.3 shows a summary of the explained methods in this section, sorted by their chronological order. They are compared for the dataset and set of emotions they use as well as the main characteristics of their approach.

2.2.2 Emotion mining research on Twitter

With more than 300 million active users and 500 million tweets per day [1], Twitter is a popular network for sharing personal feelings and moods with acquaintances and friends. Hence, significant research is devoted to Twitter data with the purpose of analyzing the emotions expressed in tweets. Being short, informal, having misspellings, using hashtags, special symbols such as emoticons and emojis, short forms of words, and abbreviations are properties that discriminate tweets from normal texts and add to the complexity of the task.

J. Bollen et al. [46] analyze emotions of all tweets in a specific time frame. They use a psychometric test, named *Profile of Mood States (POMS)* consisting of 793 adjective terms, each related to a particular emotion. Then the probability that each tweet shows an emotion is calculated based on these features and results are aggregated over all tweets of one day. Finally the overall emotions of tweets are compared with global events of that period and some correlations are found. Although this method does not consider the reader's perspective, it may still be classified as a social emotion detection task, introduced earlier in this section.

[1] https://about.twitter.com/company

Hashtags are space-free phrases following the '#' character such as #mickeymouse and #iamhappy. They can be used as indexes to search for related content or grouping messages. Hashtags are widely used in Twitter as they convey valuable information in a short piece of text. W. Wang et al. [47] build a dataset from Twitter, containing $2,500,000$ tweets and use hashtags as emotion labels [2]. In order to validate this type of labeling, they select 400 tweets randomly and label them manually. Comparing manual labels and hashtag labels show acceptable consistency. Next, they explore the effectiveness of different features such as n-grams, different lexicons, POS, and adjectives in detecting emotions. Their best result is obtained when unigrams, bigrams, lexicons and POS are used. Finally, they show that increasing the size of the training set has direct effect on accuracy. While their dataset is a good source of emotional tweets, it is highly imbalanced and the use of some unclear hashtags as emotion labels, such as #embarrass for *sadness*, makes the soundness of the dataset open to criticism.

M. Hasan et al. [48] also validate the use of hashtags as emotion labels on a set of 134,000 tweets. To this end, they compare hashtag labels with labels assigned by a group of people as well as those assigned by a group of psychologists. They found that crowd labels are not in agreement even with themselves; however, psychologists' labels are more consistent and show more agreement with hashtags. Therefore, they cast doubt on the use of crowd labeling such as in Amazon's Mechanical Turk for tasks related to emotion mining. They also introduce a supervised classifier, named *EmoTex*. It essentially uses the feature set of unigrams, list of negation words, emoticons, and punctuations and runs KNN and SVM on the training data.

K. Roberts et al. [49] create a corpus of 7,000 manually labeled tweets that are retrieved by searching for 14 emotion evoking topics, such as World Cup and Christmas. There are a total of 7 emotions where each tweet can have zero, one or multiple of them. Seven binary SVMs, one for each emotion and each with a different feature set are trained. Features include ngrams, punctuations, hypernyms, and topics. To obtain topics, they assume that each tweet associates with a probabilistic mixture of topics and they are inferred using *Latent Dirichlet Allocation (LDA)*. Their best performance is over the emotion *fear* which led them to infer that *fear* is highly lexicalized with less variation than other emotions.

S. M. Mohammad in [50] introduces his corpus, called *Twitter Emotion Corpus (TEC)* collected from Twitter that will be explained in Section 2.4 and similar to [49] builds binary SVMs, one for each emotion, using unigrams and bigrams as features. He then shows

[2] their dataset is available for download at `http://knoesis.org/projects/emotion`

Table 2.4: Summary of current emotion mining methods on Twitter

Name	Dataset	Emotions	Method	Labeling Process
J. Bollen et al., 2011	crawled about 9,000,000 tweets	tension, depression, anger, vigour, fatigue, Confusion	Profile of Mood States	no labeling
W. Wang et al., 2012	crawled about 2,500,000 tweets	anger, fear, joy, love, sadness, surprise, thankfulness	linear classifier	using hashtags
K. Roberts et al., 2012	crawled 7,000 tweets from 14 emotion evoking topics	anger, disgust, fear, joy, love, sadness, surprise	Support Vector Machine	manual
S. M. Mohammad, 2012	built Twitter Emotion Corpus (TEC) by crawling about 21,000 tweets	anger, disgust, fear, joy, sadness, surprise	Support Vector Machine	using hashtags
M. Hasan, 2014	crawled about 134,000 tweets	2 dimensional model: active, inactive / happy, unhappy	Support Vector Machine and K-Nearest Neighbors	using hashtags
W. Li and H. Xu, 2014	16485 posts from Weibo, a Chinese microblogging website	anger, disgust, fear, joy, sadness, surprise	Support Vector Regression	manual

the effectiveness of this corpus in cross-domain classifications by using this data to predict emotions on another dataset, SemEval 2007. He also builds a lexicon from this corpus that will be introduced in Section 2.3.

Table 2.4 depicts the summary of the explained methods working on Twitter data, sorted in chronological order. They are compared for the dataset and set of emotions they use, as well as the main characteristics of their approach.

2.2.3 Emotion mining for other languages

Most of the works in textual emotion mining have been on English, as is ours, nevertheless it is worth mentioning the very few works done on other languages, since the ideas and techniques may still be used in a language agnostic way.

W. Li and H. Xu [51] try to detect emotions from messages in Weibo, a Chinese mi-

croblog website with functionalities thoroughly similar to Twitter. They believe that the accuracy of detecting emotions in a text can be increased if we look for the events that cause emotions. In this manner, their work is similar to [41]. Therefore, they adopt the notion of *cause events* that are meant to be the reasons for emotions. To spot cause events and use them as features, they exploit a marker list, containing keywords to mark occurrence of cause events, an emotion list, containing keywords expressing emotions, and a linguistic pattern set, describing how emotions and cause events are arranged in a text. All of these resources are adapted to informal environment of Weibo. Then a *Support Vector Regression (SVR)*, an algorithm from the family of SVMs, is trained using these features. According to the results, performance is boosted for some emotions; though it is decreased for others such as *fear* and *sadness*. [26] is another example of an emotion mining study in Chinese, that will be explained in Section 2.3. Also, the aforementioned method of P. K. Bhowmick [43] has addressed the emotion mining task on an Indian dataset in a multi-label environment.

2.3 Emotion related lexicons

Almost all the emotion mining works in both lexical and learning-based methods, rely on using a lexicon. Lexicons are very useful in that they give prior information about the type and strength of emotion carried by each word or phrase. In this section we introduce some of the lexicons useful for the emotion mining task. Their characteristics are summarized in Table 2.5.

2.3.1 Wordnet Affect

Wordnet Affect [3] is an emotional lexical resource, including a list of sets of synonym words, referred to as *synsets*. The set of emotions in this lexicon is hierarchically organized. C. Strapparava and A. Valitutti [52] build this lexicon on top of their previous lexicon, Wordnet. They manually form an initial set of $1,903$ affective words and expand them by adding their corresponding nouns, verbs, adjectives, adverbs, etc. Then a subset of synsets of Wordnet that contain at least one of these affective words are selected and the rest are rejected. This forms the core of the lexicon. The lexical and semantic relations between synsets of this core lexicon and other synsets of Wordnet are then examined to see if they preserve the affective meaning represented by those core synsets. After adding new synsets, Wordnet Affect contains $2,874$ synsets and $4,787$ words. One interesting feature of this lexicon is

[3] `http://wndomains.fbk.eu/wnaffect.html`

the notion of stative/causative for words. A word is causative if it refers to some emotion that is caused by that entity (e.g. amusing). On the other hand, a word is said to be stative if it refers to the emotion owned or felt by that subject (e.g. amused).

2.3.2 WPARD

Using an online form, D. A. Medler et al. [53] collected information from 342 undergraduate students. Participants were asked to rate how negative or positive were the emotions they associate with each word, using a scale from -6 (very negative) to $+6$ (very positive). They built the lexicon Wisconsin Perceptual Attribute Rating Database (WPARD) [4] from this data such that each word has a corresponding polarity and a real number showing the strength of that polarity. Although WPARD does not give information about exact emotions expressed by each word, it is still a good source of sentiment information.

2.3.3 LIWC

Linguistic Inquiry and Word Count (LIWC) [5] is another emotion related lexicon developed by J. Pennebaker et al. [54]. In the first step of generating this lexicon, some initial category scales are generated in a psychological process and then by brain-storming sessions various scales are added to initial lists. In the next step, three independent judges rate the words in two phases, such that after completion of each phase, all category scale lists are updated according to judges' rates. The initial LIWC judging took place in 1992 and since then, it is updated and largely expanded.

2.3.4 NRC

S. M. Mohammad and P. D. Turney [55] develop the NRC word-emotion association lexicon [6]. Using Amazon's Mechanical Turk, they asked Turkers to annotate words, from non-specific domains, according to the emotion they evoke. One important challenge in this process is malicious annotations that can happen in cases that words in different senses evoke different emotions. To solve this problem, the target sense needs to be conveyed to annotators. Hence, they asked additional questions from Turkers, including word choice questions, that help identify instances where the annotator may not be familiar with the target term. In addition to building a lexicon, they also concluded that a regular crowd can produce reliable emotion annotation, given proper guidelines. This is in contrast with

[4] http://www.neuro.mcw.edu/ratings/

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

[6] http://www.saifmohammad.com/WebPages/lexicons.html

Table 2.5: Summary of emotion related lexicons

Name	Author	Year	Size (words)	Set of Emotions
Wordnet Affect	C. Strapparava	2004	4,787	a hierarchy of emotions
WPARD	D.A. Medler	2005	1,400	positive, negative
LIWC	J.W. Pennebaker	2007	5,000	affective or not, positive, negative, anxiety, anger, sadness
NRC	S. Mohammad	2010	14,182	anger, fear, anticipation, trust, surprise, sadness, joy, disgust
NRC hashtag	S. Mohammad	2013	32,400	anger, fear, anticipation, trust, surprise, sadness, joy, disgust

findings of M. Hasan et al. [48] who showed that crowd labeling of emotional tweets have relatively low inter-agreement with each other and with emotional hashtags of tweets.

2.3.5 NRC Hashtag

In another attempt, the main author of NRC lexicon, S. M. Mohammad, developed another useful lexicon, called NRC hashtag emotion lexicon [7] [50]. Using a corpus of $21,000$ tweets, called *TEC* that will be introduced in Section 2.4, the Strength of Association (SoA) for a ngram n and an emotion e is calculated to be:

$$SoA(n, e) = PMI(n, e) - PMI(n, \neg e) \tag{2.1}$$

where PMI is the pointwise mutual information, calculated as:

$$PMI(n, e) = log \frac{freq(n, e)}{freq(n) * freq(e)} \tag{2.2}$$

where $freq(n, e)$ is the number of times that n occurs in a tweet that has the label e. $freq(n)$ and $freq(e)$ are frequencies of n and e respectively in corpus. $PMI(n, \neg e)$ is calculated likewise. Words having SoA greater than zero are kept in the lexicon.

In addition to these publicly available lexicons, there are other lexicons generated for specific tasks that are not accessible, nevertheless reviewing their method of generation can still give some ideas, if one wants to build his/her own special-purpose lexicon. P. Katz et al. [56] create a word-emotion mapping from the SemEval 2007 dataset that will be introduced in Section 2.4. A weight vector is assigned to each lemmatized word w from the corpus, such that each element in this vector is corresponding to one emotion. The value of

[7] http://www.saifmohammad.com/WebPages/lexicons.html

Table 2.6: Summary of emotion related datasets

Name	Author	Year	Size	Type of Data
ISEAR	K.R. Scherer	1997	7,666	crowd written paragraphs
fairy tales	C. Ovesdotter Alm	2005	15,000	sentences from children's stories
SemEval	C. Strapparava	2007	1,250	news headlines
TEC	S. M. Mohammad	2012	21,000	tweets

this element then is calculated to be the average emotion score observed in all samples that w participated in.

2.3.6 Chinese lexicon

J. Lei et al. [26] propose a framework of generating a domain and context dependent emotion lexicon. Firstly, they select a well-formed train set from the corpus of news headlines taken from the Sina website, a popular news site in China. The criterion for selecting a headline is to be among those with the highest rating for at least one emotion. Then, the lexicon is built such that for each word f_j and each emotion e_k:

$$P(e_k|f_j) = \frac{\sum_{i=1}^{D} \sigma_{ij} r_{ik} \epsilon_i}{\sum_{k=1}^{E} \sum_{i=1}^{D} \sigma_{ij} r_{ik} \epsilon_i} \tag{2.3}$$

where σ_{ij} is the relative term frequency of f_j in document d_i; r_{ik} is the co-occurrence number of document d_i and emotion e_k and ϵ_i is the prior probability of document d_i. Results of their experiments show an improvement over existing lexicon generation methods such as [56].

2.4 Emotion related datasets

One of the old challenges in most machine learning works is collecting data, especially labeled ones. Apart from the costs of manual labeling, in specific problem of emotion annotation, results are often subject to misunderstandings, subjective interpretations of annotators, their personality, the perspective that the content is analyzed, and so on [57]. In this section we introduce some useful datasets that have a reliable labeling process and/or are widely used. Table 2.6 shows a summary of these datasets.

2.4.1 ISEAR

One of the oldest emotion labeled datasets, freely available for download, is ISEAR [8], presented in [58]. The data was collected during 1990s, by a large group of psychologists all over the world who were working on International Survey On Emotion Antecedents And Reactions (ISEAR) project. In this survey, $3,000$ students, both psychologists and non-psychologists, in 37 countries on all 5 continents were asked to report situations in which they had experienced 7 major emotions: *joy, fear, anger, sadness, disgust, shame,* and *guilt*. Respondents are asked to write sentences or paragraphs to explain how they had appraised the situation and how they reacted. For non-English speakers, the text was translated to English. Hence the format of data is a sentence or paragraph, labeled with exactly one emotion. This dataset is reliable in terms of labeling, since the authors, themselves, have annotated their text. However, translating from other languages to English might change the senses and emotions. Surprisingly, ISEAR was not used for emotion mining purposes until 2008.

2.4.2 Fairy Tales

A set of fairy tales is another dataset [9] developed by C. O. Alm and R. Sproat [19]. It contains 185 children's stories written by Beatrix Potter, Brothers Grimm, and Hans Christian Andersen, with a total of about $15,000$ sentences that are labeled by one of the emotions: *anger, disgust, fear, happiness, sadness, positively surprised, negatively surprised* or *neutral* if it does not show any emotion. The annotation is done manually by 6 female native English speakers. Note that unlike the ISEAR dataset in which texts are annotated on the document level, in fairy tales dataset, annotation is done on the sentence level.

2.4.3 SemEval 2007

C. Strapparava and R. Mihalcea [59] developed a dataset for the Semantic Evaluation (SemEval) 2007 workshop, shared task of affective computing [10]. It consists of news headlines from major newspapers such as New York Times, CNN, and BBC News, as well as from the Google News search engine. The annotation is done manually by 6 annotators and the set of labels includes 6 emotions: *anger, disgust, fear, joy, sadness,* and *surprise*. Instead of the usual $0/1$ binary annotation, they run a finer-grained labeling process. An interval

[8] http://www.affective-sciences.org/researchmaterial

[9] http://people.rc.rit.edu/~coagla/affectdata/index.html

[10] http://nlp.cs.swarthmore.edu/semeval/tasks/task14/data.shtml

$[0, 100]$ is set for each emotion and the annotator decides to what degree from 0 to 100 the headline shows that emotion. Hence, a headline can have multiple emotions, each with a different degree. To justify why news are selected to build this dataset, they claim that news have typically a high load of emotional content and are written in a style meant to attract the readers' attention. In fact, there is a popular concept in news world, called *Emotional Framing* [60], saying that each news item is shaped to a form of story with layers of dramatic frames, such as emotion *fear* caused by danger or alarming news. Although this idea backs up the development of SemEval 2007 dataset, our statistical analyses show that the data are most likely to be neutral and there is not much tangible emotion expressed by news. For example, the average degree of all emotions for a headline is only 15.48 (out of 100) in average. Also, in a coarse-grain scale, if we define that a headline shows emotion e, if its degree $d_e \geq 50$, and does not express e if $d_e < 50$, then only 6.8%, 3.6%, 11.6%, 13.6%, 15.6%, and 3.2% of headlines express *anger, disgust, fear, joy, sadness,* and *surprise*, respectively.

2.4.4 TEC

S. M. Mohammad [50] created a corpus of emotional tweets from Twitter, called "*Twitter Emotion Corpus (TEC)*" [11] in 2012. He targeted the 6 basic emotions proposed by Ekman [20]: *anger, disgust, fear, joy, sadness,* and *surprise* and chose 6 hashtags addressing these emotions (e.g. #anger, #disgust, etc.) to search for appropriate tweets using Twitter Search API [12]. He discarded very short tweets, those with very bad spellings and also those with the prefix "RT" that are retweets of other tweets. We use some of the ideas exploited in creating process of *TEC* when building our corpus, that will be discussed in details in Chapter 3. After this post-processing, *TEC* includes $21,051$ tweets where 7.4%, 3.6%, 13.4%, 39.1%, 18.2%, and 18.3% of the corpus have labels *anger, disgust, fear, joy, sadness,* and *surprise* respectively. This shows how imbalanced this dataset is.

[11] `http://saifmohammad.com/WebPages/lexicons.html`
[12] `https://dev.twitter.com/docs/using-search`

Chapter 3

Cleaned Balanced Emotional Tweets (CBET) Dataset

In this chapter, we explain the details of building our corpus. To do so, we focus on Twitter for several reasons. First, with millions of active users, Twitter reflects daily thoughts and concerns of people beyond compare. Second, being free to use, it promises a wider diversity of users for Twitter. Third, Twitter data are publicly available to download and several tools are developed that help fetching tweets in a fast and easy way. We use NodeXL[1] which is a free extension on Microsoft Excel and allows us to search for tweets based on a keyword.

The motivation that backs up collecting a new dataset for emotion classification research is twofold :

1. Those Twitter datasets that are collected for other research purposes have very few emotional tweets, making them inappropriate for emotion research use. Examples include [61, 62]. Hence, a devoted corpus for emotion research is required indeed.

2. To the best of our knowledge there are only three datasets from English tweets, available for public use [47, 50, 48] where emotion expression is labeled. Each of these corpora have drawbacks that make them open to criticism for being used in emotion mining research. W. Wang et al. [47] use keywords that are not really reflecting the proper emotion, such as the use of the hashtag #embarras as a clue for tweets having the emotion *sadness*. S. M. Mohammad's dataset, called TEC [50], is imbalanced and labeling in M. Hasan's dataset [48] is based on a very different model of emotions which has only two dimensions of active-inactive and happy-unhappy. Therefore, it seems a new dataset is needed to overcome the drawbacks of previous ones.

As users write hashtags or actually "label" their tweets only for stating their status, and

[1] http://nodexl.codeplex.com/

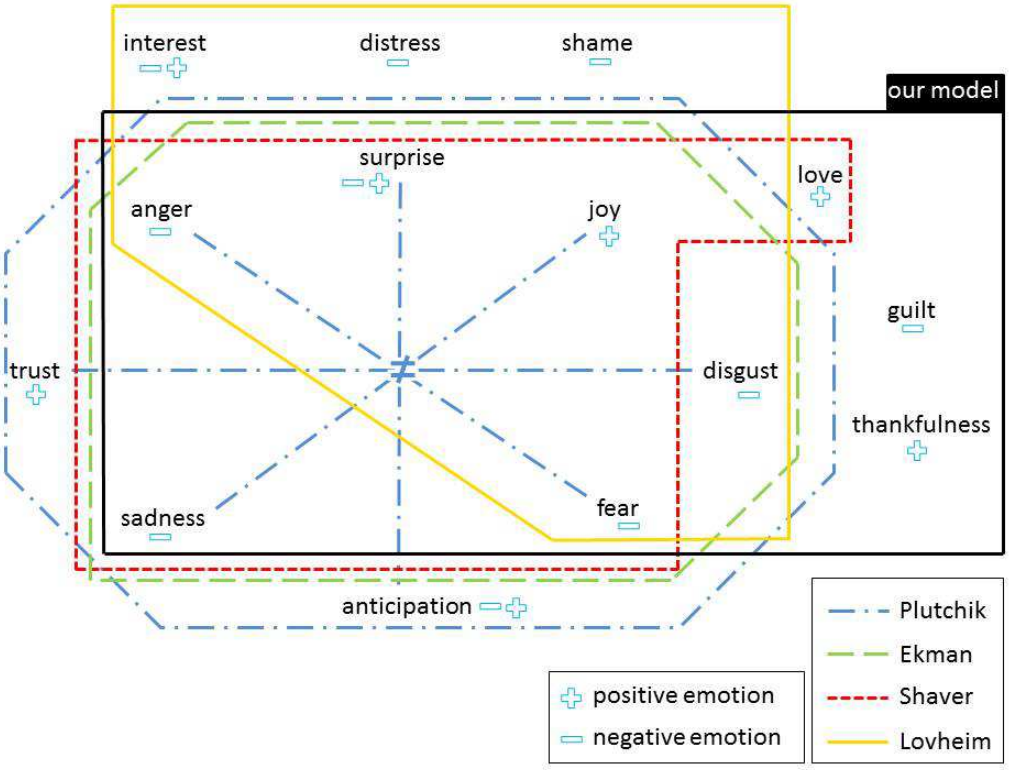

Figure 3.1: The illustration of our model of emotions among others

not specifically for a classification task, labels seem not to be 100% trustworthy. However, previous research has shown that hashtags serve as acceptable emotion labels for tweets [47, 48]. They validated this type of labeling by comparing them to manual labeling for a set of randomly selected tweets. Therefore, we decide to use this finding by searching for tweets with emotional hashtags and use those hashtags as tweets' labels. Nevertheless, note that there might be some situations such that there are emotions in a tweet that the writers just did not feel compelled to include a corresponding hashtag in their tweet or there was not enough space to do so. Handling such cases is harder since the label is not provided but the content is there. These cases are not addressed in the literature yet.

Our work is based on P. Ekman's model of basic emotions [20] as well as P. Shaver's [17], which later was explored more by W. G. Parrott [63]. Ekman states that there are 6 basic emotions: *anger, disgust, fear, joy, sadness*, and *surprise*. Shaver and Parrott suggest the same basic set of emotions with the exception of removing *disgust* and adding of *love*. We merge the two models and also add *thankfulness* and *guilt*. Figure 3.1 illustrates our model of emotions among others. The lack of positive basic emotions, as discussed in Chapter 2, motivates us to consider *thankfulness* since it is a positive emotion that captures

Table 3.1: Hashtags used to search for tweets

Emotion	List of Hashtags
anger	#anger, #angry, #rage
fear	#fear
joy	#happy
love	#love
sadness	#sad
surprise	#surprise
thankfulness	#thankful
disgust	#disgust, #disgusted, #disgusting
guilt	#guilty, #sorry

some important situations in daily human activities. *Guilt* is known as a basic emotion by some other psychologists such as C. E. Izard [64] but is not included in neither Shaver's nor Ekman's. We embrace *guilt* in our model because detecting it helps psychologists determine if a patient faces or will face depression or stress. They can even predict if the patient would commit suicide in the future. This is an extremely important use case of emotion mining research and we include the detection of *guilt* emotion in our work to address such cases.

Table 3.1 shows the corresponding hashtags that we use as keywords in NodeXL to search for tweets of each emotion. According to this table, in the cases of *anger*, *disgust*, and *guilt* more than one hashtag is used to retrieve emotional tweets. The reason is that the number of tweets fetched using only one hashtag was not sufficient and would make the dataset imbalanced, so we added more hashtags by making very slight variations (i.e. variations in the word or synonyms) in order to take more tweets. In addition, the hashtags #anger, #fear, #love, #surprise, and #disgust are identical to the name of their corresponding emotions. The rest of the hashtags, however, differ slightly. For instance, we preferred #happy over #joy for the emotion *joy* because #happy is a more informal and common word for describing joy on Twitter. The same reason applies for #thankful over #thankfulness for *thankfulness* and #sad over #sadness for *sadness*.

A total of $208,544$ tweets were collected in a time frame of 4 weeks from Oct. 31st 2014 to Nov. 27th 2014. Tweets of this corpus do not belong to any specific domain and form a general-purpose dataset suitable for analysis of people's day-to-day use of Twitter.

3.1 Preprocessing

Since tweets are written by the crowd, they contain a significant number of informal words, short forms and abbreviations, special symbols, and spelling errors. These anomalies add

noise to the input data that is going to be used by a classifier in the training step and can make the prediction process more complicated in the testing step. This is why extensive cleaning of the crawled data is required before being used for any purpose. In what follows, the details of preprocessing are explained:

- NodeXL sometimes fetches repeated tweets, so a total of $96,048$ duplicate tweets are detected and removed.
- As our focus is only on English text here, non-English tweets are deleted. To this end, we use a language detection library for Java by N. Shuyo [65] which has the precision of over 99%. $21,599$ non-English tweets are detected and removed using this library.
- A total of $1,691$ tweets that contain 5 mentions or more (mentioning other users, with the pattern of "@" followed by a username, such as @Graham) are removed. Given the 140 characters limit for each tweet, we believe that if a tweet contains at least 5 mentions, each taking 11 characters in average, it could not have enough meaningful information to process or predict.
- All capital letters are converted to small ones, as to make all the similar words in a single unified shape.
- As mentioned before, hashtags are space-free phrases. In many cases, people write several words back to back in a hashtag form. Keeping such hashtags in their original form reduces the readability of the tweet for machines and even for humans. Therefore, we segment these phrases and detect their constituent words. For this purpose, we use a python word segmentation library [66] which exploits Google's N-gram corpus. After segmenting, the original form of hashtag as well as its single words are kept. For example, having the hashtag "*#animalrights*" in a tweet, the words "*animal*" and "*rights*" will be added while "*#animalrights*" is also preserved.
- All URLs, stop words, numbers, useless punctuation marks, and redundant white spaces are removed. These not only do not bear any significant and useful meaning, but also increase the number of words involved in the corpus, which in turn makes the task of training a classifier harder. Note that as stated in Chapter 1, emoticons are not considered as normal words and are specific to informal text but since they are valuable resources of emotional information we did not remove them. Table 3.2 shows the list of 96 preserved emoticons.

Table 3.2: List of preserved emoticons

:)	:-)	:))	:-))	:(	:-(	:((	:-((	:\")	:\">	:-?	:/
:-/	:\\	^_^	:<	:-<	:p	:-p	:P	:-P	>:P	>:p	;)
;-)	;]	:-]	;D	:O	:-O	:o	:-o	8-<	:|	:-|	-_-
=_=	>_<	o_o	O-O	O_O	O_o	o_O	<3	</3	\\o/	%)	%-)
#-)	:&	:-&	>:)	:X	:-X	:x	:-x	:#	:-#	:$	:*
:-*	D:<	v.v	:')	:'-)	:'(	:'-(	:-||	:@	>:(	;(	:c
:-c	:[	:-[	:{	>:[	:D	:-D	x-D	xD	XD	=D	=-D
=3	=)	:}	:^)	:o)	:]	:3	:c>	:>	=]	8)	8-)

Table 3.3: Some samples of CBET

Tweet	**Hashtag**	**Label**
Say No Fur! animal rights #animalrights #vegan #compassion	#disgusting	disgust
@user We send super surprise gift to japan #Fiverr #gift #japan	#surprise	surprise
Thank Lord Thank blessings guiding everyday You never fail #blessed	#thankful	thankfulness
#jewelry #sets Vintage shell necklace matching earrings real gold marked	#love	love
#rain #hail #thunder #storm	#fear	fear

- All mentions to all users are changed to a single unified form, "@user".
- Very short tweets (those with less than 3 words) are removed. These tweets are not likely to convey enough information about their writer's emotion. This step omitted $1,121$ tweets.
- Tweets having several emotional hashtags (e.g. "The good thing is happening #happy #love.") are separated from singly labeled ones in order to not add ambiguity in identifying their label. This subset of data, containing $4,325$ tweets, will be used later in multi-label classification task and is explained in the next chapter.
- There are some tweets that look very similar to each other and could probably be written by software robots. In order to detect them we use the Dice coefficient which states that if *A* and *B* denote the bag-of-words (BOW) of two tweets and $|A|$ and $|B|$ show their sizes, then:

$$s = \frac{2|A \cap B|}{|A| + |B|} \tag{3.1}$$

s ranges between 0 and 1 where 0 shows two texts do not have any word in common

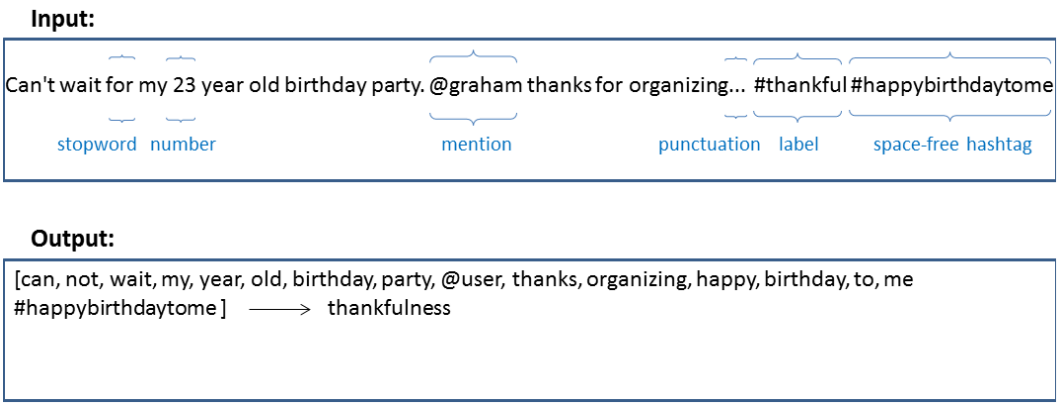

Figure 3.2: An example of running preprocessing steps

and 1 shows two texts are identical. In our work, if $s > 0.3$ for two tweets, only one of them is kept. This threshold is obtained by some manual checking of similar and non-similar tweets. Following this technique, about $6,900$ tweets are omitted. The Jaccard similarity coefficient can be used alternatively; however, we detect more cases of similar tweets when using Dice's.

- The remained tweets are tokenized. Tokenization is the process of decomposing a text into its constituent parts, that can be single words, phrases, etc. This is a crucial step in all text mining works as it extracts meaningful details from the text that has previously been seen as a chain of characters only. Our experiments show that although the Stanford tool [67] is a powerful tokenizer, it does not work properly for informal contents of tweets. For example, given the text "*@Graham*", the Stanford tokenizer splits it to "@" and "*Graham*" while this text is a single unit of word, used to mention a user in Twitter and we do not intend to split it. Therefore, we tokenize the tweets only based on space character. However, it should be noted that there are still many special cases that should be taken care of. For instance, "*can't*", "*I'm*", and "*easy-to-understand*" cannot be split based on space character.

- Finally, hashtags that serve as the label of tweet are removed.

Figure 3.2 shows an example of execution of the preprocessing steps on a sample tweet. The message before and after preprocessing is illustrated. The interesting point is that even though the output vector is less understandable for humans, it is more meaningful and informative for a machine. This is actually the whole idea behind cleaning and preprocessing texts. Also, we should note that in text preprocessing systems, after the tokenization step,

words are often checked with an English dictionary in order to remove misspellings and meaningless words. In our special case, however, we leave the words as they are. The justification for this is that in informal writings, some people tend to show their focus or interest on something using character flooding of that word, e.g. *"something greeeeeeaat happened today!"*. If we were to keep only dictionary words, we would lose words such as *"greeeeeeaat"* that can be rich sources of emotional information for us. Moreover, some slang, prevalent in Twitter, is not included in standard dictionaries, e.g. "*lol*" and "*cya*" meaning "*laughing out loud*" and "*see you*", respectively. By pruning such words, we again lose some information.

After preprocessing the original corpus, $76,860$ tweets are left. To have a balanced dataset, we select $3,000$ samples (tweets) of each emotion which gives us a dataset with total of $27,000$ samples. From now on, we refer to this dataset as *Cleaned Balanced Emotional Tweets (CBET)*. Table 3.3 shows some tweets of *CBET*. We plan to explore larger portions of the cleaned corpus in future works.

3.2 Threat to validity

Choosing appropriate hashtags to search for tweets is probably the most important part in the creation of our dataset. There are two opposite perspectives in this regard. From one point of view, hashtags should be as close as possible and even identical to the emotion they intend to represent. For instance, #anger is the best choice for *anger* and other possible hashtags should be avoided in order not to have discrepancy in retrieved tweets. The other perspective, however, suggests that different synonyms of an emotion should be considered as hashtags to ensure that a variety of tweets are collected and the dataset is not overfitted over one particular hashtag and the words used along that hashtag. Tweets labeled as *anger*, *disgust*, and *guilt* in CBET can be subject to criticism from the proponents of the first idea because we use additional hashtags such as #rage and #sorry for them. Tweets of other emotions, on the other hand, can be questionable from the second perspective, as they stick only to one hashtag and do not cover different variants of an emotion. Overall, we believe that this dataset is a good representation of emotional Twitter data. We used one hashtag in some cases because we think it is the most prevalent hashtag that people use to address that emotion and used more in other cases because there are different words that people exploit to describe those emotions.

Chapter 4

Emotion Classification on CBET

In this chapter, we adopt methods that automatically predict the emotion of a tweeter described in his/her tweet. The problem can mathematically be formulated as follows: Let $E = \{e_1, e_2, ..., e_k\}$ be the set of k possible labels (emotions) and $S = \{s_1, s_2, ..., s_n\}$ be the set of n training samples where each s_i is associated with an e_j. The problem is to find a function $h : S \rightarrow E$ such that as many samples as possible from S can fit in h. Then h can be used to predict the emotion e_t associated with a test sample t. Note that in ***CBET***, k is 9 and n is 27,000, if we use up the whole dataset for training but would be less if we intend to reserve part of the data for testing purpose. Later, we use n to refer to the number of training samples, even though it is a subset of the 27,000 tweets.

We first introduce a lexical-based approach toward the problem and then explore several settings of learning-based methods. In particular, feature selection, dimension reduction, different configurations and learning algorithms are investigated. Afterwards, we try to address the problem of sparsity in data and explore the area of multi-label emotion classification.

To evaluate and compare different methods we use two metrics, taken from the information retrieval field, called precision and recall. For emotion i, precision and recall are defined as:

$$precision_i = \frac{|TP_i|}{|TP_i + FP_i|} \tag{4.1}$$

and

$$recall_i = \frac{|TP_i|}{|TP_i + FN_i|} \tag{4.2}$$

where TP_i or True Positive is the set of samples that have correctly been classified, FP_i or False Positive is the set of samples having a label other than i that are predicted to

have i, and FN_i is the set of samples having the label i that are predicted to have another label. The values of precision and recall over all labels is simply the average of precision and recall of each label. F-measure or F1 score is the harmonic mean of precision and recall and is calculated as:

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{4.3}$$

Note that the average F1 score over all labels is not equal to the average of F1 scores for each label and should be calculated from the averaged precision and recall.

In order to compare the results, one can think of a simple baseline as a majority voting classifier that assigns to all test instances the label with the maximum frequency among training instances. Since our training set is perfectly balanced, this can be done by selecting a random emotion for each test instance. Such a classifier has a precision, recall and F1 measure all equal to $1/9 \sim 11.11\%$ averaged over all emotions. This very low number indicates how difficult the task is.

4.1 Lexical-based method

One of the very widely used approaches toward the problem is the lexical-based method. The simple intuition behind this technique is to look for emotional clues inside the text. In these approaches, one or more external resources are exploited for classification. Most frequently, these resources are in the form of lexicons that contain information about the emotion(s) or at least the polarity that words or phrases convey. Having such lexicons, the content of a message is evaluated based on the emotion(s) that its words or phrases have and a decision is made based on this information. In the problem of working with tweets, most of the existing lexicons are not suitable to use due to heavy load of abbreviations and informal language used in them. Therefore, we decided to build an emotion lexicon from our Twitter corpus.

The idea of developing this emotion lexicon is adopted from [56], explained in Section 2.3. More concretely, dividing the corpus into training and test sets, we inspect the training set S word by word to see which words express which emotions and to what degree that is done. This is the whole idea behind our lexical-based method. For this purpose, we build a lexicon from the vocabulary V of all the single words (unigrams) contained in S. The lexicon is actually a $V \times E$ matrix where the element at index (j, i) denotes the degree that the word w_j expresses emotion e_i. In other words, each word has a corresponding

weight vector that contains weights associated to each of the 9 basic emotions. The weight $F(e_i|w_j)$ is calculated as the number of times that w_j has occurred in tweets that have label e_i in the training set. That is:

$$F(e_i|w_j) = \sum_{s \in S} F(e_i|s) \times I_s(w_j) \tag{4.4}$$

where $F(e_i|s)$ is the presence of emotion e_i given sample s and $I_s(x)$ is an *indicator function* which is equal to 1 if $x \in s$ and is 0 otherwise.

The naïve assumption supporting this idea is that all the words in a tweet are in agreement with the label of that tweet. For example, if the training set contains *"Today is my birthday"* with label *joy*, *"I just forgot my mother's birthday"* with label *sadness*, and *"Hey! I was invited to her birthday!"* with label *joy*, then the weight vector for word *birthday* would be $\{0, 0, 2, 0, 1, 0, 0, 0, 0\}$ where index 3 and 5 are corresponding to *joy* and *sadness* respectively.

When classifying a new tweet, the weight vectors of its unigrams in the previously built lexicon are looked up and aggregated. The emotion that has the maximum aggregated weight would be the predicted label for the tweet, if we want a single label per tweet.

Table 4.1 indicates the precision, recall, and F1 measure values, all in percent, after executing the lexical method on the corpus. All results are averages of 5 independent runs of the experiment. In each run, the corpus is shuffled and then 75% of tweets are randomly selected to be the training samples and the remained 25% form the test set. This is the convention that we follow for the rest of this chapter. As the table shows, the average F1 measure for all emotions is 40.50% which is a great improvement over the baseline. *Thankfulness, love,* and *fear* are the easiest emotions to predict while *sadness* and *anger* are the hardest.

The lexical-based method is easy and fast to build; however, it has some major drawbacks listed as below:

1. If an external lexicon is to be used, it is many a time hard to obtain and more often than not domain-specific. If the lexicon is built from the working dataset, the model may not be reusable for other datasets and the whole process should be repeated for any new collection.

2. Even if the lexicon and training data are taken from the same domain, some words have different meanings in different sentences. For example, "*I had a great time*

Table 4.1: Results of running lexical method

Emotion	P	R	F1
anger	40.28	24.10	30.10
fear	55.96	39.48	46.27
joy	46.88	35.52	40.39
love	51.50	43.58	47.17
sadness	30.69	24.54	27.26
surprise	48.00	34.62	40.20
thankfulness	42.36	57.26	48.64
disgust	43.50	30.34	35.73
guilt	23.14	58.86	33.16
ALL	**42.48**	**38.70**	**40.50**

with my grandfather today" and "*My great-grandfather passed away yesterday*" show different meanings of "*great*".

3. Syntax structure of sentences can also influence the interpretation of words even if the meaning is clear. For instance, "*I laughed at him*" and "*He laughed at me*" differ only in the order of words, nevertheless, they most probably have different emotions from writer's point of view. These linguistic information are not usually included in normal lexicons and should be added in the form of an ontology [30].

4.2 Learning-based methods

Machine learning approaches have shown very good results in sentiment classification of text messages. These methods essentially try to learn patterns from a training set in which messages are labeled and then these patterns are used to guess the label of some new messages that the algorithm has not seen before.

The Support Vector Machines (SVM) approach is a well-known and widely used machine learning algorithm. SVM is a linear classification algorithm that tries to fit a hyperplane separating the samples of one class from the other such that it has the largest distance to samples of both classes. Hence, it is basically a binary classifier. It is used in the field of textual emotion classification too, such as in [68, 31, 10, 39, 48, 49, 50].

Considering the capabilities of the binary SVM, we decide to use it as the learning algorithm for our task. In order to have a 9-class classifier, we train 9 SVM classifiers, one for each emotion. This means that SVM i is able to predict to what degree a tweet has emotion i. This is shown by a probability in the interval [0, 1] meaning that the higher the probability, the stronger that tweet conveys emotion i. The emotion that has the highest

probability among all 9 emotions will be predicted as the label of the test tweet. For an SVM responsible for learning emotion i, there are 3,000 positive samples (i.e. tweets with label i) and 3,000 negative samples (i.e. those with any label other than i). Negative samples are selected using an undersampling process to make the balance between samples of the two classes. To do the undersampling, the negative samples are randomly permuted and then the first 3,000 ones are selected.

Selecting features that distinguish samples of different classes plays an important role in the performance of SVM. We experiment several configurations of feature selection. The most straight-forward way to come up with a set of features, is to use a lexicon and represent each tweet by a binary vector such that element i in the vector is 1 if the message has the word i from the lexicon and is 0 otherwise. This way, the features are those constituent words (unigrams) of text that exist in that lexicon and the set of all features is called "bag of words" since the words that are used in the training samples are considered but their order in making sentences is ignored. This method, generates presence-based features as it only keeps information about presence or absence of words (features) in binary format. Another alternative, namely frequency-based features, captures how many times each word is occurring in the text and the value of the features are thus positive integer values, instead of binary. Our results did not show any improvement when using frequency-based features, so we stick to presence-based representation of samples. Therefore, the input to the SVM algorithm can be seen as a $S \times V$ matrix, where each row is a training sample and each column is a vocabulary word taken from a lexicon.

We run 3 experiments each with a different lexicon suitable for the emotion detection task that are introduced in Section 2.3: LIWC, NRC, and NRC-hashtag. If a word from the lexicon is not used in at least 3 of the training tweets, that word is removed from the feature set. The reason is that such words very much enlarge the feature space but very rarely contribute to describing the messages. The feature set sizes after removing rare words is 440, 600, and 1300 when exploiting LIWC, NRC, and NRC-hashtag, respectively. Other lexicons such as Wordnet Affect and WPARD have a much smaller feature set and will probably produce poor results over *CBET*.

For the implementations, the Libsvm library [69] is used which is an easy-to-use and efficient implementation of SVM algorithm. Values of γ, C, and ϵ parameters are set to 0.1, 0.1, and 0.01 respectively and a linear kernel is selected. The values of these parameters and type of kernel are chosen by running some parameter tuning experiments such that in each run, the value of only one parameter or type of kernel is changed and all other param-

Table 4.2: Results of running SVM fed with words from 3 lexicons

	LIWC			NRC			NRC-hashtag		
Emotion	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
anger	39.57	21.99	28.22	31.31	21.42	25.41	39.55	29.27	33.59
fear	60.75	39.88	48.13	42.83	38.03	40.27	59.71	53.26	56.27
joy	41.68	26.54	32.38	29.30	10.14	14.92	50.01	36.75	42.34
love	49.19	20.36	28.78	47.17	19.86	27.84	52.84	48.67	50.65
sadness	29.54	16.78	21.30	22.87	12.73	16.27	34.87	26.13	29.83
surprise	41.94	23.23	29.82	39.04	30.51	34.18	48.02	37.04	41.78
thankfulness	47.32	45.55	46.41	47.10	39.91	43.16	56.38	53.72	54.95
disgust	16.60	66.26	26.55	33.76	34.65	34.19	39.44	43.13	41.16
guilt	28.75	19.41	23.10	14.60	50.83	22.67	24.24	52.35	33.11
ALL	**39.48**	**31.11**	**34.80**	**34.22**	**28.67**	**31.20**	**45.01**	**42.26**	**43.59**

eters are kept fixed. Table 4.2 shows results of testing of 3 SVM-based classifiers using the mentioned lexicons. As mentioned before, all numbers are averages of 5 independent runs of each experiment. When words from NRC-hashtag are used as feature set, the performance is much higher compared to LIWC and NRC. This reveals two important facts. First, although SVM has a great performance on many domains, it is very much dependent on its defined feature set. Second, the *nature* of data is a key factor in selecting appropriate features. In our problem, since NRC-hashtag lexicon is taken from Twitter (although from a different corpus than our training data), it describes features of data, such as informal words and abbreviations, more precisely; while other lexicons contain only standard and formal words which are less likely to be used in tweets.

Considering the nature of data reflected in this experiment, it seems that if we had the vocabulary of our own data, it would have had the maximum relevance and would boost the F1 value. Hence, we decide to take the feature set directly from the training tweets, by selecting those unigrams that have occurred at least 3 times in the whole training set. The size of features now is about 3,000 for each SVM. Figure 4.1 shows the overall procedure of our method for building an SVM-wise solution, which we call *the core SVM model*. In the test phase, the emotion that has the highest probability among all 9 emotions is the predicted label of the test tweet. This is depicted in Figure 4.2.

Table 4.3 part *a* shows the classification results using this model on the test data. The F1 value has an average of 47.05% over all emotions which is 6.55% higher than lexical-based method, showing a significant improvement. Standard deviation of precision, recall, and F1 values over 5 runs of the experiment are reported in Table 4.4. The value of standard deviation in all cases is lower than 0.03% which shows the core SVM model is stable over

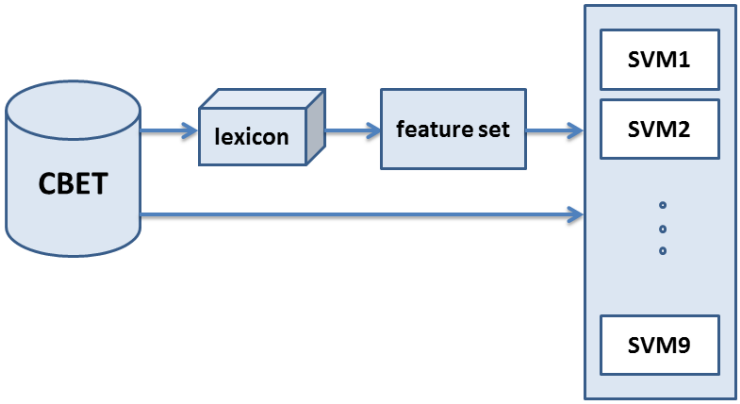

Figure 4.1: The overall procedure of training in the core SVM model

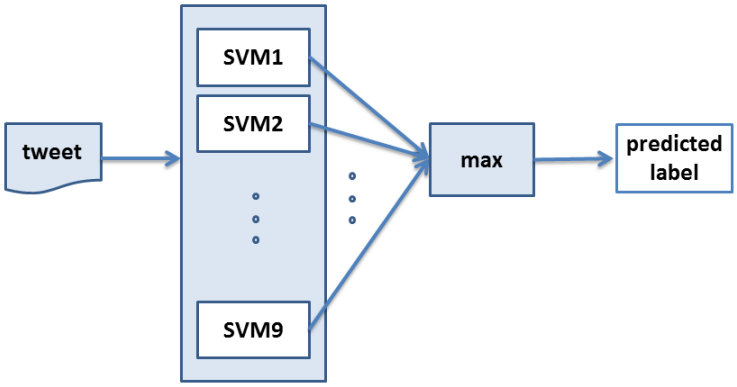

Figure 4.2: The overall procedure of testing in the core SVM model

Table 4.3: Results of running SVM fed with words from our lexicon

	(a) unigrams (core)			(b) informativeness			(c) informativeness + emoticons		
Emotion	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
anger	39.92	36.50	38.08	39.38	38.59	38.95	39.10	38.74	38.90
fear	56.61	57.31	56.95	57.18	58.26	57.70	57.24	58.21	57.70
joy	48.30	46.73	47.48	48.66	44.48	46.47	48.90	44.77	46.74
love	55.51	52.82	54.07	54.90	53.75	54.29	54.88	54.02	54.41
sadness	36.45	28.87	32.19	33.76	31.14	32.39	33.89	31.36	32.55
surprise	48.09	45.17	46.57	48.97	45.86	47.34	49.21	45.62	47.33
thankfulness	58.32	59.07	58.66	57.96	58.41	58.15	58.31	58.38	58.32
disgust	41.20	51.14	45.62	42.66	48.97	45.55	42.29	48.92	45.31
guilt	39.22	45.72	42.14	40.88	44.65	42.63	41.25	44.56	42.79
ALL	**47.07**	**47.04**	**47.05**	**47.15**	**47.12**	**47.14**	**47.23**	**47.17**	**47.20**

randomness.

A confusion matrix can help better analyze the results. Table 4.5 shows the confusion matrix for the core SVM configuration. The values in this table are averages of the afore-

Table 4.4: Standard deviation of running the core SVM model

Emotion	P	R	F1
anger	0.019	0.022	0.013
fear	0.015	0.011	0.011
joy	0.017	0.011	0.010
love	0.027	0.023	0.016
sadness	0.014	0.025	0.018
surprise	0.019	0.005	0.009
thankfulness	0.027	0.022	0.020
disgust	0.011	0.016	0.009
guilt	0.027	0.020	0.012

Table 4.5: Confusion matrix for the core SVM model

Emotion	anger	fear	joy	love	sad	surprise	thankful	disgust	guilt
anger	**275.6**	61	50.8	31.6	65	45.2	28.4	110.2	87.8
fear	44.8	**433.4**	30.2	31.2	41.2	26	30.4	69.6	49.4
joy	35.8	26.2	**350.6**	99.2	30.2	82.4	65.4	19.8	40.4
love	29	46	86	**397**	27.4	49	43.4	33.8	39
sad	82	49.2	39.8	33.6	**215.4**	49	40	125	111.6
surprise	53.6	29.8	74.8	50.6	40.2	**338.6**	54	48.2	59.8
thankful	30.8	33	56.8	38.6	34.8	46	**451.4**	34	38.6
disgust	75	48	12.6	13.8	61	28.8	28.8	**380.4**	95.4
guilt	64.4	39.2	24.4	21	75.6	39.4	32.4	102.8	**335.4**

mentioned 5 independent runs of the experiment. The number at row i and column j is the number of samples that have true label i and are predicted to have label j. An ideal classifier should have all the numbers on non-diagonal positions equal to 0. According to this table, the most confusing cases are *sadness* samples that are classified as *disgust* or *guilt*, 125 and 111.6 times respectively. This confusion makes sense in that, when someone feels disgusted or guilty, he/she might feel sad at the same time. In fact, although *sadness* is counted as a basic emotion, it is often accompanied or even raised by other emotions and this is why it has the lowest F1 value among all.

Although the core SVM model gives a reasonable accuracy for a 9-class classification task, we believe that there are some parts of the algorithm that can be improved in the hope of boosting the performance. Selecting proper features, reducing the dimensions of our huge feature space, and applying a suitable learning algorithm are factors that will probably influence the output of the method. Possible improvements in these areas are elaborated in the following sections.

4.2.1 Feature selection

Informativeness: In the core solution, we use all the non-rare unigrams without applying any filter on them; however, it remains questionable whether all the unigrams are useful features. To answer this question, a criterion should be developed that discriminates useful unigrams from misleading ones. We define the notion of *informativeness* as a measure to see how informative a word is. Here, the concept of informativeness is close to *support* and *confidence* from association rule mining [70]. It includes both how frequent the word is and how much useful information it provides for us. The informativeness is calculated using a lexical-based approach. Suppose we have n tweets and using the leave-one-out method, classify each of them based on a lexicon that is built from the other $n-1$ ones and we do this n times so that all tweets are classified. Thus, for a unigram u, the informativeness, t_u is defined as:

$$t_u = \frac{CorrectClassify(u)}{TotalClassify(u)} \tag{4.5}$$

where $TotalClassify(u)$ shows the total number of times that u is used for classification and $CorrectClassify(u)$ is the number of times that we classify a tweet containing u correctly, if we solely use the weight vector of u for classification. In other words, $CorrectClassify(u)$ is an indicator of how much the emotion(s) coupled with u are consistent with the total emotion conveyed by the test tweet. The informativeness value ranges between [0,1] where 0 means the word is not informative at all and 1 shows the word is perfectly informative.

Table 4.3 part *b* shows results when unigrams are filtered by their informativeness value. For each unigram u, only if $t_u \geq 0.5$ the word is considered as a feature. The informativeness filter, as seen in the table, increases the performance, but slightly.

Emoticons: One of the key features of informal texts is the use of emoticons. Emoticons are easy to use and universally understandable symbols that are embedded in text and portray a wide range of emotions and hence are helpful resources for our problem. Particularly in *CBET*, more than 3% of the samples have at least one of the emoticons listed in Table 3.2. The most frequently used ones are :), :(, :D, and ;) that respectively form 34%, 16%, 8%, and 8% of the whole emoticons used in the corpus. We expect that these emoticons would considerably help in emotion detection. Hence, in addition to unigrams, we add some boolean features, one for each of a set of 96 most used emoticons, representing the existence of that emoticon in the tweet. The average results of 5 independent runs

are shown in Table 4.3 part *c*, where in addition to the informativeness filter, the emoticon features are also added in building the classifiers. Exploiting emoticons leads to a slight improvement on precision, recall, and hence F1, which might not be inline with what we expect of emoticons. The reason could be in the method of using them. Further work on coming up with other methods of employing emoticons may lead to more substantial improvements. One suggestion is to use emoticons as final discriminator between two or more labels if their generated probabilities by SVM classifiers are so close that making a decision between them is hard for the system. In such cases, the existence of an emoticon in the test tweet may help to decide in favor of the correct label.

4.2.2 Dimension reduction

From a mathematical point of view, a sample represented by n features can be seen as a point in an n-dimensional space, where each feature is actually a variable or dimension. In cases that n is too large such as in our problem, we need to work within a high-dimensional space. Analyzing and mining data in a high-dimensional space often yields problems that are referred to as the *curse of dimensionality*. One approach to tackle this problem is to reduce the dimensions of data. *Principal Component Analysis (PCA)* is an algorithm that transforms a set of n-dimensional data to a new m-dimensional space where $m \leq n$ such that the new dimensions are linearly uncorrelated.

We use Weka [71] for an implementation of PCA. Weka is an open source workbench in Java, containing a set of useful machine learning algorithms, including PCA. Running PCA reduces the number of unigram features of the *CBET* dataset from about 3,000 to about 2,000 which are then used to train 9 SVM classifiers, similar to the aforementioned core model. This method has the average precision, recall, and F1 value equal to 35.86%, 35.83%, and 35.85%, respectively. Comparing these results to the core model, it can be seen that mapping unigrams to a new space has reduced the performance by more than 11%. It seems that each SVM classifier requires at least a certain number of features to be able to separate the training samples of the two classes efficiently.

4.2.3 Other configurations

There are different ways to do a multi-class classification using a binary classifier such as SVM. The approach that we have been following so far is to train k classifiers, where classifier i learns to discriminate samples of label i against all other samples. Here we describe another configuration of SVM classifiers that allows us to have a multi-class prediction.

Algorithm 1 training procedure of the second SVM model

```
corpus: the training set, containing tweets of all 9 emotions
n: size of training set
k: number of labels
function DoTrain(corpus, n, k)
    declare models[k][k]
    for i = 1 to k do
        for j = i + 1 to k do
            subCorpus = getSubsetByLabel(corpus, i) ∪ getSubsetByLabel(corpus, j)
            models[i][j] = trainSVM(subCorpus, i, j)
        end for
    end for
    return models
end function
```

In this approach, we train $k(k-1)/2$ classifiers, each for learning to distinguish a possible combination of two labels i and j such that $i \neq j$. From the whole training set, such a classifier takes only those samples that have label i and j as it is not meant to care about other labels. In our problem, there are 36 of these classifiers, one for *anger* vs. *fear*, one for *anger* vs. *joy*, and so forth, each trained on 6,000 balanced samples of its two emotions. We call this method *the second SVM model* and the procedure of its training phase is provided in Algorithm 1.

Predicting the label of a test sample is done in a voting system. Depending on the probability that a classifier c outputs, c votes for only one of its two associated labels. In the rare case that the probability is equal to 0.5, c votes for both labels. The label that gets the highest vote is predicted as the emotion of that sample. The procedure of the classification phase is described in Algorithm 2. The simple intuition behind the idea of the second model is that creating a group of classifiers where each of them is expert in only two emotions and is exclusively trained over the samples of those two emotions may show better performance compared to the core model in which each classifier is responsible for detecting samples of one emotion against all the other ones. To put it another way, when a test tweet is given, it is compared between each and every possible pair of emotions, so it may be evaluated more carefully. Consider an example in which the test tweet has the true label *love*. Classifiers trained on pairs excluding *love* are not expert in this particular tweet and may generate close to random results that does not help with detecting the correct label; however, all classifiers related to *love* should vote in favor of it and therefore *love* gets the maximum vote possible for an emotion and consequently the tweet is classified correctly.

The average results of 5 independent runs of the second SVM model on *CBET*, using

Algorithm 2 test procedure of the second SVM model

```
1: models: 2-dimensional list of trained SVM classifiers
2: sample: a test sample
3: k: number of labels
4: function DoTest(models, sample, k)
5:     declare votes[k]
6:     for i = 1 to k do
7:         for j = i + 1 to k do
8:             prob = models[i][j].classify(sample)
9:             if prob > 0.5 then
10:                votes[i] = votes[i] + 1
11:            else if prob < 0.5 then
12:                votes[j] = votes[j] + 1
13:            else
14:                votes[i] = votes[i] + 0.5
15:                votes[j] = votes[j] + 0.5
16:            end if
17:        end for
18:    end for
19:    max = getIndexOfMaximum(votes)
20:    return max
21: end function
```

unigrams as features is shown in Table 4.6. As it can be inferred from the table, the main influence of the second model is on the precision since it is boosted by more than 1.5% compared to the precision of the core SVM model.

Table 4.6: Results of running the second SVM model

Emotion	P	R	F1
anger	36.17	40.44	38.13
fear	62.35	51.96	56.67
joy	47.17	48.14	47.64
love	64.60	49.61	56.11
sadness	34.30	33.49	33.87
surprise	48.57	47.70	48.13
thankfulness	64.54	55.33	59.55
disgust	41.58	49.67	45.24
guilt	38.28	47.82	42.49
ALL	**48.62**	**47.13**	**47.86**

4.2.4 Learning algorithms

So far, we have used the SVM algorithm to train on our training corpus. There are other algorithms that have been shown to be effective on textual data, among them is Naïve Bayes.

Table 4.7: Results of running the Naïve Bayes model

	(a) core model			(b) second model		
Emotion	**P**	**R**	**F1**	**P**	**R**	**F1**
anger	46.55	35.89	40.48	42.82	34.51	38.74
fear	62.01	58.58	60.22	55.57	55.82	55.68
joy	50.25	47.75	48.95	49.11	44.60	46.74
love	71.77	39.66	51.07	73.74	37.99	50.08
sadness	37.71	33.53	35.46	32.63	36.92	34.62
surprise	45.73	52.31	48.78	40.42	52.01	45.47
thankfulness	53.33	65.79	59.01	56.72	59.08	57.87
disgust	47.60	51.48	49.43	44.55	47.47	45.92
guilt	37.24	53.57	43.78	38.53	47.47	42.53
ALL	**50.27**	**48.73**	**49.49**	**48.23**	**46.31**	**47.25**

Naïve Bayes is a simple classifier relying on Bayes' theorem. It uses the very simplistic assumption that all features are statistically independent from each other. Even though this assumption does not hold in textual content, as there do exist correlations between words of a text, or at least words of a sentence, it is shown in the literature that the algorithm still works very well and is even comparable to more sophisticated methods such as SVM. To the best of our knowledge, none of the emotion classification works on Twitter have tried the Naïve Bayes approach.

Similar to the core SVM model, we train a set of 9 Naïve Bayes classifiers on the training data, one for each emotion. Each classifier is fed with a balanced training set consisting of 3,000 positive (expressing that emotion) and 3,000 negative (expressing other emotions) instances. The set of features is taken directly from the vocabulary of the training tweets with rare words removed. We call this *the core Naïve Bayes model* and its results are demonstrated in Table 4.7 part *a*. Naïve Bayes has the average F1 value of 49.49% which is the best, compared to all previous methods. Table 4.8 represents the standard deviations of 5 runs of the core Naïve Bayes model for precision, recall, and F1 values of each emotion. Similar to the core SVM model, standard deviations are very low here which is a proof of stability of the method. Interestingly, the best predictive power is achieved with the *fear* samples, which might mean that people describe their fear feelings clearly and without mixing with other emotions.

The confusion matrix resulting from the Naïve Bayes method is shown in Table 4.9 with the structure similar to Table 4.5. Here, in addition to conflicts in pairs of *sadness-guilt* and *sadness-disgust*, observed previously in Table 4.5, the pairs *disgust-guilt*, *anger-guilt*, and *joy-surprise* show a high confusion as well. However, one of the very intertwined pairs

Table 4.8: Standard deviation of running the core Naïve Bayes model

Emotion	P	R	F1
anger	0.019	0.017	0.009
fear	0.023	0.019	0.015
joy	0.013	0.009	0.004
love	0.017	0.020	0.020
sadness	0.025	0.028	0.023
surprise	0.016	0.010	0.010
thankfulness	0.033	0.023	0.027
disgust	0.012	0.022	0.007
guilt	0.027	0.010	0.014

Table 4.9: Confusion matrix for the core Naïve Bayes model

Emotion	anger	fear	joy	love	sad	surprise	thankful	disgust	guilt
anger	**260.6**	49.2	44.4	14	76.8	54.4	36.2	94.2	113.2
fear	34.2	**455.6**	24.4	10.4	45.8	32	42.8	61.6	60.2
joy	29.8	27.8	**353**	31.4	37.8	112.4	92	12.8	45.2
love	25.2	53.4	107.2	**305.6**	35.2	81	79.2	22.4	51.2
sad	71.2	36.6	34.4	14	**245**	47.2	52.4	103.8	139
surprise	36.4	21.4	63.2	14	45.6	**384**	65.4	30	86.4
thankful	19.8	22.8	37	8.2	40	64.8	**487.4**	25	46.4
disgust	55.8	25.4	12.4	4	73.8	39.6	33	**391.8**	114.8
guilt	45	22.8	20.6	8.8	92.2	51.6	42.6	73.6	**388.2**

of emotions, i.e. *joy-love*, is managed better in the Naïve Bayes classifier (31.4 confused cases) rather than SVM (99.2 confused cases). According to both tables, positive emotions such as *love* and negative ones such as *guilt* or *disgust* are the most separable labels.

In addition to the core model, we develop another model to use binary Naïve Bayes classifiers for our multi-class classification task, similar to the second SVM model, i.e. training $k(k-1)/2$ classifiers, each for learning to distinguish a possible combination of two labels i and j such that $i \neq j$ and then predicting the label of a test sample in a voting system, such that each classifier votes for only one of its two associated labels. We call this *the second Naïve Bayes model* and its performance can be seen in Table 4.7 part *b*. Although the second configuration was able to boost the performance of SVM classifiers, it decreases the values of all measures of precision, recall, and F1 by about 2% in the case of Naïve Bayes.

One observation worth mentioning is that in spite of higher results with the core Naïve Bayes model, SVM has lower differences between precision and recall values corresponding to each emotion. For example, according to Table 4.3 part *a* and Table 4.7 part *a*, SVM has an average of only 4.00% difference between each precision and its corresponding re-

call value over 9 emotions, while this difference for Naïve Bayes is 10.21%. This can be an evidence that SVM performs more robustly and in applications that both precision and recall matter, it might be a better choice. On the other hand, in environments where time is a crucial factor, Naïve Bayes should be considered as the solution since it is much faster than SVM.

4.3 The problem of sparsity

One of the main reasons preventing classification systems from achieving high enough accuracy is data sparsity. This problem is more severe in systems that work on short messages such as tweets. A data matrix is said to be sparse if most of its elements are zero. As noted in the previous section, the input to a learning-based method is a $S \times V$ matrix where each row is a tweet and each column is a word taken from a lexicon. In *CBET* each tweet has 7 words on average while the size of the used lexicon, $|V|$, is about 3,000, meaning that only 7 out of 3,000 entries for each tweet are non-zero. Thus, the input matrix is more than 99% sparse. Here we try to tackle this problem by concentrating on topics transmitted in tweets.

So far, we have focused on lexical clues of texts, i.e. the occurrences of words in them. Now we consider a higher level of words, i.e. the hidden layer of semantic behind each tweet. In other words, we target what the tweeter meant in his/her tweet instead of only searching for verbal signs. To this end, we use the idea of X. Phan et al. [72], who introduce a generic framework for classifying short and sparse texts. Their idea is to discover hidden topics of the training data and use this information to expand samples. At the heart of their method is *Latent Dirichlet Allocation (LDA)* [73], a well-known hidden topic analysis model. LDA is a generative graphical model used for inferring unobserved or latent information from a group of observations such as a set of documents. In LDA, documents are assumed to be random combinations of some latent topics with a Dirichlet distribution and topics are actually Multinomial distributions over words. In a generative process for all documents of the given corpus, a topic is assigned to each word of the document, using Dirichlet distribution. Then a topic for a word placeholder is determined by sampling from Multinomial distribution of topics and finally a word is generated for that placeholder by sampling from Multinomial distribution of that topic. LDA has many parameters to be estimated. Phan et. al. use Gibbs Sampling [74], an approximate estimation method exploiting Markov-chain Monte Carlo model.

The next step after topic inference is to integrate topic distribution into the original

Table 4.10: Results of running Naïve Bayes on expanded tweets

Emotion	P	R	F1
anger	46.23	32.20	37.92
fear	62.09	58.50	60.19
joy	50.41	46.42	48.32
love	71.44	41.42	52.42
sadness	36.00	34.15	34.97
surprise	46.01	52.67	49.08
thankfulness	53.54	65.42	58.87
disgust	48.31	51.99	50.08
guilt	37.51	55.41	44.71
ALL	**50.17**	**48.69**	**49.42**

document. This depends on the classification algorithm that is going to be used afterwards. To have a discrete set of attributes for a document, in which we are interested, the name of a topic, e.g. *"topic:37"*, is attached to the document as many times determined by the probability of that topic for that document. This not only expands each document, but also makes documents more related to each other by showing their common topics.

We follow this approach to alleviate the problem of sparsity seen in tweets. Parameters of number of topics and number of words per topic are set to 50 and 100, respectively. Using a parameter tuning method, we find out that the best result is achieved with these values for the parameters. In the integration phase, different scenarios for combining information from topics and words to the content of tweets are examined. The best result is achieved when the name of the topic that each word of the tweet belongs to is added to the tweet. For example, the tweet *"photos shock at public cow slaughter"* will be expanded to *"photos shock at public cow slaughter topic:35 topic:30 topic:14 topic:1 topic:30"* if *"photos"* belongs to *"topic:35"*, *"shock"* belongs to "topic:30" and so on. Lastly, the core Naïve Bayes model is trained on this less sparse training set. The test data is expanded the same way. Results are shown in Table 4.10. Compared to Table 4.7 part *a*, this method is able to boost the performance for *love, surprise, disgust,* and *guilt* but decreases the performance for the others, such that on average it does not show significant changes, in neither positive nor negative directions.

4.4 Multi-label classification

People sometimes experience situations in which they feel two or more feelings simultaneously, hence they might report several emotions in one single piece of text. For this reason, the classification system should be able to discriminate multiple emotions within one tweet.

Table 4.11: Number of emotions co-occurred together in a tweet

	anger	**fear**	**joy**	**love**	**sad**	**surprise**	**thankful**	**disgust**	**guilt**
anger	–								
fear	87	–							
joy	55	18	–						
love	125	439	1720	–					
sad	180	7	173	248	–				
surprise	2	1	241	375	8	–			
thankful	0	1	132	128	3	11	–		
disgust	15	4	2	6	47	0	1	–	
guilt	9	5	4	25	20	3	2	5	–

The formulation of the problem, explained in the beginning of this chapter can be reformulated as: given the training set S and the set of emotions E, find a function $h : S \rightarrow \wp(E)$ where $\wp(E)$ is the powerset of E. For our problem, the size of powerset $|\wp(E)|$ is 2^9, meaning that there are theoretically 512 possible labels. This is a huge number of labels in classification problems.

According to statistical analyses on our cleaned dataset, about 4% of the data have more than one label. $4,090$, 223, and 12 tweets have 2, 3, and 4 labels, respectively, which forms a total of $4,325$ multi-label tweets. Table 4.11 shows the distribution of each two emotions being mixed in double-label tweets. As it can be seen, the distribution is highly skewed and most of the combinations do not happen very often which is very natural regarding the emotions that humans may experience at the same time. For example, it is very unlikely that a person is thankful and guilty simultaneously. On the other hand, the table shows a few combinations that happen a lot together, such as *love* and *joy*. Overall, about 40% of all double-label data are labeled with the pair of *joy-love* and 50 % of combinations (18 out of 36) happen less than 10 times in the dataset. Combinations of 3 or more emotions cannot be visualized easily. In general, the triples {*joy, love, surprise*} and {*anger, joy, sadness*} are more prevalent than other mixtures. Anyway, the corpus of multi-label tweets is highly imbalanced; however, since it is collected from real data, it could be a sample of the distribution of real world problems. Therefore we decide to leave it as is and do not balance it. The dataset that we work on in this part, finally, is the aggregation of *CBET* and the multi-label dataset, coming to a corpus of approximately $31,000$ tweets.

We adopt the core Naïve Bayes model to address the multi-label emotion classification, because it showed the best results for single-label classifications. We train 9 binary Naïve Bayes classifiers, one for each emotion. The training data fed to them is 75% of the mixed dataset, randomly selected. When classifying a test sample, all the emotions that their

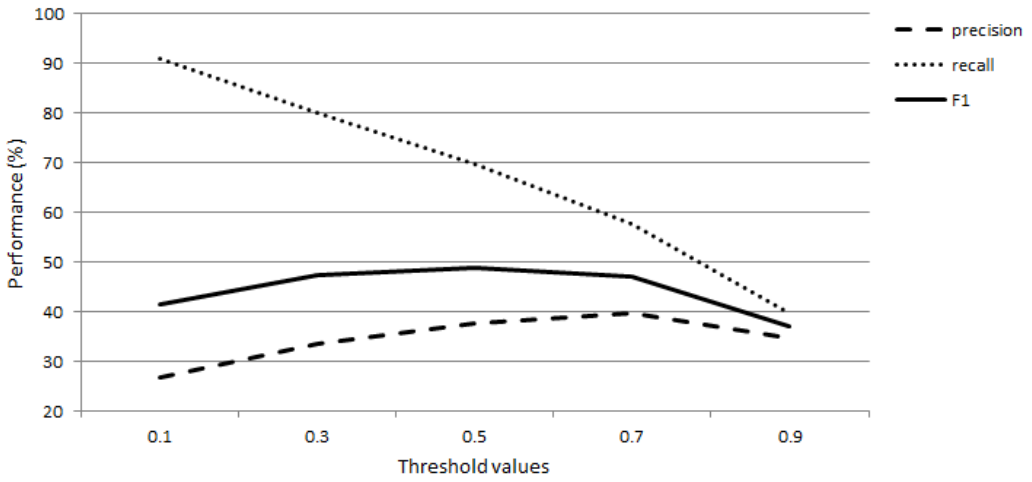

Figure 4.3: Tuning the t parameter

probability is higher than a defined threshold, t, form the predicted multi-label. Determining the best value for t is an influencing part of the algorithm.

Evaluation of multi-label classification is different from that of single-label ones because the predicted labels may not be completely right or wrong, they could be partially right. The precision and recall in this configuration is defined as:

$$precision = \frac{|T \bigcap P|}{|P|} \tag{4.6}$$

and

$$recall = \frac{|T \bigcap P|}{|T|} \tag{4.7}$$

where T and P are the true and predicted set of labels for one sample. Note that here, the precision and recall are calculated for each single sample while in single-label mode, precision and recall are calculated for a set of samples. Hence, the precision and recall values that will be reported later are the averages of precisions and recalls over all samples. F1, as before, is the harmonic mean of precision and recall. Moreover, in spite of the previous experiments, precision, recall, and F1 values cannot be calculated per emotion. Setting the threshold t to 0.5, we get 37.72%, 69.61%, and 48.93% for precision, recall, and F1, respectively. These numbers are averaged over all test samples that form 25% of the dataset and are randomly selected.

As stated above, in the classification phase, we need to set a threshold, t. This threshold ranges between 0 and 1 and controls the probability of the predicted labels for a sample to

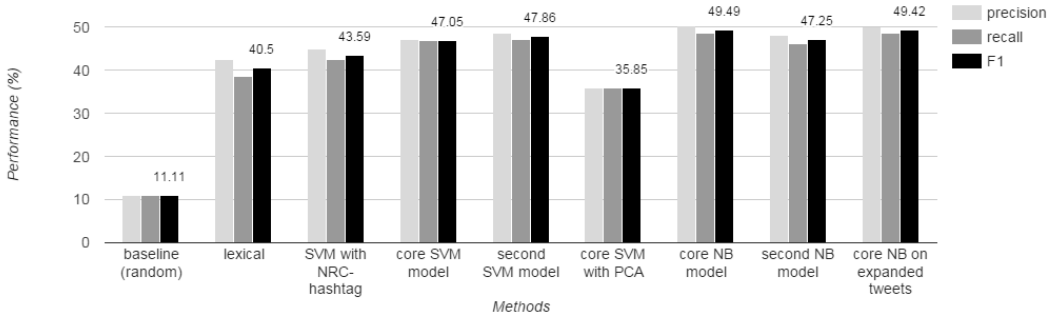

Figure 4.4: The summary of performance of the proposed methods

be higher than a predefined value. Figure 4.3 shows the impact of different values for t on precision, recall, and F1 values. The lesser the value of t, the more the number of labels to be predicted for each sample. Predicting more labels for a sample improves the recall as it increases the probability of predicting one or all of its true labels (increases the nominator in Formula 4.7). This is inline with what the figure shows. Conversely, the precision value drops with smaller values of t. Hence, in systems that recall is a more important factor, smaller value of t should be chosen. If high precision is required, $t \simeq 0.7$ is the best choice and the best F1 value is achieved by $t = 0.5$.

4.5 Summary

In this chapter, we showed that the emotion described in a tweet can be predicted automatically fairly accurately. We introduced several methods for tackling this multi-class emotion classification problem. The precision, recall, and F1 value of some of the best performing methods is summarized in Figure 4.4. The value of F1 for each method is shown on top of its associated bar. The core Naïve Bayes model is the best of all methods, and using the same model on expanded tweets is very close as well, which might be a clue that putting more efforts on solving the problem of sparsity may lead to better results.

Overall, almost all of the proposed methods greatly outperform the baseline. Deciding on which method is definitely superior is not straight forward as it depends on feature selection, parameter values, the applications and other criteria. However, learning-based methods guarantee more promising results compared to lexical-based method probably because they take into account not only the lexical cues of tweets, but also learn more sophisticated models on top of them. Regarding the individual emotions, it seems that *sadness* is one of the most confusing basic emotions and often is predicted with a lower accuracy, compared to others. This might be due to the fact that people sometimes misinterpret their negative

feelings and think it is just the *sadness*. Therefore, the crowd labeling of texts for *sadness* may not be trustworthy enough.

Chapter 5

Emotion Classification on Other Datasets

The proposed methods introduced in Chapter 4 achieved acceptable results over our Twitter corpus, *CBET*; nevertheless, it is required to assess the methods on other datasets and compare the results with state-of-the-art methods. For this purpose, in what follows we test our methods on another Twitter dataset and a dataset of formal documents. Afterwards, we explain how our system is integrated in a social network analysis tool called Meerkat.

5.1 TEC dataset

Twitter Emotion Corpus (TEC) is collected by S. M. Mohammad [50] in 2012. As noted in Chapter 2 Section 2.4, Mohammad targets 6 basic emotions: *anger, disgust, fear, joy, sadness,* and *surprise* and searches for tweets having a hashtag corresponding to one of these emotions. After pre-processing, *TEC* includes 21,051 tweets where 7.4%, 3.6%, 13.4%, 39.1%, 18.2%, and 18.3% of the corpus have the aforementioned emotions, respectively, which shows the corpus is imbalanced.

In order to address the emotion classification problem, the author builds 6 binary SVM models with Sequential Minimal Optimization [75], one for each emotion, using unigrams and bigrams as features. When classifying an unseen tweet, for each emotion the corresponding classifier is applied to decide whether the tweet has that emotion or not. This way, a tweet may get zero, one or multiple labels. Precision, recall, and F1 value of this method is shown in Table 5.1. According to this table, *joy* and *disgust* have the best and worst prediction results, respectively. However, the effect of the size of training samples for each emotion should not be neglected. The better results for *joy* may be due to the large number of training tweets labeled as *joy* (39.1% of the dataset). Figure 5.1 shows

Table 5.1: Results of Mohammad approach on TEC

Emotion	P	R	F1
anger	37.3	22.31	27.9
fear	59.6	43.9	50.6
joy	64.5	60.4	62.4
sadness	41.9	36.0	38.7
surprise	50.6	40.5	45.0
disgust	30.7	13.4	18.7
ALL	**47.4**	**36.1**	**40.98**

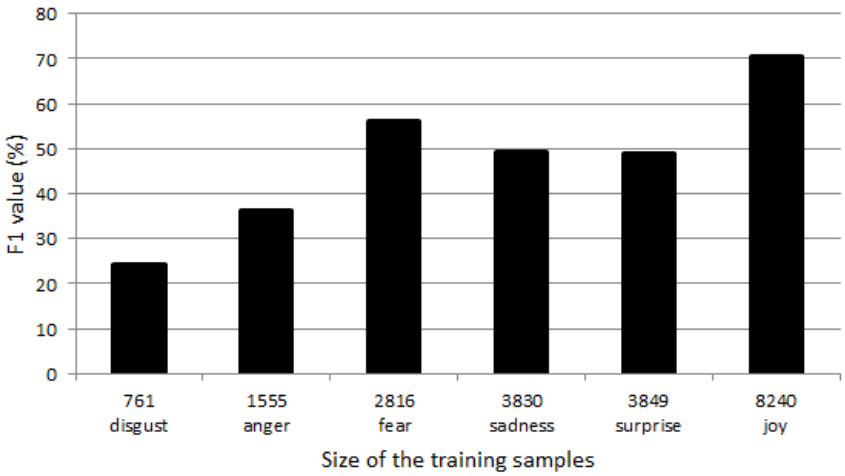

Figure 5.1: Effect of the size of training samples on the performance

the relation between the number of training samples provided for each of the emotions and their corresponding F1 values, sorted by the number of training samples. It seems that there exists a correlation between the size of the training set and the performance, such that, if the number of training samples increases, then the system achieves a better F1 value. This is inline with the findings of W. Wang et al. [47] who suggest that "*learning from large training data can play an important role in emotion identification*". The only exception in *TEC* is for the emotion *fear*, that in spite of fewer training samples than *sadness* and *surprise* achieves a better performance.

To evaluate our methods, we conduct experiments in three stages. All experiments are done in a 10-fold cross validation manner, to be comparable to Mohammad's results. The stages are elaborated in the following:

1. **Labeling procedure:**

   Similar to Mohammad, we train 6 binary SVM classifiers using unigrams and bi-grams as features; however, in the classification phase, we follow the method that

was explained in Chapter 4 Section 4.2, i.e. the core SVM model. Each classifier outputs a probability for a given test tweet and the emotion showing the maximum probability is the predicted label. This means that one, and only one, label is assigned to each sample. In this way, we keep all stages similar to what Mohammad does except for the labeling procedure. Nevertheless, it should be noted that due to memory limits, we require to confine both unigrams and bigrams to those that are repeated at least 3 times in the training set. The results of this experiment can be seen in Table 5.2 part *a*. Compared to the results of Table 5.1, it can be inferred that only changing the labeling procedure can have a remarkable influence on the performance, boosting it in this case by about 5%. While the recall increases significantly, the precision drops a little.

2. **Features:**

   In the second step, the impact and importance of the features is assessed. Here we use only unigrams for training SVM classifiers to evaluate how informative the bigrams are and if their avoidance has any effect. All other details are kept similar to the previous step. The results are shown in Table 5.2 part *b*. The difference between the F1 values of part *a* and *b* of this table shows the impact of using bigrams. This difference is only 0.38% meaning that adding bigrams to the set of features does not add much new information. In fact, it can even have negative impacts in terms of the running time of the algorithm, thus, we suggest ignoring bigrams, as it would produce a much faster system at the cost of a negligible drop of the F1 measure.

3. **Learning algorithm:**

   Finally, the impact of the learning algorithm is tested. We train 6 Naïve Bayes classifiers using the unigram features, that is the configuration of our core Naïve Bayes model, introduced in the previous chapter. All parts of the system are kept similar to the previous step, except that the SVM algorithm is replaced with the Naïve Bayes. The results of this experiment, shown in Table 5.2 part *c*, suggest that using the Naïve Bayes method instead of SVM boosts the performance by over 4%. Overall, by modifying the labeling procedure, pruning the feature set and using a better learning algorithm we can improve the F1 value from 40.98% by Moahammad to 49.30% which is a great achievement. More importantly, it indicates that the methods trained and tested on *CBET* can be used for other Twitter datasets with an acceptable outcome. It should be noted however, that even an F1 value of 49.3% is

Table 5.2: Results of 3 experiments on TEC

	Labeling procedure			Features			Learning algorithm		
Emotion	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
anger	27.07	39.82	32.11	26.69	42.37	32.72	30.37	45.22	36.29
fear	51.22	52.99	52.03	50.91	51.94	51.38	63.41	50.31	56.06
joy	72.17	58.66	64.71	71.59	58.15	64.16	71.99	69.08	70.49
sadness	48.10	39.95	43.59	47.08	39.24	42.79	47.00	51.71	49.21
surprise	51.06	49.37	50.17	49.79	49.40	49.52	62.60	40.28	48.94
disgust	13.65	40.61	20.39	14.52	38.48	21.04	17.08	42.67	24.29
ALL	**43.88**	**46.90**	**45.34**	**43.43**	**46.60**	**44.96**	**48.74**	**49.88**	**49.30**

not satisfactory enough, particularly for the emotions that are relevant to stress like anger, disgust, and thankfulness. In order to be used in sensitive real world applications, such as stress detection or suicide prevention systems, more work on emotion mining methods is indeed necessary.

5.2 ISEAR dataset

In this section, we test the methods that are developed for Twitter data on the *ISEAR* dataset, introduced in Section 2.4, that consists of formally written pieces of text. The motivation of doing this experiment is twofold: first, the proposed methods can be validated on a different domain other than the one that they were built specifically for; second, the results can be compared to others' in order to give a better understanding of the efficiency of our approaches.

Samples in *ISEAR* are basically paragraphs written by the crowd, describing a situation they have experienced an emotion. Each paragraph is labeled with one of 7 emotions of *anger, disgust, fear, guilt, joy, sadness,* and *shame* which all of them are covered in our set of 9 emotions except for *shame*. It contains about $7,600$ paragraphs and is almost balanced. The baseline for *ISEAR* can be a random classifier that assigns to each test sample one of the 7 emotions. Hence precision, recall, and F1 values are equal to $1/7 \sim 14.28\%$.

In the literature, there are some works done on emotion detection from *ISEAR* which were explained in Section 2.2. The work by D. T. Ho and T. H. Cao [41] uses a high-order Hidden Markov Model (HMM) to address the problem. They take into account only *anger, fear, joy,* and *sadness* emotions where *anger* covers both *anger* and *disgust*. The best reported value for F1, averaged over 4 emotions, is 35.3% for the configuration of a 2nd-order HMM with 45 states trained on $2/3$ of the samples and tested on the rest. S. M. Kim et al. [34] target the *ISEAR* dataset as well. They build 4 types of classifiers: discrete

Table 5.3: Results of running lexical and Naïve Bayes methods on ISEAR

	(a) lexical method			(b) Naïve Bayes method		
Emotion	**P**	**R**	**F1**	**P**	**R**	**F1**
anger	33.60	41.71	36.98	45.83	38.24	41.64
fear	57.29	56.41	56.74	65.68	64.28	64.92
joy	63.86	51.49	56.82	59.05	74.15	65.71
sadness	59.44	48.42	53.25	55.91	59.14	57.45
disgust	63.12	37.72	47.14	55.97	55.53	55.66
guilt	27.94	57.49	37.58	48.17	43.17	45.44
shame	58.95	30.26	39.80	51.25	50.50	50.85
ALL	**52.03**	**46.21**	**48.95**	**54.55**	**55.00**	**54.78**
5 emotions	**55.46**	**47.15**	**50.97**	**56.49**	**55.40**	**55.94**

classifiers with LSA, PLSA, and NMF dimension reduction methods and a dimensional classifier. Similar to [41], they also consider *anger + disgust, fear, joy,* and *sadness*. The reported F1 values averaged for all emotions are 22.77%, 26.95%, 16.55%, and 37.22% for each of the mentioned classifiers, respectively. Note that since they consider 5 emotions, a random classifier acting as a baseline has F1 value of $1/5 = 20\%$.

We test our lexical-based method and the best performing learning-based method, i.e. the core Naïve Bayes model on *ISEAR*. For more details one can refer to Chapter 4. Table 5.3 part *a* and *b* represent results of the lexical and Naïve Bayes approaches. F1 values of 48.95% and 54.78% for two models show a great achievement. To make the comparison fairer with previous works, we also consider only those 5 emotions suggested by them. The F1 values averaged over the 5 emotions for the lexical and the Naïve Bayes method are 50.97% and 55.94% which is significantly higher than both previous attempts. *Fear* and *joy* are the best predictable emotions while *anger* is the hardest among all. It seems that the ability of the classifiers to predict a specific emotion varies highly from one dataset to another. For instance, *sadness* is one of the toughest emotions to predict in *CBET* while it is predicted with the pretty good result in *ISEAR*.

In summary, our methods achieve good results on a dataset from Twitter and another one from formal texts. They could even outperform others' works on the two investigated datasets, which makes them a candidate for being used on other domains as well.

5.3 Meerkat analysis tool

So far in this chapter, we showed that our method is capable of predicting the emotion of texts in both informal and formal datasets. Hence, we decide to use our method for

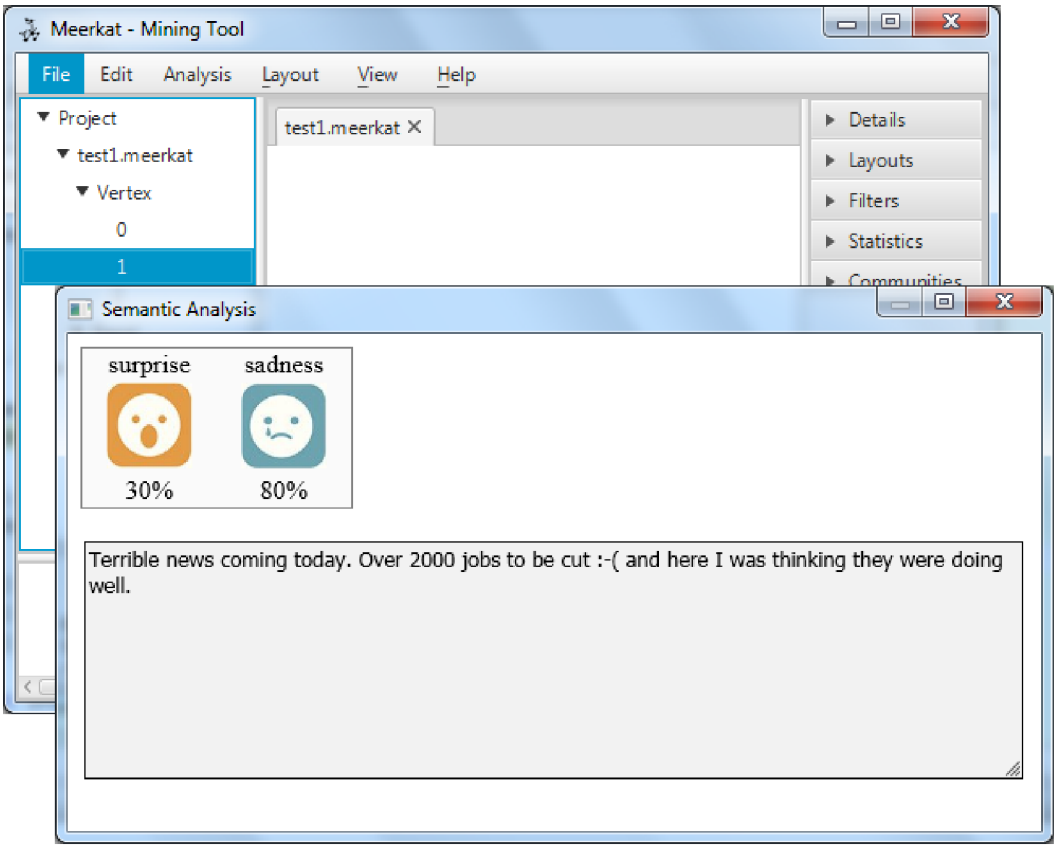

Figure 5.2: A snapshot of emotion classification tool in Meerkat

a multi-label and domain-independent emotion classification task, defined in a social network analysis (SNA) tool called *Meerkat*. Developed at the *Alberta Innovates Centre for Machine Learning (AICML)*[1], Meerkat offers a wide range of visualization and SNA analytic algorithms as well as a set of text analysis tools for datasets consisting a group of messages such as forums, discussions, etc. Our approach is embedded within the text analysis package of Meerkat. Given a single message, our emotion classification system is able to predict if it has no, one, or more emotions. In addition, the estimated certainty of the system for predicting each of the emotion(s) is shown to the user. For a group of messages, the average of certainty of emotions of all of its members is also shown.

A snapshot of the emotion detection module of the Meerkat application is shown in Figure 5.2. The selected message is predicted to show *surprise* with 30% confidence and *sadness* with 80% confidence. Figure 5.3 depicts the icons representing the 9 emotions.

[1]http://www.aicml.ca/

Figure 5.3: Icons representing the 9 emotions

5.4 Summary

In this chapter, another Twitter dataset and a dataset of formal documents, both emotionally labeled, were introduced. The main purpose was to explore the ability of our methods (that were previously tested only on *CBET*) to classify the emotion(s) conveyed by members of other textual corpora. By inspecting the performance results, fine-grained over individual emotions, we concluded that there is a direct relationship between the number of provided training samples for an emotion and the performance of the system over that emotion. Furthermore, results of experimenting our method on the two discussed datasets revealed that our system outperforms previous works on the similar task and is a strong candidate for being used in other emotionally concerned applications. Considering this fact, we integrated our system in a social network analysis (SNA) tool called Meerkat, that is designed for visualization and analysis of social networks as well as providing a set of text analysis tools.

Chapter 6

Conclusions and Future Work

In this work, we addressed the problem of text-based emotion classification which refers to a fine-grained classification of emotion(s) conveyed by a text into one (or more) of a set of predefined emotions. Personal notes, emails, news headlines, blogs, tales, novels, and chat messages are some types of text that can convey emotions. Particularly, popular social networking websites such as Twitter, Facebook, and MySpace are common places to share one's feelings. Emotion classification is an interesting topic in many disciplines such as neuroscience, cognitive sciences, psychology, and computer science and has many applications including e-learning systems, human-computer interaction, customer care services, and psychological cognition.

One of the contributions that we brought to this field was completion of a thorough survey of the previous research in emotion mining. Moreover, we introduced useful emotion related resources including lexicons and datasets.

We targeted Twitter messages in our study. Twitter is an online microblogging media that allows users to post short messages, called *tweet*, limited to 140 characters. Being short, informal, having misspellings, special symbols such as emoticons and emojis, short forms of words, and abbreviations are properties that discriminate them from normal texts and add to the complexity of the task.

To address the emotion classification problem, we first compiled a corpus of 27,000 emotional tweets, called *CBET*, that contains a balanced number of samples from 9 basic emotions: *anger, fear, disgust, joy, love, sadness, surprise, thankfulness,* and *guilt*. Next, we proposed a lexical-based method that basically evaluates the content of a message regarding the emotion(s) that its words or phrases have and a decision is made based on this information. In addition, several learning-based methods were suggested. They essentially try to learn patterns from a training set in which messages are labeled and then these patterns

are used to guess the label of some new messages that the algorithm has not seen before. Also, the effects of different feature selection methods, dimension reduction approaches, other configurations of classifiers, and various learning algorithms were investigated. Our methods showed promising results over *CBET*. Additionally, they were shown to be capable of doing multi-label classification and domain-independent performance such as being used for other Twitter and non-Twitter domains including *TEC* and *ISEAR*. Proposing these methods and their ability to outperform other methods experimented on these two domains in our another major contribution. Finally, our system was embedded into a social network analysis tool, called *Meerkat*.

6.1 Future Work

Emotion classification is a challenging task and our work can be expanded to address more of these challenges. First, analyzing the effect of other types of features including punctuation marks, words containing all capital letters (e.g. HAPPY), words with character flooding (e.g. happyyyy), part-of-speech tags, and negation words may lead to new results. There are also some other textual features, exploited in other applications, such as the concept of "Coh-Metrix" [76] used in the assessment of cohesion of a document, which is a popular task in the field of computational linguistics. Coh-Metrix introduces some linguistic and discourse measurements of a text including descriptiveness, lexical diversity, situation model, syntactic complexity, and so on. The effect of each of these groups of features can be experimented on emotion classification problem. Second, applying more accurate natural language processing techniques, such as robust tokenization and stemming methods or reliable ways to prune non-discriminant words could improve the quality of the input text. These are just a few instances out of many possible improvement ideas.

Throughout this work, we assumed the emotion label of the training samples to be provided to the algorithm which might not always be true. In fact, in most real world problems, available data are not labeled, so it is worth trying to classify texts in an unsupervised manner. Moreover, in some applications it is important to detect the event, person, or subject that causes a particular emotion. This is called emotion cause detection and is an emerging area of research within the emotion mining field. Finally, one should note that the changing and evolving nature of each and every language is a main issue. As the language or slang used in messages evolves, the system developed for recognizing emotions in texts should adapt to the changes.

Bibliography

[1] Y.-H. Yang and H. H. Chen, "Machine recognition of music emotion: A review," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, p. 40, 2012.

[2] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[3] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 1, pp. 39–58, 2009.

[4] A. Kleinsmith and N. Bianchi-Berthouze, "Affective body expression perception and recognition: A survey," *Affective Computing, IEEE Transactions on*, vol. 4, no. 1, pp. 15–33, 2013.

[5] S. K. D'mello and J. Kory, "A review and meta-analysis of multimodal affect detection systems," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, p. 43, 2015.

[6] N. Gupta, M. Gilbert, and G. D. Fabbrizio, "Emotion detection in email customer care," *Computational Intelligence*, vol. 29, no. 3, pp. 489–505, 2013.

[7] S. Voeffray, "Emotion-sensitive human-computer interaction (hci): State of the art-seminar paper," *Emotion Recognition. p1-4*, 2011.

[8] F. Rangel and P. Rosso, "On the impact of emotions on author profiling," *Information Processing & Management*, 2015.

[9] M. De Choudhury, M. Gamon, S. Counts, and E. Horvitz, "Predicting depression via social media.," in *ICWSM*, 2013.

[10] K. Luyckx, F. Vaassen, C. Peersman, and W. Daelemans, "Fine-grained emotion detection in suicide notes: A thresholding approach to multi-label classification," *Biomedical informatics insights*, vol. 5, no. Suppl 1, p. 61, 2012.

[11] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan, "Analysis of emotion recognition using facial expressions, speech and multimodal information," in *Proceedings of the 6th international conference on Multimodal interfaces*, pp. 205–211, ACM, 2004.

[12] A. Wieczorkowska, P. Synak, and Z. W. Raś, "Multi-label classification of emotions in music," in *Intelligent Information Processing and Web Mining*, pp. 307–315, Springer, 2006.

[13] Y. Zhang, L. Shang, and X. Jia, "Sentiment analysis on microblogging by integrating text and image features," in *Advances in Knowledge Discovery and Data Mining*, pp. 52–63, Springer, 2015.

[14] E. Fox, *Emotion science cognitive and neuroscientific approaches to understanding human emotions*. Palgrave Macmillan, 2008.

[15] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, vol. 6, no. 3, pp. 169–200, 1992.

[16] R. Plutchik and H. Kellerman, *Emotion: theory, research and experience*. Academic press New York, NY, 1986.

[17] P. Shaver, J. Schwartz, D. Kirson, and C. O'connor, "Emotion knowledge: further exploration of a prototype approach.," *Journal of personality and social psychology*, vol. 52, no. 6, p. 1061, 1987.

[18] H. Lövheim, "A new three-dimensional model for emotions and monoamine neurotransmitters," *Medical hypotheses*, vol. 78, no. 2, pp. 341–348, 2012.

[19] C. O. Alm and R. Sproat, "Emotional sequencing and development in fairy tales," in *Affective Computing and Intelligent Interaction*, pp. 668–674, Springer, 2005.

[20] P. Ekman, W. V. Friesen, and P. Ellsworth, *Emotion in the human face: Guidelines for research and an integration of findings*. Pergamon Press, 1972.

[21] J. B. Walther, "Interpersonal effects in computer-mediated interaction a relational perspective," *Communication research*, vol. 19, no. 1, pp. 52–90, 1992.

[22] J. B. Walther, T. Loh, and L. Granka, "Let me count the ways the interchange of verbal and nonverbal cues in computer-mediated and face-to-face affinity," *Journal of language and social psychology*, vol. 24, no. 1, pp. 36–65, 2005.

[23] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Textual affect sensing for sociable and expressive online communication," in *Affective Computing and Intelligent Interaction*, pp. 218–229, Springer, 2007.

[24] Y. Rao, Q. Li, L. Wenyin, Q. Wu, and X. Quan, "Affective topic model for social emotion detection," *Neural Networks*, vol. 58, pp. 29–37, 2014.

[25] G. Mishne and M. De Rijke, "Capturing global mood levels using blog posts.," in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pp. 145–152, 2006.

[26] J. Lei, Y. Rao, Q. Li, X. Quan, and L. Wenyin, "Towards building a social emotion detection system for online news," *Future Generation Computer Systems*, vol. 37, pp. 438–448, 2014.

[27] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[28] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[29] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015.

[30] E.-C. Kao, C.-C. Liu, T.-H. Yang, C.-T. Hsieh, and V.-W. Soo, "Towards text-based emotion detection a survey and possible improvements," in *International Conference on Information Management and Engineering (ICIME)*, pp. 70–74, IEEE, 2009.

[31] M. C Jain and V. Y Kulkarni, "Texemo: Conveying emotion from text-the study," *International Journal of Computer Applications*, vol. 86, no. 4, pp. 43–49, 2014.

[32] K. Gao, H. Xu, and J. Wang, "Emotion cause detection for chinese micro-blogs based on ecocc model," in *Advances in Knowledge Discovery and Data Mining*, pp. 3–14, Springer, 2015.

[33] J. T. Hancock, C. Landrigan, and C. Silver, "Expressing emotion in text-based communication," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 929–932, ACM, 2007.

[34] S. M. Kim, A. Valitutti, and R. A. Calvo, "Evaluation of unsupervised emotion models to textual affect recognition," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pp. 62–70, Association for Computational Linguistics, 2010.

[35] C. O. Alm, D. Roth, and R. 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*, pp. 579–586, Association for Computational Linguistics, 2005.

[36] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Compositionality principle in recognition of fine-grained emotions from text," in *ICWSM*, 2009.

[37] F.-R. Chaumartin, "Upar7: A knowledge-based system for headline sentiment tagging," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 422–425, Association for Computational Linguistics, 2007.

[38] C. Strapparava and R. Mihalcea, "Learning to identify emotions in text," in *Proceedings of the 2008 ACM symposium on Applied computing*, pp. 1556–1560, ACM, 2008.

[39] T. Danisman and A. Alpkocak, "Feeler: Emotion classification of text using vector space model," in *AISB 2008 Convention Communication, Interaction and Social Intelligence*, vol. 1, p. 53, 2008.

[40] R. E. Schapire, "A brief introduction to boosting," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 99, pp. 1401–1406, 1999.

[41] D. T. Ho and T. H. Cao, "A high-order hidden markov model for emotion detection from textual data," in *Knowledge Management and Acquisition for Intelligent Systems*, pp. 94–105, Springer, 2012.

[42] G. Mishne, "Experiments with mood classification in blog posts," in *Proceedings of ACM SIGIR 2005 workshop on stylistic analysis of text for information access*, vol. 19, Citeseer, 2005.

[43] P. K. Bhowmick, "Reader perspective emotion analysis in text through ensemble based multi-label classification framework," *Computer and Information Science*, vol. 2, no. 4, p. 64, 2009.

[44] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *Proceedings of the 17th international conference on Computational linguistics*, vol. 1, pp. 86–90, Association for Computational Linguistics, 1998.

[45] N. C. for Biomedical Computing, "2011 nlp shared task." `https://www.i2b2.org/NLP/Coreference/Call.php`, 2011. [Online; accessed 19-Apr.-2015].

[46] J. Bollen, H. Mao, and A. Pepe, "Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena," in *ICWSM*, 2011.

[47] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, "Harnessing twitter "big data" for automatic emotion identification," in *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 International Confernece on Social Computing (SocialCom)*, pp. 587–592, IEEE, 2012.

[48] M. Hasan, E. Agu, and E. Rundensteiner, "Using hashtags as labels for supervised learning of emotions in twitter messages," *health informatics workshop (HI-KDD)*, 2014.

[49] K. Roberts, M. A. Roach, J. Johnson, J. Guthrie, and S. M. Harabagiu, "Empatweet: Annotating and detecting emotions on twitter," in *LREC*, pp. 3806–3813, 2012.

[50] S. M. Mohammad, "#emotional tweets," in *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pp. 246–255, Association for Computational Linguistics, 2012.

[51] W. Li and H. Xu, "Text-based emotion classification using emotion cause extraction," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1742–1749, 2014.

[52] C. Strapparava and A. Valitutti, "Wordnet affect: an affective extension of wordnet.," in *LREC*, vol. 4, pp. 1083–1086, 2004.

[53] D. A. Medler, A. Arnoldussen, J. Binder, and M. Seidenberg, "The wisconsin perceptual attribute ratings database." `http://www.neuro.mcw.edu/ratings/`, 2005.

[54] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, p. 2001, 2001.

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

[56] P. Katz, M. Singleton, and R. Wicentowski, "Swat-mp: the semeval-2007 systems for task 5 and task 14," in *Proceedings of the 4th international workshop on semantic evaluations*, pp. 308–313, Association for Computational Linguistics, 2007.

[57] E. C. O. Alm, *Affect in text and speech*. ProQuest, 2008.

[58] K. R. Scherer and H. G. Wallbott, "Evidence for universality and cultural variation of differential emotion response patterning.," *Journal of personality and social psychology*, vol. 66, no. 2, p. 310, 1994.

[59] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 70–74, Association for Computational Linguistics, 2007.

[60] P. Corcoran, "Emotional framing in australian journalism," in *Australian & New Zealand Communication Association International Conference, Adelaide, Australia, ANZCA*, 2006.

[61] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing, "A latent variable model for geographic lexical variation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1277–1287, Association for Computational Linguistics, 2010.

[62] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, p. 12, 2009.

[63] W. G. Parrott, *Emotions in social psychology: Essential readings*. Psychology Press, 2001.

[64] C. E. Izard, *The psychology of emotions*. Springer Science & Business Media, 1991.

[65] N. Shuyo, "Language detection library for java." `http://code.google.com/p/language-detection/`, 2010.

[66] J. Kun, "Word segmentation, or makingsenseofthis." `http://jeremykun.com/2012/01/15/word-segmentation/`, 2012. [Online; accessed 15-Nov.-2014].

[67] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, 2014.

[68] H. Binali, C. Wu, and V. Potdar, "Computational approaches for emotion detection in text," in *4th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, pp. 172–177, IEEE, 2010.

[69] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[70] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *Proceedings of the fourth international conference on knowledge discovery and data mining*, 1998.

[71] 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*, vol. 11, no. 1, pp. 10–18, 2009.

[72] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proceedings of the 17th international conference on World Wide Web*, pp. 91–100, ACM, 2008.

[73] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[74] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[75] J. C. Platt *et al.*, "Using analytic qp and sparseness to speed training of support vector machines," *Advances in neural information processing systems*, pp. 557–563, 1999.

[76] D. S. McNamara, A. C. Graesser, P. M. McCarthy, and Z. Cai, *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge University Press, 2014.