Automatic Emotion Detection Using Twitter Data

Ali Moayedi Azarpour

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
June 2019
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Acting Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Hadi Işık Aybay
Acting Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Ekrem Varoğlu
Supervisor

Examining Committee

1. Prof. Dr. Hakan Altınçay

2. Prof. Dr. Ekrem Varoğlu

3. Assoc. Prof. Dr. Hüseyin Öztoprak

ABSTRACT

Emotions play an important role in human communication. People express their emotions in daily life and understanding emotions enrich interactions. Understanding emotions has been a topic of physiological studies for decades. In recent years, emotions in interactions of humans with computers have become an active topic of research as they can affect users' concentration and decision making skills. Except trivial ways of expressing emotions such as language skills, changes in tone of voice, and body or facial gestures, other ways such as writing short texts has become more prevalent due to the increasing influence of social media. Affect computing is the science of studying people and their emotions at the time of interaction with computers with the ultimate goal of producing systems that are able to detect and understand human emotions and their intensity.

Many studies in detection of emotions from a textual context such as novels and newspaper headlines have been conducted. However, due to the increasing interests toward social media in recent years, Twitter as the fastest growing social networking system, has received more attention as a valuable free source of texts.

In this thesis, the aim is to generate an automated system that classifies tweets based on the experienced intensity level of emotions for four different emotions: anger, joy, fear, and sadness. A linear SVM model is chosen as the classification algorithm. Different sources of feature sets are introduced and used such as affect lexicons, word2vec models, query terms, and tf-idf scoring. Furthermore, in an attempt to increase classification performance, wrapper based feature subset selection

algorithms including Forward Selection (FS), Simplified Forward Selection (SFS), Random Forward Selection (RFS), and Backward Selection (BS) are applied on the feature sets. Similar approaches have also been applied for classifier selection. In classifier combination, majority voting method is used to combine scores from different classifiers. Both simple and weighted voting schemes utilized and the results are compared. Results of this study suggest that recommended subsets of feature sets or classifiers give slightly better performances. However, it is shown that different subsets work better for classifying emotion intensities for different emotions.

Keywords: Tweet Classification, Emotions, Support Vector Machines, Lexicons, Feature Selection, Classifier Selection, Machine Learning, Text Mining.

ÖZ

Duygular insan iletişiminde önemli bir rol oynamaktadır. İnsanlar günlük yaşamda duygularını ifade eder ve duyguları anlamak etkileşimleri zenginleştirir. Duyguların anlaşılması on yıllardır fizyolojik çalışmaların bir konusu olmuştur. Son yıllarda, insanların bilgisayarlarla etkileşimlerindeki duygular, kullanıcıların konsantrasyon ve karar alma becerilerini etkileyebilecekleri için aktif bir araştırma konusu haline gelmiştir. Dil becerileri, ses tonundaki değişiklikler ve vücut veya yüz hareketleri gibi duyguları ifade etmenin basit yolları dışında, kısa metinler yazmak gibi diğer yollar sosyal medyanın artan etkisine bağlı olarak daha yaygın hale gelmiştir. Etki hesaplama, insan duygularını ve yoğunluğunu tespit edebilen ve anlayabilen sistemler üretmek amacıyla bilgisayarlarla etkileşim sırasında insanları ve duygularını inceleyen bir bilim dalıdır.

Romanlar ve gazete manşetleri gibi metinlerden duygularının tespiti konusunda birçok çalışma yapılmıştır. Ancak, son yıllarda sosyal medyaya olan ilginin artması nedeniyle, en hızlı büyüyen sosyal ağ sistemi olan Twitter, değerli bir serbest metin kaynağı olarak daha fazla dikkat çekmiştir.

Bu tezde, dört farklı duygu için, öfke, sevinç, korku ve üzüntü, deneyimlerin duygu yoğunluğu seviyesine göre tweetleri sınıflandıran otomatik bir sistem geliştirilmiştir. Sınıflandırma algoritması olarak doğrusal Destek Vektör Makineleri (SVM) seçilmiştir. Öznitelik kümesi olarak word2vec modelleri, sorgu terimleri ve tf-idf gibi farklı öznitelikler seti tanıtılmış ve kullanılmıştır. Ayrıca, sınıflandırma performansını arttırmak için, İleri Seçim (FS), Basitleştirilmiş İleri Seçimi (SFS),

Rastgele İleri Seçimi (RFS) ve Geri Seçimi (BS) yöntemlerini içeren sarmalayıcı tabanlı öznitelik alt kümesi seçim algoritmaları uygulanmıştır. Buna ek olarak, sınıflandırıcı seçimi için de benzer yaklaşımlar uygulanmıştır. Sınıflandırıcı birleştirme yöntemi olarak, farklı sınıflandırıcılardan alınan puanları birleştirmek için çoğunluk oylama yöntemi kullanmıştır. Çoğunluk oylama yönteminde hem basit hem de ağırlıklı oylama düzenleri kullanılmış ve sonuçlar karşılaştırılmıştır. Bu çalışmanın sonuçları, önerilen öznitelik alt kümelerinin veya sınıflandırıcı alt kümelerinin biraz daha iyi performans verdiğini göstermektedir. Bununla birlikte, farklı alt kümelerin, farklı duygular için duygu yoğunluğunu sınıflandırmak için daha iyi çalıştığı gösterilmiştir.

Anahtar Kelimeler: Tweet Sınıflandırma, Duygular, Destek Vektör Makineleri, Sözlükler, Öznitelik Seçimi, Sınıflandırıcı Seçimi, Makine Öğrenmesi, Metin Madenciliği.

Dedicated to

my beloved parents, brother, and sister-in-law

for their love, endless support, and encouragement . . .

ACKNOWLEDGMENTS

I hope everyone that is reading this is having a really good day. And if you are not, just know that in every new minute that passes you have an opportunity to change that.

Gillian Anderson

I would like to express my deepest appreciation to my thesis advisor Prof. Dr. Ekrem Varoğlu whose contribution in stimulating suggestions and encouragement, helped me to coordinate my thesis. I would also like to acknowledge with much appreciation the crucial role of Prof. Dr. Hakan Altınçay and Assoc. Prof. Dr. Hüseyin Öztoprak for their careful review and useful comments on my work.

Finally, I must express my very profound gratitude to my parents, brother, and sister in law for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BS	Backward Selection
CBoW	Continuous Bag of Words
FN	False Negative
FP	False Positive
FS	Forward Selection
IDF	Inverse Document Frequency
SB	Single Best
SFS	Simplified Forward Selection
SVM	Support Vector Machine
TF	Term Frequency
TN	True Negative
TP	True Positive
WSD	Word Sense Disambiguation

Chapter 1

INTRODUCTION

1.1 Background

Emotions undoubtedly play an important role in human life and affect decisions and relations among people more or less. Emotions as a psychology and neural science have been subject of studies through decades. However, in recent years their automatic detection has been a topic of research in areas such as artificial intelligence since recognition of emotions and their influence can enhance productivity and effectiveness of working with computers. Education systems, website customization and games are just a few samples of intelligent systems wide range of usage.

Study of emotions and computations in this domain was firstly introduced in 1995 as "Affective Computing" [1]. Scientists and in some cases businesses benefit from various automatic classification techniques to correctly detect emotions. These techniques are mainly similar in core and have developed in three main phases of gathering train data, extracting characteristics (preferably discriminative ones) and constructing a model that will be tested later on a new set of data [2]. The developed model is supposed to be able to correctly classify previously unseen data.

In reality due to the various ways that emotions are expressed such as changes in breathing rhythm and heart rate, changes in facial form or the ways of uttering words,

extracting features and developing models is not simple [3]. Therefore, selecting appropriate technique for model construction is directly related to the type of data. As an example, deciding on the emotion inferred from a photo needs image processing techniques for feature collection. Among different means of communication used to express emotions (i.e. speaking, facial expressions, etc.), writing has received more attention in recent years as telecommunication systems are getting more prevalent. Every day the number of people conveying emotions and feelings by writing and sharing through internet increases. However, detecting emotions from texts is not similar and has some unique challenges which may be different then detecting them from images or voice. Texts usually contain miss-spelled terms, slangs, abbreviations, emoticons, and might be in different languages. Expressed emotion can also differ by changes in voice, body gestures, or facial expressions [4]. As a result, steps of preprocessing (e.g. tokenization, lemmatization, parsing and part-of-speech tagging) are needed [3].

Nevertheless, valuable sources of texts are freely available. Different studies have already focused on newspapers and novels. However, Twitter as the fastest growing social networking service in comparison with other platforms has received more attention [5]. Furthermore, posts on Twitter, so called tweets, have limited length and due to this limitation users have to briefly express their thoughts. Researchers also revealed that tweets often state emotions of their authors [6]. Thus Twitter has become a noteworthy data source for emotion detection studies. Development of intelligent machines regarding human language was introduced first in 1950 as Natural Language Processing (NLP). NLP was formed with the idea of combining

computer science and artificial intelligence (AI) as a mean of interaction between machines and humans language [7]. In detection of emotions from texts, essentially two methods of sentiment analysis and emotion analysis are used. Sentiment analysis as a sub topic of Natural Language Processing (NLP), detects positivity or negativity of feelings regarding an input text, while emotion analysis decides on the emotion type (e.g. joy) [4].

After the data is preprocessed, the next step is to change data into some concepts and relations. Concepts or so called features and attributes in machine learning context, translate data into usable information for learning algorithms and based on the type of data different feature extraction techniques can be applied [2].

Term Frequency (TF), Inverse Document Frequency (IDF), and their combination are examples of extracted features from a textual corpus. Lexicons, i.e. dictionaries of terms-scores, are also commonly practiced in converting texts to vectors of scores [8, 9, 10, 11].

Constructed feature sets are then fed into machine learning algorithms. Machine learning algorithms are a set of data analyzing techniques based on the assumption that machines can learn as humans do and the final purpose is to automate analytical model building [12]. The very basic form of learning is memorization when machines memorize a set of rules. Yet, memorization cannot help since the important third step of developing an automated classifier system, i.e. ability to classify unseen data, is missed. Developed models should be able of generalizing their learning to new instances [13].

Machine learning algorithms are divided into four categories of supervised, unsupervised, semi-supervised, and reinforcement learning. Deciding on a proper algorithm is based on the task and the type of data used. Support Vector Machines (SVM), applied classifier in this study, is one of the well-known supervised classification algorithms. This algorithm classifies input data by looking at attributes and decides on a decision boundary that separates classes in the feature space with the largest margin from other classes.

1.2 Scope of Study

This thesis is inspired by the SemEval 2017 international workshop on semantic evaluation (a competition on automatic emotion detection) [9]. However, in contrast with the competition task which reports levels of emotions in real-valued scores, here classification of tweets into four discrete levels of emotion intensity is discussed.

Throughout this study, a corpus of tweets is used that was released and is publicly accessible on the competition web-page [14]. The data set is basically divided into three sets: First two sets, train and development sets, are employed for model development and the third part, test set, is saved for later model evaluation.

The data is first preprocessed by tokenization and afterwards feature extraction techniques such as lexicon and tf-idf scoring and word2vec algorithm is applied to convert textual data into a set of characteristics (vector of features) for model learning. Support Vector Machines (SVM) as the learning algorithm is applied to construct classification models. The trained models are validated on the

development data set. Models validation helps to optimize parameters and since this is a classification task, precision, recall, and micro and macro F-scores are observed and efforts are carried out to increase them. In total 19 feature sets are considered in this study and their effectiveness in emotion detection is checked. In addition as a novel work, wrapper based feature and classifier selection techniques are employed to investigate the effectiveness of the subsets of features and classifiers, respectively. In particular forward (random, simplified, and greedy) and backward selection methods are applied and their performances are compared to the single best and combination of all features and classifiers.

1.3 Outline

This thesis starts with a very brief review of basic concepts, ideas, and techniques to form a general overview of automated emotion detection techniques in mind. The rest of this thesis is organized as follows. In Chapter 2, basic concepts and definitions such as emotions, sentiments, their differences and lexicons are discussed and a wide variety of available lexicons are introduced. Reasons of tweets' popularity and extraction methods for data set construction is explained in detail. Machine learning and classification algorithms along with feature selection techniques are also discussed. Chapter 3 starts with a quick introduction to the SemEval 2017 shared task and continues with the novel study in this thesis. Techniques used, preprocessing, developed models, and all basic systems are introduced through this chapter. Chapter 4 discusses the results obtained and efforts are focused on improving performance. Finally, Chapter 5 summarizes the results and presents a conclusion of all results obtained.

Chapter 2

BASICS AND CONCEPTS

2.1 Literature Survey

Holzman and Pottenger in 2003 studied chat messages and annotated 1201 samples according to Ekman's six emotions plus two additional classes, irony and neutral [15]. Alm, Roth, and Sproat (2005) focused on annotation of 22 Grim fairy tales based on Ekman's set [16]. Brooks et al. worked on annotation of 27344 chat messages according to 40 affect classes inferred from Plutchik's emotion set [17]. Mohammad used Twitter API to retrieve tweets including hashtag terms corresponding to Plutchik's emotion set [18]. His work revealed that hashtag terms can work as good as labels for tweets and are in a comparable level with explicitly annotated emotions [19].

Many other works were carried out on different smaller sets of emotions. The ISEAR project focused on supervised machine learning techniques using developed dataset from participation of 3000 students that were asked to report situations of experiencin joy, fear, anger, sadness, disgust, shame, and guilt [19]. Pearl and Steyvers (2010) worked on detection of politeness, rudeness, embarrassment, formality, persuasion, deception, and disbelief by developing online Games with a Purpose (GWAP) [20]. There were studies too, on emotion detection in other languages except English. Wang's studies (2014) were focused on annotation of Chinese news and blog posts according to Ekman's emotions [21].

Sun, Quan, Kang, Zhang, and Ren (2014) worked on detection of emotions in Japanese customers' reviews [22].

In addition to studies on different emotions and data sets, many works are engaged in better automatic emotion detection systems. Shivhare, Shiv Naresh, and Saritha Khethawat (2012) developed an automated system that was based on keyword spotting technique, learning-based, and hybrid methods [23]. Studies of Shaheen, Shadi, Wassim El-Hajj, Hazem Hajj, and Shady Elbassuoni (2014) orientated toward deployment of semantic and syntactic analyses in training model [24]. They also used WordNet and ConceptNet (i.e. a lexicon) for rule setting. Their study proved that between-terms' relation consideration results in higher accuracy than simple score assignments. Tilakraj et al. developed a system to handle negative sentences with positive words [25]. Agrawal, Ameeta, and Aijun (2012) made a comparitive study on context-based unsupervised approach against context-free technique in emotion detection from text. They found that context-based methods always outperform context-free methods [26].

2.2 SemEval Workshops

Since 1998 a series of evaluation competitions named as SemEval (Semantic Evaluation) started to explore the nature of meanings in language and assess computational semantic analysis systems. These competitions, which were initially held under the name of SensEval, focused on the evaluation of the quality of Word SenseDisambiguation (WSD) algorithms.However, since 2006 with a change in the primary goal, organizers looked for replication of human cognitive processing by the use of computer systems.

Semantic analysis commonly refers to the task of automated detection of valence or polarity of a text, where valence shows positive, negative, or neutral inferred emotion [19]. More specifically, the task tries to determine one's attitude towards a topic. Since attitude is categorized by some authors under a wider class, called feeling, sentiment analysis can be considered as the task of automatic feeling detection. Nonetheless, automatic feeling detection is considered as a challenging task for different reasons.

One of the challenges is due to the variety of emotions a word can convey in different contexts. For example, the word "close" can convey senses of "shutting", "blocking", or "ending". The next challenge is tone of reading texts. Emotions are generally conveyed through the way the text is uttered by changes in tone, pitch, or emphasis. Furthermore, texts can express feelings of the speaker without implicitly or explicitly stating them. Questions that are informally asked in declarative forms are examples of this issue. Challenges can even go further to the body language and facial expression that are commonly used to convey emotions, though, are not present in written texts. The other remarkable subject in automatic detection of feelings is written texts rich in irony, sarcasm, misspellings, and creatively-spelled words. Such texts convey emotions indirectly and detection of emotions in them requires high level of intelligence and understanding of the context. "The teacher fails the test", or "lov u mom" are examples of sentences with irony and creatively-spelled words. In addition to the discussed points, studies have revealed that detection of emotions even for humans is a difficult task. Annotators, when they are asked to decide on the inferred emotion from a tweet, show low levels of

agreement. This is considered as inconsistency and is addressed in MaxDiff scoring technique. Desire for larger data sets is another issue when it comes to developing models and training systems. Besides the challenges mentioned, different reactions to same utterances is an area of research which is not explored much[1].

In respect to sentiment analysis and emotion detection competition, in 2017 for the first time a shared task on emotion intensity detection was held under the title of "WASSA-2017 Shared Task on Emotion Intensity", with the aim of determining felt level of emotion by speaker [9]. Intensities are expected to be real values in the range of 0 to 1, while 1 shows the highest level of experienced emotion and 0 the lowest. Four common emotions such as anger, joy, fear, and sadness were proposed in the competition. The best team among the twenty-two participant teams was Prayas and achieved the best performance with Pearson correlation 0.747 on the Gold (test) set [11]. The competition is accessible on the CodeLab website.

2.3 Emotions

There is a wide range of definitions for emotion. Kleinginna and Kleinginna in 1981 listed 92 different definitions of emotion plus their own [27, 28]. *"Sudden trouble, transient agitation caused by an acute experience of fear, surprise, joy, etc."* (Larousse Dictionary, 1990) or *"mental feeling or affection (e.g., pain, desire, hope, etc.) as distinct from cognitions or volitions"* (Oxford English Dictionary, 1987) are two of many definitions for emotion. Emotions can be defined on the basis of time as well [29]. In this sense it is *"a reaction to stimuli that lasts for seconds or minutes"* [29]. Accordingly, mood and personality are defined as an emotional state that lasts

1 People on the opposite sides, for example of a match, can have different feeling on same sentences.

for hours, and an inclination to feel certain emotions, respectively. Therefore, emotional state is known as current state of a person irrespective of its origin (stimuli, mood, or personality) [29].

In spite of the beliefs in relations of emotions to physiological processes, there is not agreement on a basic set of emotions. Ekman considered joy, sadness, fear, anger, disgust, and surprise as six basic emotions [30] while Plutchik considered trust and anticipation in addition to Ekman's six basic emotions (Figure 2.1) [31, 32]. Parrot [33], Frijda [34] and others introduced different sets of basic emotions. Although in detection of emotions from texts, as will be discussed later, labeled sets of training data are needed for model construction, developing data sets with thousands of terms which are manually annotated is both expensive in time and cost. Consequently, consideration of a small set of emotions keeps expenses low as a positive aspect while, unfavorably, there will be fewer resources for non-basic emotions. Hence, there are many studies on different data sets with different sets of emotions.

2.4 Sentiment

Emotion, which is roughly defined as a mental feeling, along with opinion describes a private state known as sentiment [35]. Pang and Lee [36] define sentiment as an opinion which reflects ones feelings, and loosely it is considered as a positive or negative opinion [36, 37, 38, 39]. Sentiment is extremely context dependent and is appertaining to an individual [35]. Sentiment analyses which is alternatively called subjectivity analysis, opinion mining or affective computing, studies "linguistic expressions of private states in context" [40].

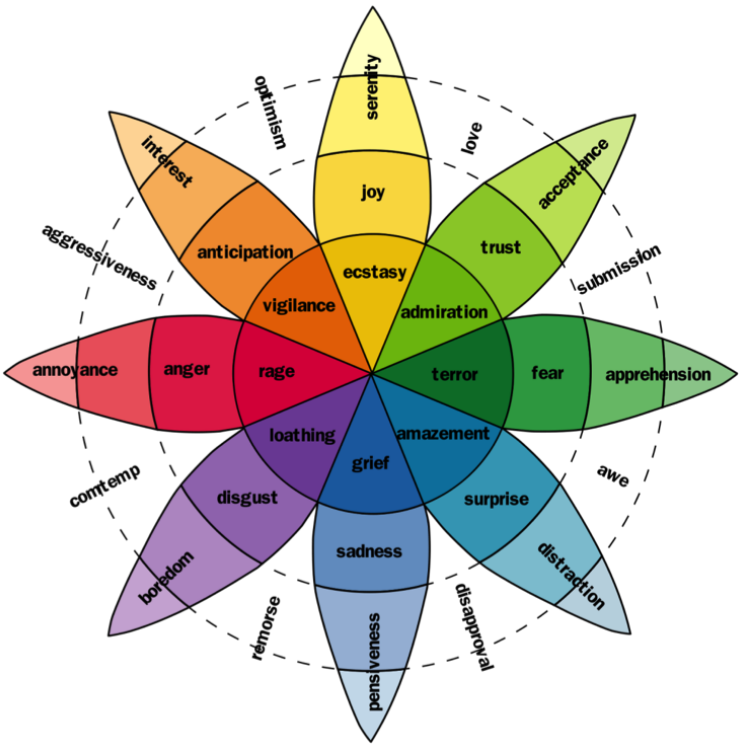


Figure 2.1: Plutchik's wheel of emotions

In sentiment analysis the effective issues are [41]:

- the way opinions are expressed, i.e. either explicit or implicit,
- the target of discussion (e.g. fear interprets differently when the target is a movie unlike an event),
- effect of the author on the context.

Sentiment analysis and automatic detection of emotions take place at different levels of textual chunks such as words, phrases, sentences, documents, tweets, comments, and reviews on different data sets. In its finest level, the word level, some terms convey valence as a part of their meaning (e.g. good, bad, nice), while others have strong associations with positive or negative valences (e.g. death or party). There are some words as well that are not attached to any of the positive or negative emotions and are considered as neutral. However, the boundary between positive and neutral or

negative and neutral valences is sometimes fuzzy. Similar to valence, words convey emotions directly (e.g. anger) or belong to an emotion (e.g. fight shows anger).

2.5 Lexicons

Inferring emotions and senses through verbal communication is almost certain and explicit. However, in the absence of voice, writing is one of human's alternative communication means and understanding of emotions through text is not as explicit. Prior researches demonstrated that readers activate mental representations of a character's emotional state while reading [42, 43]. This effect has been shown by inquiring participants to infer the emotional state of a character based on the description of the text (i.e. emotion inferences). Studies have also revealed that if readers are provided with longer texts that convey sufficient information (e.g. stories), make more specific emotional inferences [44]. However, in short contexts readers infer a more general feeling composed of different emotional components shared by several emotional terms. Therefore, a list of term-sentiment pairs is needed to be developed manually for later use by sentiment analysis systems as prior knowledge. Undoubtedly, these lists in comparison with the number of words and phrases in a language are limited and small. Hence, development of an automatic annotation system is of interest. Before continuing into details of automatic systems, developing such a list (known as a lexicon) requires a brief overview on what lexicons are and how they are manually constructed.

Lexicons, similar to dictionaries, are collection of words of a language [45], with scores or labels instead of meanings. Lexicons provide us with a list of words and their associated emotions or valences. Valence association lexicons are dictionary-

liked collections with word-valence pairs (e.g. shout-negative), on the other hand, affect association lexicons hold term-emotion pairs that are usually developed for a predefined emotion (e.g. shout-anger). Furthermore, in affect lexicons a term might be associated to more than one emotion and can have more than one entry.

Creation of lexicons can be done manually in a limited size or automatically with hundreds of thousands of records. Often automatically developed lexicons include real-valued scores for term-sentiment pairs. Developed lexicons in the word-level are used in sentence-level valence classification. In this level, sentences, a collection of words, are labeled with positive, negative, or neutral tags. However, valence of a sentence is not simply the summation of its terms valences. Thus, machines use learning techniques to decide on the valence based on a set of extracted features. Same techniques are used to detect emotions in sentences and label them as joy, fear, anger, or sadness; although, fewer attempts are done in this area.

Osgood, Suci, and Tannenbaum (1957) in their book, *Measurement of Meaning*, made the first study in this area and their developed lexicon determined the position of each term within several semantic dimension [46]. The General Inquirer (GI) [47] and Multi-Perspective Question Answering (MPQA) subjectivity lexicon [48] are two more examples of early lexicons. GI, a list of 3600 words, covers 1500 entries from Osgood list. MPQA similarly contains more than 8000 words from both GI and other resources in which terms are labeled with valences. Affective Norms for English Words (ANEW) by Bradley and Lang is another lexicon covering 1034 English words along with their corresponding valence, arousal and dominance [49].

Nielsen [50] introduced AFINN lexicon including 2477 English words with their valence rating from -5 (most negative) to 5 (most positive) in discrete values.

In a conducted comparative study on customer reviews in 2006, ordering relation between two sets of entities with respect to some shared featured were studied [51]. The main tasks of study were identification of comparative sentences from texts (e.g. reviews and forums) and extraction of comparative relation from identified comparative sentences [52]. In the study, authors used an opinion lexicon that contained two lists of negative and positive opinion words (or sentiment words) separately and in total covered around 6800 words [53].

WordNet [54, 55, 56] is another lexicon developed at Princeton University and is used for sentiment analysis with terms that are grouped based on their roles (i.e. verbs, nouns, adjectives and adverbs). In 2006 Esuli and Sebastiani, enriched WordNet by labeling terms according to their polarity [57]. WordNet-Affect is another version of this lexicon developed from Strapparava and Vlitutti (2004) works [58].

One of the largest lexicons in the sense of number of included words and emotions is NRC Emotion Lexicon [59, 60]. This lexicon, that covers eight Plutchiks' emotions in addition to the sentiment (i.e. positivity or negativity) of each word includes approximately 25000 word-senses and in its word-level version (i.e. union of all the senses of a word token) contains 14000 terms. NRC lexicon is created by use of the crowdsourcing technique [61]. In this technique a large task is broken into smaller and independent sub tasks, and distributed over internet or through other mass mediums. This technique benefits from variation in participants since annotators can

have different levels of education or familiarity with the target language. Another example of a lexicon formed over the crowdsourcing technique is the one developed by Warriner, Kuperman and Brysbaert which contains valence, arousal, and dominance of 13915 words [62].

All of the reviewed dictionaries until now demonstrate levels of emotions using discrete scores since assigning real numbers is not easy for humans and results might be inconsistent, i.e. different people have different levels of feelings toward terms. However, in the real world, words convey different continuous and comparative levels of an emotion. Therefore, with emphasize on this relativeness, it can be easy for individuals to compare a set of terms and order them according to the level of an emotion they convey. For example it is easier for people to say that "worse" is more negative than "bad". This idea is used in maximum difference (MaxDiff) or best-worst scaling method. In this technique participants are given a set of terms with size four and are asked to decide on the most positive, and the most negative ones. These two questions determine 4 out of 6 possible comparative relations of terms in the set. By assigning each set to a number of annotators and ranking terms from the most positive to the most negative, outcome is a list of terms with assigned real values. Clearly, if a term receives votes as the most negative by majority of annotators will fall far apart from another term that is mainly considered as the most positive. Also, if two terms voted equally the most positive (negative), in the ranking they will appear close to each other and associated scores would be close in value. Lexicons developed using the MaxDiff technique were used in the SemEval 2012 and 2015 shared tasks which are discussed later in this chapter. Kiritchenko et al.

used the same technique as well to develop a 1500 Twitter terms data set with real valued scored words and showed that calculated scores using this technique are reliable [63].

Automatic generation of lexicons benefits from the use of statistical and mathematical techniques for model development. Models learn from context or a set of already annotated samples and assign sentiment scores to unseen terms. Hatzivassioglou and McKeown [64], Turney and Littman [65] and Esuli and Sebastiani [57] studied over automatic generation of lexicons. Mohammad, Dunne and Dorr [66] generated a sentiment lexicon with 60000 terms from a thesaurus. Mohammad, Kiritichenki and Zhu [67] develop a new lexicon using tweets. Their developed lexicon had advantage of covering creatively spelled words, slangs, abbreviations, hashtags, and other informal forms of terms. These lexicons are covering both unigrams and bigrams.

2.6 Why Twitter and Tweets?

Many studies are conducted on different kinds of documents such as novels, reviews, emails, blogs, newspaper headlines and tweets. Among these resources, tweets are of high interest. In recent years, social media services are playing more important and active role in individuals' life. People use these services to freely share their thoughts, beliefs, emotions, feelings, and even their daily experiences with millions of people around the world. To have a better understanding, by January 2019, 500 million tweets are posted every day and monthly more than 326 million people use Twitter. Such a huge repository filled with emotions and thoughts is a valuable source for researchers. A group of studies revealed the correlation between changes

in number of tweets and stock market fluctuations since number of tweets is a sign for an important event [68]. In prediction of election results, Jahanbakhsh and Moon (2014) used tweets and with help of sentiment analysis, along with other techniques, truly determined that Obama will lead the 2012 election [69]. Shi et al.(2012) applied tweets sentiment analysis in combination with number of tweets for the republican primary election and perfectly predicted public opinion regarding two out of four candidates [70]. Customer satisfaction, election prediction, e-commerce, public health, social welfare, and intelligence gathering are just few examples of fields interested in tweets.

Such studies support the great predictive power behind tweets. They have found that Twitter is becoming more important than Facebook [71] since connections in Facebook are based on the levels of friendship, while in Twitter connections are focused on getting informed about events and news [72]. Nevertheless, working with tweets unfolds new set of challenges. Tweets are short messages with limited number of terms and unique characteristics that make them different from formal texts. Tweets are basically limited to 280 characters (140 before 2017) and due to this limitation people try to express their emotions completely in different way as do in long texts. Though, this limitation has not avoided tweets with mixture of emotions. Moreover, tweets are filled with informal terms such as abbreviations, emoticons, slangs, misspelling words, hashtags, and emoji. Some of these challenges are addressed in the following section.

2.7 Tokenization

In this section the represention of a document as a vector of features is discussed briefly. Documents, specifically tweets in our case, can be assumed as a set of consecutive terms or words. Terms can be either in their correct lingual form or written informally (e.g. character flooding or punctuation flooding) [82]. To convert a document to a set of features and consequently a vector of scores, it should be split into its components. Act of breaking a string into pieces of words, phrases, symbols, or any other substrings is called tokenization. According to the given definition in [75], token is defined as: *"An instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing."*

Tokenization and splitting conditions are generally an issue of language and may be problem dependent. One of the common techniques is white space tokenization where documents are split on white spaces. Consider a document (tweet) *"Not sure if thats an accomplishment or something to worry about"*. By passing it through white-space tokenization, the tokens returned are: *"Not", "sure", "if", "thats", "an", "accomplishment", "or", "something", "to", "worry", "about"*.

As the example shows, "thats" is considered as a single term. Therefore, due to inconsistencies in the way of writing terms (e.g. hyperplane vs. hyper plane vs. hyper-plane) tokenization should be done with care. Finally, the developed vector of tokens after tokenization is the feature vector for that document and it can be converted to a vector of score by applying different scoring techniques such as using lexicons, word2vec method, tf-idf scoring, etc.

2.8 Preprocessing

Data preprocessings are applied transformations on data before using them to develop models and generally includes data cleaning, normalization, transformation, and feature extraction and selection [73].

Data cleaning deals with outliers, illegal (e.g. out of range), and missing values (i.e. NAs) that can result in different inferred statistics. Differences in data types and ranges are also other important real-life challenges and obviously deciding on the correct algorithm to map them from one type or space to another can affect the developed model's performance. Two different frequently used techniques in data cleaning are discretization and normalization. By discretizing, continuous features are converted into discrete ones with a finite number of values. However, deciding on the best splitting value is yet an important issue. Supervised and unsupervised approaches are two common sub-branches of discretization technique [73].

Normalization as a preprocessing step is employed to map data into smaller or similar range of values. Two common techniques for normalization are Min-Max scaling and z-score normalization [73]. In this study the later method according to Eq. 2.1 is applied and data are mapped in a range with mean 0 and standard deviation 1.

$$x' = \frac{x - \bar{x}}{\sigma_x} \qquad \text{(Eq. 2.1)}$$

In Eq. 2.1, x and x' are respectively values of an instance before and after normalization, $\bar{x}$ represents the average value, and σ_x is the standard deviation of instances [73].

Feature selection and extraction is one of the fundamental subfields of data preprocessing. Data sets in real world contain large number of samples with hundreds of thousands of features while a few of them may actually be related to the target. Features mainly are grouped into three categories as relevant, irrelevant and redundant [73]. Therefore, feature selection algorithms have two basic parts; selection, that generates a subset of attributes and evaluation, that determines how well the generated subset is [73]. Basically these two steps are performed in a recursive way until a stopping criteria meets. Deciding on the most discriminative features is also a topic of data preprocessing that will be discussed in this chapter.

Number of instances and imbalanced data sets are two more remarkable issues concerned in data preprocessing. Large data sets, although are of interest, can result in infeasibility of learning [73]. Thus, for data reduction, sampling techniques such as random, clustered, and stratified sampling are among well known and commonly practiced methods. Regarding imbalanced data sets, removing samples from over presented classes or duplicating train samples are two solutions [73].

Required preprocessing techniques in working with tweets such as outlier detection and normalization are discussed further in the next chapters when they are applied on real data set. Two techniques for feature extraction are discussed next.

2.9 Tf-idf scoring

Tf-idf, which stands for term frequency-inverse document frequency, is a statistical measure to define how important a word is to a document in a corpus [74]. The term's weight increases as it occurs more in a document and drops as it appears frequently

in different documents. Tf-idf score is composed of two parts. The first part, Term Frequency (TF), is defined as the number of times a term appears in a document. A normalized version of TF is given by Eq. 2.2 where n_{td} is divided by the total number of terms in document d, (n_d) [75]. The second part is inverse document frequency (idf, Eq. 2.3), that is defined as the natural logarithm of number of documents in a corpus (N) divided by the total number of documents where a specific term appears in (i.e. document frequency (df_t)) [75].

$$tf_{t,d} = \frac{n_{td}}{n_d} \quad \text{(Eq. 2.2)}$$

$$idf_t = \ln(\frac{N}{df_t}) \quad \text{(Eq. 2.3)}$$

If a term appears occasionally in a few documents, it is conveying some information regarding those specific documents. However, if it happens often in the entire corpus, it cannot be discriminative. For example in a corpus with combination of economics and medical science documents, the term "exchange stock" receives higher weight since it appears less frequently, and hopefully it is more discriminative in comparison to stopwords (commonly used terms) such as "the" and "is" which usually receive low scores or are basically ignored. According to Eq. 2.2 and Eq. 2.3, tf-idf score is defined as the product of tf and idf [76].

$$tf - idf_t = tf_{t,d} \times \ln(\frac{N}{df_t}) \quad \text{(Eq. 2.4)}$$

2.10 Word2vec

Word2vec model has received extensive attention in machine learning and especially in text mining due to its ability in sentiment detection. Indeed, it is used to find similar words which are used in the same context within a sentence (tweet).

Sentence completion, selecting irrelevant term from a list of given terms, and synonyms detection are examples of word2vec model application, without which extensive programming would be needed [77]. Starting with a typical example, consider following statement.

"Woman is to queen as man is to king".

Word2vec vision is to represent "man", "king", "woman" and "queen" in form of vectors and discover a relation such as Eq. 2.5. Hence, it can offer valuable sentiment information if words can be presented in form of vectors [78].

$$v(king) - v(man) + v(woman) = v(queen) \qquad \text{(Eq. 2.5)}$$

Word2vec is similar to a neural network structure with a single hidden layer (also named projection layer [77]) and is of two models, Continuous Bag of Words (CBoW) and Skip-Gram. These two models work in opposite directions. CBoW predicts a word based on a provided context (e.g. a sentence) while Skip-Gram predicts a context given a word [78]. Skip-gram, introduced by Mikolov et al., is an efficient model to represent large amounts of unstructured text data in form of vector that can be used for machine learning [79, 80]. The next section discusses CBoW as it is applied later in this study.

2.11 Continuous Bag of Words (CBoW)

CBoW model, as the name conveys, develops over a bag of words in which orders are not taken into account. The input of the model is a binary vector of size V (vocabulary size) with elements corresponding to each word. Consequently, vector elements are all zero except for the given terms to the model. In the simplest form, single-word-context, the target word is predicted by a single given word. Thus, both input (x) and output (y) are one-hot encoded vectors in which all elements are zero except for

x_k and y_k, where k is the index of the input and output single term in the vector of words. In CBoW model there are two weight matrices; one of size $V \times N$ from input to hidden layer (W, Eq. 2.6) and an $N \times V$ from hidden to output layer (W') where V is the vocabulary size and N is the number of parameters in the hidden layer. In Eq. 2.6 each row is an N dimensional vector of weights that represents its associated word. By passing the one-hot vector of input word (x, Eq. 2.7) to the system we obtain:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{V1} & w_{V2} & \dots & w_{VN} \end{bmatrix} \quad \text{(Eq. 2.6)}$$

$$x^T = \begin{bmatrix} x_1 & x_2 & \dots & x_k & \dots & x_V \end{bmatrix} \quad \text{(Eq. 2.7)}$$

$$h_{IW} = x^T W = \begin{bmatrix} x_1 & x_2 & \dots & x_k & \dots & x_V \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{V1} & w_{V2} & \dots & w_{VN} \end{bmatrix} \quad \text{(Eq. 2.8)}$$

$$h_{IW} = x^T W = \begin{bmatrix} x_k w_{k1} & x_k w_{k2} & \dots & x_k w_{kN} \end{bmatrix} \quad x_k = 1 \; and \; x_{k'} = 0 \quad \forall k \neq k' \quad \text{(Eq. 2.9)}$$

$$= \begin{bmatrix} w_{k1} & w_{k2} & \dots & w_{kN} \end{bmatrix} \quad k^{th} \text{ row of } W \quad \text{(Eq. 2.10)}$$

The result, h_{IW}, at the hidden layer is a vector of scores of size N representing the input word. From the hidden layer a new matrix of weights (W') is applied to h_{IW}

and multiplication yields a vector with scores of each term in the vocabulary. u_j is the score of term j that is equal to ${v'_{w_j}}^T h$ where v'_{w_j} is the j^{th} column of matrix W'. The scores obtained measure the level of match between context (input word) and the next (predicted) word. Now posterior probability of each term is computable using softmax (log-linear) classification model.

$$p(w_j \mid w_I) = y_j = \frac{exp(u_j)}{\sum_{j'=1}^{V} exp(u_{j'})} \qquad \text{(Eq. 2.11)}$$

CBoW model is similar to other models which have a training phase. In this phase the conditional probability of observing the actual output (j^{th} element) of a given word is maximized by getting the derivative with respect to u_j. Maximization first finds the best values of weights between hidden layer and output. Then, by computing the derivative of hidden layer parameters with respect to W's components, weights between input and hidden layer are optimized. Nevertheless, initial weights can be set empirically [78, 77].

Although word2vec performances well in sentiment detection, it has weaknesses such as ambiguity in selection of correct word in case of having more than one choice (e.g. having many cities named London), difficulty in parameter setting, and difficulty in performance evaluation since it is an intellectual task [77].

2.12 Machine Learning and Model Development

Automatic detection of sensed emotion intensity by a speaker or in general determination of emotion requires developing a system with the ability of learning from provided samples and making decisions on new and not already annotated instances. Generally, there are two ways of training a model. The classification system should either be taught beforehand with already labeled data or learn a series

of rules by itself to make decisions accordingly. Machine Learning (ML), a sub-discipline of Artificial Intelligence (AI), is the science of machines facilitation by algorithms and experiencing new samples to automatically learn and improve answers accuracy.

2.12.1 Supervised and Unsupervised Learning

Machine learning based classification techniques are categorized into two major branches as supervised and unsupervised learning. In supervised learning, the machine is provided with a set of already labeled instances and with sufficient number of samples, the algorithm learns to predict the labels (classes) for new inputs. In supervised learning algorithms it is also possible to compare the true labels of train instances with the predicted ones and optimize the model parameters before testing it on real data. Accuracy, precision, recall, and F-score are some major evaluation measures which show the success of a model.

Unsupervised learning algorithms, in contrast to supervised ones, do not provide the system with labeled samples. Thus, the system tries to infer and model a function to describe the hidden patterns among data. Moreover, in unsupervised learning the system cannot measure how close the predicted labels are to the true ones.

Although using a supervised or unsupervised technique is often problem dependent, both of these techniques are formed over a set of derived characteristics. In the machine learning context, a set of measurable characteristics of an instance (e.g. tweet length) is known as the feature set. Hence, learning is the process of understanding how a set of features represent a label (supervised) or how features form a pattern among themselves (unsupervised). In the first case the machine tries

to find a pattern between feature and given labels. In the later case, the algorithm attempts to discover hidden patterns among the features. Learning algorithms usually develop tens of hundreds of features and terms. Their combination is a known feature which is used in inferring the emotions. "n-gram" (e.g. unigram, bigram, or pair-gram) is one of the widely used feature generator techniques that partitions a document into a set of n consecutive or paired tokens.

In the supervised learning approach, each instance of training data is encoded as a vector of features (f) with length l (Eq. 2.12), and a class label (L). This vector is passed to the machine for analysis and developing a prediction function (model) that can be applied to unseen test data for label prediction.

$$f = [f_1,\ f_2, \ldots,\ f_l]_l \qquad \text{(Eq. 2.12)}$$

In the field of automatic emotion classification, the unsupervised learning approach is referred as the affect lexicon-based approach [81]. As the name expresses affect lexicons are used for voting or scoring each term in a tweet. Finally, majority of votes or summation of scores determines the dominant emotion. Affect lexicon, as described earlier, is a list of terms with assigned emotions or scores. For example, "celebrating" is a term under "joy" category and depending on the lexicon type, comes with a real-value or categorized score as an indication to its intensity level. Thus, if a tweet contains the term "celebrating", it receives one vote or a score for emotion "joy".

Using affect lexicons for model development is usually simple and memory efficient, although training is greatly influenced by the lexicon's quality and can be less accurate

since it is mainly a look up process [82]. For instance, consider the term "kill" with primarily negative sentiment, which in case of a detergent advertisement conveys positive sense. Similarly, sentences commonly convey emotions indirectly through meaning. Nevertheless, unsupervised methods have been used widely for sentiment analysis in commercial needs by many researches [57, 82, 83, 84, 85, 86]. Supervised learning, in contrary, results in more accurate predictions as it considers the sentence's arrangement and terms' combination. Studies of Mohammad (2012) on newspaper headlines and blog post demonstrated that the combination of supervised techniques with affect lexicons can improve accuracy in predictions too [81].

2.13 Model Evaluation

Performance evaluation is a critical step in model development. Consider the care of developing a supervised classification model on a data set with two classes: positive and negative. By testing the model on a test set, the results obtained belong to one of the following categories:

- **True Positive (TP)**: model correctly predicts the positive class.
- **True Negative (TN)**: model correctly predicts the negative class.
- **False Positive (FP)**: model incorrectly predicts a negative class as positive
- **False Negative (FN)**: model incorrectly predicts a positive class as negative

Above outcomes can be tabulated as in Table 2.1. Based upon Table 2.1, the fraction of correctly labeled instances defines the accuracy of the model as given in Eq. 2.13.

Table 2.1: Confusion matrix

		True class	
		positive	**negative**
Predicted class	**positive**	TP	FP
	negative	FN	TN

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{(Eq. 2.13)}$$

Although, models with higher accuracy are preferred, this can be misleading as well. Assume an unbalanced two-class classification problem. If all the samples of the larger class are labeled correctly, the accuracy will be a reasonably high value, while the model indeed failed to classify samples from the smaller class. Therefore, precision, the fraction of relevant samples among labeled instances, and recall, the fraction of correctly labeled relevant samples over all relevant samples, are two other measures used for model evaluation.

Using Table 2.1, precision and recall are defined as,

$$precision = \frac{TP}{TP+FP} \quad \text{(Eq. 2.14)}$$

$$recall = \frac{TP}{TP+FN} \quad \text{(Eq. 2.15)}$$

Usually, an increase in precision results in a decrease in recall and vice versa. Therefore, a single metric for better comparison of models, known as the F_β-score, is defined as the weighted average of precision and recall (Eq. 2.16).

$$F_\beta - score = (1+\beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall} \quad \text{(Eq. 2.16)}$$

For the special case of $\beta = 1$, F_1-score is defined as:

$$F_1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad \text{(Eq. 2.17)}$$

Through this thesis F-score refers to F_1-score. Generalization of the two class case to a data set with more than two classes results in achieving different precision, recall, and F-score values for each class. To summarize these measures into a single value,

micro or macro averaging methods are used. Micro average adds up individual values from Table 2.1 for each class and calculates an F-score value. However, macro average is an arithmetic average of precision, recall, or F-score on all classes [87]. The micro precision, micro recall, and micro F-score can be calculated using Equations 2.18, 2.19, and 2.20 below:

$$micro\ precision = \frac{TP_1 + TP_2 + \ldots + TP_c}{TP_1 + TP_2 + \ldots + TP_c + FP_1 + FP_2 + \ldots + FP_c} \quad \text{(Eq. 2.18)}$$

$$micro\ recall = \frac{TP_1 + TP_2 + \ldots + TP_c}{TP_1 + TP_2 + \ldots + TP_c + FN_1 + FN_2 + \ldots + FN_c} \quad \text{(Eq. 2.19)}$$

$$micro\ F - score = 2 \times \frac{micro\ precision \times micro\ recall}{micro\ precision + micro\ recall} \quad \text{(Eq. 2.20)}$$

2.14 Regression, Pearson and Spearman Correlation

Classification is concerned with predicting labels that are either from a set of discrete numbers or some textual values. However, if the predicted values are quantities with real numbers then regression techniques are applied. Regression is a statistical method for developing mathematical functions to represent a relation between a set of features and target variables. Given that target values are continuous, performance in regression is reported as error, by measuring the distance of predicted values with their true ones. As we know the most common scale used is Mean Squared Error (MSE). Mean squared error is defined as the average squared difference between true and estimated values. In addition to MSE, Pearson and Spearman correlations are other simple performance measures. Pearson correlation computes the degree of strength and linear relation between estimated (Y) and real values (X) and reports a result in range of -1 to 1 (Eq. 2.21). Value 1 is an indication for complete positive correlation and -1 shows variables are perfectly correlated but in reverse direction.

In Eq. 2.21, $cov(X,Y)$ is the covariance of true (X) and predicted (Y) values, and μ_X, μ_Y, σ_X, and σ_Y are respectively average and standard deviations of variables X and Y.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sigma_X \sigma_Y} \qquad \text{(Eq. 2.21)}$$

Spearman correlation, similar to Pearson correlation, is a metric which measures how predictions are correlated to the true values. However, here correlations are measured between ranks of actual and predicted target (Eq. 2.22).

$$\rho_{r_X,r_Y} = \frac{cov(r_X,r_Y)}{\sigma_X \sigma_Y} \qquad \text{(Eq. 2.22)}$$

2.15 Feature Selection

With prevalent collecting and storing devices in recent years, we are facing with massive amounts of high dimensional data in our daily life. Data collected from wide ranges of resources such as social media, bioinformatics, e-commerce, etc. contains useful information and there is a growing need for effective and efficient data management. Although typically more data suggests more information and with sufficient resources using redundant features is not a major concern, studies have revealed in practice that applying machine learning and data mining techniques on high dimensional data sets may be subject of curse of dimensionality, which results in higher computational cost and model complexity, lower training speed, and over-fitting [88]. Over-fitting is defined as the condition of having a well fitted model on training data with low error rate but low performance and high error rate on test set. Curse of dimensionality may negatively affect algorithms that are designed for low dimensional data. Consequently, dimension reduction is considered as a crucial step. In text mining problems, documents are represented by vectors that

store a value for each occurrence of terms. Size of these vectors normally reaches to hundreds of thousands of terms. However, by dropping very common or very rare terms, the vector size is reduced to thousands of more representative terms [89]. There are different means of feature reduction. One possible way is checking the performance of all possible combinations of features and deciding on the well performing ones. This method, so called the brute-force approach, is very inefficient. Consider the case of having 10 features. Number of all possible combinations equals 1023 and undoubtedly, examining all feature subsets and developing a model for each is waste of resources. In real life with hundreds of features, conditions can worsen. Therefore, using some pruning and selection techniques is unavoidable. These techniques, although may ignore parts of the solution space and may result in obtaining a sub-optimal solution, nevertheless increases speed and saves time. Feature selection techniques similar to learning techniques are categorized into two main categories: supervised and unsupervised approaches [90].

2.15.1 Supervised Feature Selection

This method is generally developed for classification or regression problems and uses labels to choose the most correlated and discriminative features. Wrapper method is the selection approach that benefits from the learning algorithm's performance as a clue of feature relevancy. However, if selection method is independent of learning phase, it is known as filter method [90].

2.15.2 Unsupervised Feature Selection

Unsupervised feature selection similar to unsupervised learning algorithm does not have access to classes and is mainly used for clustering problems. Hence, the feature selection algorithm tries to find measures of relevancy. Unsupervised feature selection

is also of two types; wrapper based approach which benefits from learning algorithm, and filter method, which is independent of learning [90].

2.15.3 Wrapper Based Methods

Wrapper based methods evaluate the quality of features by relying on the performance of the learning algorithm and decide on the best ones. This method is generally composed of two main steps: searching for subsets of features and selecting the best ones. The procedure starts with an initial set of attributes, which are passed into an already defined learning algorithm and their performances are measured. According to the performances, the combination of attributes is revised and the whole process iterates until reaching a stopping criteria. The stopping criteria might be consideration of all possible attributes, consistency in the performance, or achieving the highest performance [90].

In wrapper based techniques, the selection of an initial attribute set and later deciding on the surviving features are critical issues. Since the wrapper method searches the feature space for the best solution, it can get into an exhaustive task in large spaces. Therefore, different search strategies are introduced to ease subset selection. Greedy search strategy is an intuitive approach that follows local optimal answers in hope of attaining the global one. Greedy algorithm saves time, increases speed and is robust against over fitting. However, it has two main drawbacks. First, there is no assurance that the global best solution will be achieved and the algorithm might get stuck in a local sub-optimal solution. Second, features after being selected and combined are not evaluated again [91]. Three different types of greedy search strategies are discussed next.

Forward Selection (FS)

In this strategy, all features are evaluated individually and the best performing one is selected as the initial set. In each of the succeeding iterations randomly chosen attributes expand the feature set from the previous step until the combination formed cannot improve the performance anymore. Forward selection can be performed in a non-random way as a greedy searching algorithm. In greedy search, algorithm tries to find the best (global) solution by following local optimums. However, there is always chance of getting stuck in local optimums. Here after selection of the best feature in the first iteration, following repetitions continue with evaluation of performance of each feature in combination with the attributes survived from the previous steps. The feature, that improves the performance of the combination, is kept for the next iteration. In this study, the former technique with random nature is referred as Random Forward Selection (RFS) while latter is named as Forward Selection (FS).

Simplified Forward Selection (SFS)

This approach begins with calculating the performance of all single attributes and sorts them in descending order. In the first iteration the best performing feature is combined with the second best one. If performance improves the combination is kept and the third top attribute is added and so forth. Otherwise, iteration stops and the last best combination is chosen as the best attribute set.

Backward Selection (BS)

Backward selection is similar to forward selection, though in reverse direction. In this strategy iterations begin by using the combination of all features and computing the performance. At each of the following iterations, one of the randomly selected

features is eliminated from the combination and the performance is checked again. If elimination improves the performance, the remained set is passed on to the next iteration; otherwise, the procedure terminates and the feature set before last removal is chosen as the best set. The procedure also terminates if only one feature remains.

Single Best (SB) and Combination of All Features

Single best strategy selects the best individually performing feature. In contrast combination of all features measures the performance when all features are used together. Both of these techniques are commonly practiced in order to form a baseline that results of other techniques can be compared to.

2.16 Classifier Selection

Classifiers (i.e. SVM in this case) that are trained with extracted features (e.g. tf-idf scores) are indeed mapping input data into specific classes (i.e. four intensity levels). Assume of having a combination of classifiers that are trained over a set of single or combination of well-chosen feature sets. This combination may improve performance by removing or adding classifiers, almost in the same way as feature selection techniques. In fact some classifiers might perform better on some subspaces of the input domain, but may not perform well on the whole data space. In other words, classifiers might have "domain of expertise" that is typically not the entire data space [92]. Thus, the aim is to take advantage of expertise domains and improve the results.

According to experimental studies, classifier selection and combination is an effective effort if the selected classifiers are diverse (i.e. making different errors) and accurate individually (i.e. having low error rate) [93, 94, 95]. Moreover, studies

suggest that better results are achieved with negatively dependent classifiers, a criterion that is hardly met since classifiers often make identical mistakes on difficult patterns [96]. Microarray data classification is one of such areas that classifier selection can dramatically improve results. Microarray data sets have few numbers of instances with high dimensionality that prevent classifiers to develop accurate models [97].

There are different techniques for classifier (combination) fusion such as Majority Voting and Dynamic Classifier Selection. Majority Voting, the method that is applied in this study, gives one positive vote for the correctly predicted class of each sample per classifier. Ultimate label of each sample is the class with the highest vote. Classifiers combination performance is then evaluated by comparing predicted labels with true labels using different metrics.

2.17 Linear Support Vector Machines (SVM)

There are different classification algorithms such as Naïve Bayes, logistic regression, and Support Vector Machines (SVM) just to name a few. Deciding on an appropriate technique depends on the data set and the problem in hand. In this study a linear SVM as a classification algorithm is used. Note that the majority of contents for this section are coming from [98, 99, 100, 101].

SVM is an intuitive, well founded technique with successful performance in digit recognition, computer vision, and text categorization [100]. It was first introduced by Vladimir Vapnik in 1995 with the aim of binary data classification in a D-dimensional space [102]. Assume of having N training samples $x_i,\ i = 1, \cdots, N$, where each point

(sample) is of dimensionality D (i.e. each sample has D features). In the simplest case, data are from two classes -1 and $+1$. Hence, each sample as shown in set 2.12 can be represented as:

$$\{x_i, y_i\} \quad where \quad i = 1, \ldots, D \quad y_i \in \{-1, +1\},\ x \in R^D \qquad \text{(Eq. 2.23)}$$

Moreover, assuming that the classes are linearly separable, a hyperplane in the D dimensional space of samples can be defined to split data into two classes as all points belonging to the same class fall into the same side of Eq. 2.24. Nevertheless, in real world data is not always linearly separable. Depending on the problem since mathematically simpler boundaries are preferred, data can be mapped into a new feature space using kernels where linear boundaries can be defined in a higher dimensional space (Figure 2.2) [99]. The hyperplane or the decision boundary, as is named in classification terminology, is defined by:

$$w \cdot x + b = 0 \qquad \text{(Eq. 2.24)}$$

where w is orthogonal vector to the hyperplane and $\frac{b}{\|w\|}$ is the perpendicular distance

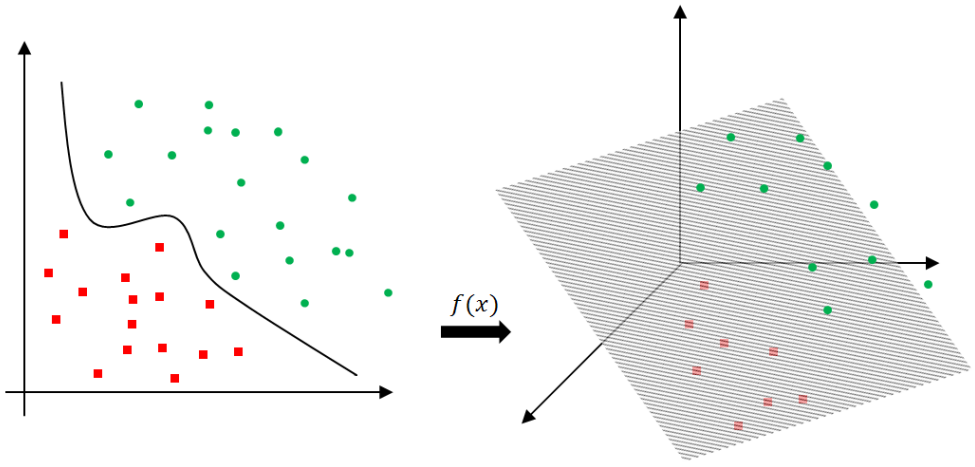


Figure 2.2: Map of data from non-linearly separable space into linearly separable space

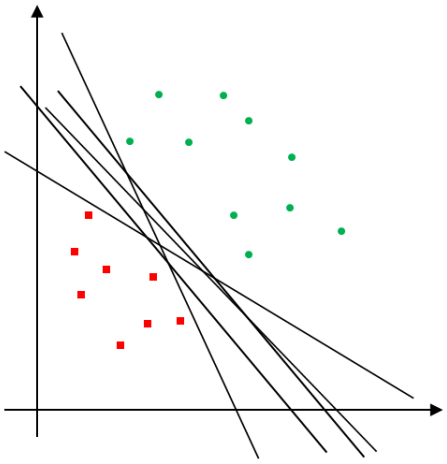


Figure 2.3: Many decision boundaries exist

from the origin to the hyperplane. Therefore, with a linear equation, w and b are found using train data satisfying following inequalities:

$$x_i \cdot w + b \geq +1 \quad for \quad y = +1 \qquad \text{(Eq. 2.25)}$$

$$x_i \cdot w + b \leq -1 \quad for \quad y = -1 \qquad \text{(Eq. 2.26)}$$

These two conditions can be merged into:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall i \qquad \text{(Eq. 2.27)}$$

As illustrated in Figure 2.3, the special case of $D = 2$, many candidate decision boundaries exist. Here the question is how to select the best one. To choose the best decision boundary among all possible ones, SVM decides on the basis of margins and selects the one with maximum distance from the closest samples. The closest samples to the decision boundary are known as support vectors (Figure 2.4). By considering the hyperplanes passing through the support vectors, two planes v_1 and v_2 are defined as:

$$x_i \cdot w + b = +1 \quad for \quad v_1 \qquad \text{(Eq. 2.28)}$$

$$x_i \cdot w + b = -1 \quad for \quad v_2 \qquad \text{(Eq. 2.29)}$$

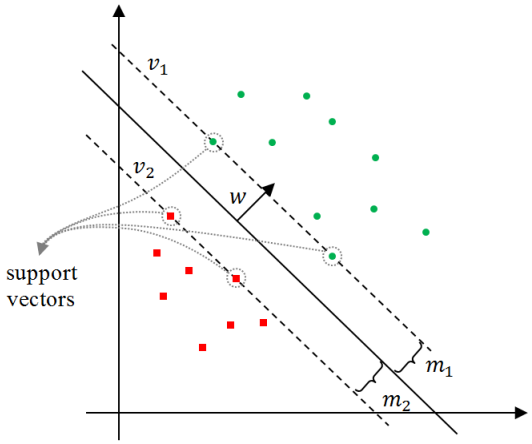


Figure 2.4: Support vectors (marked samples), margins and decision boundaries

The SVM margin (m_1 and m_2) is defined as the distance between the decision boundary and the imaginary lines passing through the support vectors. Hence the best solution is the one which maximizes the margin while $m_1 = m_2$. Support vectors are very important since can be misclassified easily and change the classification boundaries. Furthermore, as will be proved later, the decision boundary is merely specified by these points. Now consider a given hyperplane, where all pairs of $(\lambda w, \lambda b)$ define the same planes except for different distances to a given sample. Hence, to obtain the geometric distance between the samples and the boundary, the hyperplane is normalized by the length of the orthogonal vector to the hyperplane (w). From the inequality Eq. 2.27 we have

$$\frac{y_i(x_i \cdot w + b)}{\|w\|} \geq \frac{1}{\|w\|} \qquad \text{(Eq. 2.30)}$$

Here, we are interested in maximizing $\frac{1}{\|w\|}$ or minimizing $\|w\|$ subject to $y_i(x_i \cdot w + b) - 1 \geq 0,\ \forall i$. Also minimizing $\|w\|$ is equivalent to minimizing $\frac{1}{2}\|w^2\|$ which makes it possible to solve the problem using Quadratic Programming (QP) optimization algorithms.

$$\begin{aligned} &\min \quad \tfrac{1}{2}\|w^2\| \\ &s.t. \quad y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall i \end{aligned}$$

To solve this problem, we use Lagrange multiplier $\alpha, \alpha_i \geq 0 \; \forall i$, and have:

$$\begin{aligned} \min\, L_P &\equiv \tfrac{1}{2}\|w^2\| - \alpha\left[y_i(x_i \cdot w + b) - 1\right], \quad \forall i \\ s.t. &\equiv \tfrac{1}{2}\|w^2\| - \textstyle\sum_{i=1}^{L} \alpha_i \left[y_i(x_i \cdot w + b) - 1\right] \\ &\equiv \tfrac{1}{2}\|w^2\| - \textstyle\sum_{i=1}^{L} \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^{L} \alpha_i \end{aligned} \tag{Eq. 2.31}$$

In order to minimize L_P on w and b, its derivative with respect to w and b is set to zero:

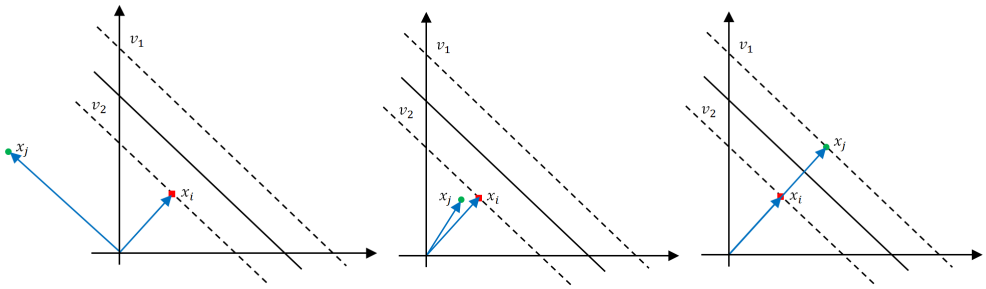
$$\frac{\partial L_P}{\partial w} = 0 \;\Rightarrow w = \sum_{i=0}^{L} \alpha_i y_i x_i \tag{Eq. 2.32}$$

$$\frac{\partial L_P}{\partial b} = 0 \;\Rightarrow w = \sum_{i=0}^{L} \alpha_i y_i = 0 \tag{Eq. 2.33}$$

The result obtained from equation 2.32 reveals this fact that w is in fact a linear combination of training samples [99]. However solving the model in Eq. 2.31 and minimizing it, is not trivial. A simpler task is solving its dual form. Thus, by substituting equations 2.32 and 2.33 in Eq. 2.31, instead of minimizing with respect to w and b subject to $\alpha, \alpha_i \geq 0 \; \forall i$, the outcome depends only on α which should be maximized accordingly subject to w and b (Eq. 2.34).

$$\begin{aligned} \min\, L_D &\equiv \textstyle\sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ s.t. &\quad \textstyle\sum_{i=1}^{L} \alpha_i y_i = 0 \\ &\quad \alpha_i \geq 0 \quad \forall i \end{aligned} \tag{Eq. 2.34}$$

In Eq. 2.34, the second constraint ensures that optimal condition for b is satisfied. By replacing $y_i y_j x_i \cdot x_j$ with H_{ij} and rewriting Eq. 2.34:



(a) Case one, x_i and x_j are dissimilar.

(b) Case two, x_i and x_j are similar and predicting same class.

(c) Case three, x_i and x_j are similar but predicting different classes.

Figure 2.5: Different cases regarding position of x_i, x_j in the feature space and their predictions

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T H \alpha \\ s.t. \quad & \sum_{i=1}^{L} \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \quad \forall i \end{aligned} \qquad \text{(Eq. 2.35)}$$

Interestingly, the dual form only requires the dot product of each input vector x_i. This characteristic helps to map data from one space to another using kernel functions. By maximizing model 2.35, three cases are possible (Figure 2.5). In the first case features x_i and x_j are completely dissimilar and their inner product equals to zero (Figure 2.5a). Consequently, they do not have any effect on L_D. In the second possible case, x_i and x_j are similar hence $x_i \cdot x_j$ is not zero and two subcases can arise. In subcase one (Figure 2.5b), x_i and x_j predict same classes; thus, value of $\alpha_i \alpha_j y_i y_j x_i \cdot x_j$ will be positive and decreases L. While in subcase two (Figure 2.5c), x_i and x_j result in opposite predictions and product of $\alpha_i \alpha_j y_i y_j x_i \cdot x_j$ will be negative and L increases.

These cases clearly prove that important features are the most discriminative ones [101]. So far optimal vector of α and accordingly optimal value of w is found. To find b, we already knew that any support vector point (x_s) is in form of:

$$y_s(x_s \cdot w + b) = 1 \qquad \text{(Eq. 2.36)}$$

By substituting w into Eq. 2.36:

$$y_s(\sum_S \alpha_m y_m x_m \cdot x_s + b) = 1 \qquad \text{(Eq. 2.37)}$$

where S is the set of indices of all support vectors (SV). Since SV are expected to have maximum distances, by multiplying y_s in Eq. 2.37 and setting $y_s^2 = 1$ (normalization):

$$\begin{gathered} y_s^2(\textstyle\sum_S \alpha_m y_m x_m \cdot x_s + b) = y_s \\ b = y_s - \textstyle\sum_S \alpha_m y_m x_m \cdot x_s \end{gathered} \qquad \text{(Eq. 2.38)}$$

Eq. 2.38 computes b for each support vector m. To have a single value, average over all values of support vectors in S is found:

$$b = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s) \qquad \text{(Eq. 2.39)}$$

Here optimal values for b and w are already computed and the separation boundary (hyperplane) is defined accordingly. Thus, a support vector machine is formed. Test data in the next step is fed into the developed model and with optimal values of b and w, and using inequality in Eq. 2.27, predicted classes are determined.

Chapter 3

SYSTEM OVERVIEW

3.1 Introduction

WASSA-2017 competition was the first shared task in emotion intensity level detection felt by the speaker of a tweet. The competition was held with twenty-two teams who were asked to develop regression models over already annotated data to decide on the level of experienced emotion by a tweeter (i.e. who posts a tweet) on a new unseen sample. The competition was narrowed into four emotions including anger, joy, fear, and sadness and teams had to report performances individually on each emotion, although final ranking was on the basis of average performance on four emotions. Pearson and Spearman correlation measures were applied for performance evaluation between predicted and actual intensities. Moreover, participants were allowed to use any set of features, regression models, and tools for model construction of their choice. Among various used tools and libraries in the competition the most popular ones were TensorFlow [103] and Sci-kit learn [104] that both use Python libraries [9, 105].

Based on the announced results, the best team, the Prayas system [9, 11], achieved Pearson correlation of 0.747 on average with highest 0.765 on anger and lowest 0.732 on fear and sadness. This team used word embeddings, word2vec, sentences embeddings and affect lexicons such as AFINN [50], Bing Liu [53], NRC lexicon set [63, 67, 106], MPQA [48], WordNet [56], and In-house lexicon as features. The

IMS system [9, 10] ranked second with average correlation of 0.722. Its best performance was on anger emotion similar to Prayas. However, its lowest performance for emotion sadness was around 4% less than the Prayas system. Regarding the features applied, Prayas and IMS worked on almost similar feature sets except some differences in the lexicons used. SeerNet system [8, 9] with average of 0.708 ranked as the third best system. It used same features similar to the Prayas system, except sentence embeddings and applied five different regression models including AdaBoost, Gradient Boosting, random forest, Support Vector Regression (SVR), and an ensemble. A detailed list of used features extraction techniques and applied regression models can be found in the competition paper [9]. Nevertheless, word embeddings and affect lexicons along with Neural Networks (NN) and SVRs were among the most practiced feature selection and regression techniques, respectively.

As stated earlier, the competition was a regression task in nature. However, this study attempts to classify tweets according to their emotion intensities instead. The same data set as the WASSA-2017 competition set is used and a wide variety of available feature set resources such as word embeddings, namely word2vec, tf-idf scoring and affect lexicons are used. Moreover, Scikit-learn [104], gensim [107], Pandas [108], and NumPy [109] are tools used and libraries for model development and system learning, that all run over Python [105].

Figure 3.1 depicts the architecture and path that is followed in development of an emotion detection system. The process uses a train data set and continues with a series of modifications and feature extractions at each step. Finally, extracted features

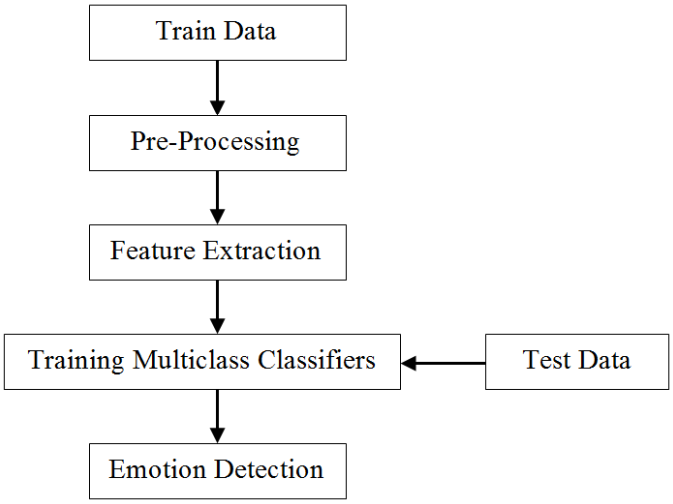


Figure 3.1: Porposed approach for emotion intensity detection

are used by an SVM classifier to train a model. The constructed model is then tested on the test data. Data set construction from gathering to manual tweets annotation is discussed in details in the next two sections. Section 3.4 discusses pre-processing techniques such as tokenization and data preparation for model training. Section 3.5 reviews feature extraction methods and finally model construction is discussed in the last section.

3.2 Data Set

The data set used in this study is the same announced data set for the WASSA-2017 shared task on emotion intensity detection [9]. The data set is a collection of tweets consisting of four emotions: anger, fear, joy, and sadness and as clarified in the competition paper, created using Twitter API. To decide on the most relevant tweets firstly a set of 50 to 100 query terms of each emotion is selected. Query terms are indeed the most relevant words to differentiate between levels of an emotion that are chosen from Roget's Thesaurus. This thesaurus provides around 1000 categories of words each containing an average of 100 closely related terms. Every category is

represented with a headword. Query terms are chosen from the candidate category with the closest headword in meaning to a target emotion. Eventually, the selected query terms represent high levels of association with different intensity levels of an emotion [9].

Data collection process using Twitter API for the data set used in this study started in November 22, 2016 and continued for three weeks. During this period tweets with query terms in them were collected and then refined by discarding retweets and tweets with URLs. Furthermore, for a uniformly distributed data set regarding query terms and tweeters, at most 50 tweets for each query term and at most one tweet for every tweeter-query term combination were kept [9]. After refining master set or the final data set, that covers 7097 tweets, it was passed on to manual annotation by applying best-worst scaling technique. In this technique, as briefly discussed in section 2.5, each participant is given four tweets (4-tuple) at a time, and is asked to determine tweets that the speaker experienced highest and lowest emotion intensities. In total $2 \times N$ distinct random 4-tuple tweet sets, where N is the total number of samples under an emotion, were generated in a way that each tweet appears in 8 different tuples and no pair of tweets occurs more than once. Each 4-tuple set was annotated by three independent persons using a questioner formed over CrowdFlower, a crowdsourcing platform. Furthermore, around 5% of samples were annotated manually by authors to avoid malicious annotations and also for later use as gold set. Finally, tweets, based on the percentage of times voted as the most and least intense, are assigned a real-value score using following formula.

$$intensity(t) = \%most(t) - \%least(t) \qquad \text{(Eq. 3.1)}$$

3.3 Train, Development and Test Data Sets

The master data set is partitioned into three subsets for model training and testing. Half of the tweets were assigned for training, 5% for development (validation), and 45% was reserved as test set. The details of the data sets is given in Table 3.1.

Table 3.1: Number of instances per data set

	Train	**Dev.**	**Test**	**All**
Anger	857	84	760	1701
Fear	1147	110	995	2252
Joy	823	74	714	1611
Sadness	786	74	673	1533
All	3613	342	3142	7097

Table 3.2 shows sample tweets from the joy data set. Each entry of the table contains an ID that uniquely identifies a tweet, an affect dimension to determine the emotion and an intensity class to indicate the level of inferred emotion.

Table 3.2: Sample tweets from joy data set

ID	**Intensity**	**Tweet***	**Emotion**	**Note**
31108	2	#happiness #recipe: an open mind,#laughter ...	joy	2: moderate level of joy can be inferred
30827	3	I love my family so much#lucky #grateful ...	joy	3: high level of joy can be inferred
30621	0	Pinterest one dessert... Next thing you know ...	joy	0: no joy can be inferred
31475	1	Accept the challenges so that you can feel ...	joy	1: low level of joy can be inferred
31129	3	i just spent $40 on big little sis tomorrow ...	joy	3: high level of joy can be inferred

*part of tweets are given.

In this study 19 feature sets are considered where 14 of them are lexicon based and the rest are tf-idf scoring, word2vec, dictionary of terms, query terms, and symbols' count. Each of these feature sets is discussed in detail in the following sections. Normalization as a technique of mapping data into equal ranges is discussed next. Finally, developed SVM model is briefly reviewed.

3.4 Pre-processing

The most basic pre-processing step in dealing with tweets is tokenization in order to split them into terms that can be scored later. Even though today different tokenization tools are available that benefit from a variety of techniques, white-space tokenization with some modifications is applied in this study. Using this technique on tweets, which often contain informal ways of writing and are inadvertently spaced, can result in meaningless tokens. However, ignoring them can result in missing important concepts. For example, consider the following tweet from the joy emotion data set:

"I WANNA GET UP AND DANCE!!!! (but everyone is in bed) this suks! Everyone wake up!! ???? #hyper #letsdance #dirtydancinginthemoonlight????"

In this tweet "wanna" is a careless way of writing "want to", although as long as it is a common practice, conveys the concept. Same issue also holds regarding miss spelled words such as "suks" instead of "sucks". Nevertheless, white space tokenization does not always work properly regarding combination of symbols and words. In a given tweet, white space tokenization returns "DANCE!!!!" as a single token that is not desirable for this study. Hence, the tokenization technique is modified using regular expressions (regex) to break a token into sub tokens properly. A collection of regular expression symbols such as [, (,), \n, ", _, -, ., !, ?, &,], +,), ?, and \s can return the desired outcome. For example in the above mentioned tweet, white space tokenization returns "DANCE!!!!" as a single token, while tokenizing with given regular expressions will split it into two tokens of "DANCE" and "!!!!", that can be used to study the effect of combination of symbols.

The decision to use white space tokenization is mainly due to two reasons. Section 2.5 introduced lexicons as a main feature source and one of the largest lexicons, NRC Affect Lexicon, is indeed developed over extracted terms from tweets and covers informal presentations. Thus, white space tokenization fits to this work better than other complex techniques. Punctuation marks and their combination are another reasons to prefer white space tokenization. As we will see later, number of exclamation or question marks is helpful in determining the level of emotion inferred from a tweet.

3.4.1 Setting Tweet Length

One of the challenges of working with tweets is deciding on a fixed tweet size. Although in practice tweet length cannot be more than a 280 characters, there are many tweets with shorter lengths. Moreover, tokenization results in different number of tokens regardless of number of characters. Hence the important issue is deciding on a fixed tweet length as passed data to SVM classifier should have equal number of features. Deciding on the best length is not a trivial task. Setting the length to a small value causes neglection of many terms, while large values are not preferable as zeros should be added to increase shorter tweets' length. Therefore, both of these actions can result in performance deterioration.

Model evaluation on a development data set before testing on the real test set is used to find the optimal tweet length. In the next chapter the optimal tweet length is determined by evaluating the performance of model on the development set with different tweets' lengths.

3.5 Feature Extraction

3.5.1 Affect Lexicons

Lexicon based scoring is basically a look up process that assigns a score to each token of a tweet that exists in a considered lexicon. Nevertheless, if a token does not exist in the lexicon its score sets to zero. List of considered feature sets for this study is given in Table 3.3. Each of the given lexicons is considered as a single feature set. The first nine feature sets ($f_1, \cdots, f_9$) are from the NRC affect lexicons set. Lexicons eight and nine are bi-gram and pair-gram versions of the sixth lexicon (f_6), respectively. In bi-gram lexicons the co-occurrence of two adjacent tokens is scored. However, pair-gram lexicons take into account the co-occurrence of two tokens in a tweet irrespective of their distance. Feature set ten (f_{10}) is WordNet lexicon [56] that was introduced earlier. Feature sets eleven through thirteen (f_{11}, f_{12}, f_{13}) are from Warriner et.al. lexicon set [62] that covers three domains of valence, arousal, and dominance. Last two lexicons used are from Bing Lui lexicon [53] that are originally composed of two sets: lexicon of words with positive opinion and lexicon of negative opinion words. However, in order to fit the data into a scoring structure, we split them into two versions of uni-grams and bi-grams, both covering positive and negative opinion words. Generally, the first three lexicons provide emotional based scores and the rest are sentiment based.

Tweet length in scoring with affect lexicons is a deterministic issue as it directly affects the feature vector size. Assuming l as the best tweet size, each lexicon extends the feature vector of tweets by length l. Moreover, recall that for tweets with less than l terms after tokenization, zeros are appended to increase the length.

Table 3.3: List of feature sets

Feature set ID	Description
f_1	NRC Affect Intensity Lexicon (4 emotions)
f_2	NRC Hashtag Emotion Lexicon-v0.2 (4 emotions)
f_3	NRC Emotion Lexicon Wordlevel-v0.92 (4 emotions)
f_4	NRC Hashtag Sentiment Lexicon-v1.0
f_5	NRC Hashtag-Sentiment-AffLexNegLex-v1.0
f_6	NRC Emoticon Lexicon-v1.0
f_7	Emoticon AFFLEX NEGLEX (uni-grams)
f_8	NRC Emoticon Lexicon-v1.0 (bi-gram)
f_9	NRC Emoticon Lexicon-v1.0 (pair-gram)
f_{10}	SentiWordNet 3.0
f_{11}	Warriner et al. Lexicon (valence)
f_{12}	Warriner et al. Lexicon (arousal)
f_{13}	Warriner et al. Lexicon (dominance)
f_{14}	Bing Liu Opinion Lexicon
f_{15}	Self-Dictionary (context based)
f_{16}	Query Terms
f_{17}	Symbols Effect
f_{18}	Tf-idf Scoring
f_{19}	Word2Vec

However, for a tweet with more than l terms, l randomly selected tokens form feature vector. Hence, by considering all lexicons, the length of the feature vector for each tweet equals $15 \times l$. For instance, considering again tweet given in section 3.4. By setting $l = 10$, feature set f_7 will generate vector $[0, 0, 0, 0, 0, 0, 0.161, 0.323, -0.503, 0.677]$, where 10 tokens are randomly chosen out of 26 tokens. If tweet had less than 10 tokens, trailing zeros would be attached to increase length to 10.

3.5.2 Tf-idf Score

Tf-idf feature set (f_{18}) evaluates whether the terms combination between different levels of an emotion is significantly different. By tokenization of the entire corpus, each tweet forms a vector of tokens. After aggregation of vectors on each level of emotion, four larger vectors of terms along with the number of tokens' occurrence is developed. Elements of these vectors as introduced in section 2.9 are referred as Term

Frequency (TF), and the number of tweets each term appears in shows the Document Frequency (DF). Combination of normalized Term Frequency (Eq. 2.2) and inverse Document Frequency (Eq. 2.3) into tf-idf, as given in Eq. 2.4, assigns a score to each term per class such that the summation of scores presents each tweet with a feature set of size 4, where summations determine the tweets class-wise score.

Considering the tweet given in section 3.4, after tokenization using regular expressions results in:

['I', 'WANNA', 'GET', 'UP', 'AND', 'DANCE', '!!!!', '(', 'but', 'everyone', 'is', 'in', 'bed', ')', 'this', 'suks', '!', 'Everyone', 'wake', 'up', '!!', '????', '#hyper', '#letsdance', '#dirtydancinginthemoonlight', '????']

where each term is represented by a vector with four tf-idf scores (one for each class). By adding assigned the scores, the feature vector for this tweet is:

$$[613.861,\ 397.982,\ 393.568,\ 537.379]^{1} \qquad \text{(Eq. 3.2)}$$

where the feature vector size equals $(15 \times l)+4$.

3.5.3 Word2vec

Another feature set used is word2vec, a Python library that is used for constructing the model in this study. Gensim word2vec initially receives a list of training texts and generates a Continuous Bag of Words (CBoW) to learn a model over co-occurrence of terms and scores them on their similarity. Although there are already trained models available online that can be simply applied to score test samples, in this study we developed our own models (one for each level of emotion) using the train data. In model construction the size of the representative vector (i.e. size of the surrounding

1 Values are rounded to 3 digits.

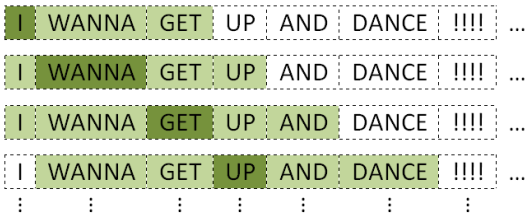


Figure 3.2: Presentation of word2vec window size

context) and the window size are set to 400 and 2, respectively (same settings as applied by participant teams in the competition). Window size defines the maximum distance between the target term and its neighboring words. Thus, with a window size of 2, for each term two terms to the right and to the left are considered (Figure 3.2).

After model construction, a list of the most similar terms to the target term along with their similarity measures can be retrieved. Summation of tokens' similarity scores with terms in their window is taken into account as the tweet's feature. Hence, by having one model per class, each tweet has a vector with four elements that is appended to the feature vector from tf-idf and affect lexicon scoring and increases vector size to $(15 \times l)+4+4$.

3.5.4 Context Based Dictionary

One of the possible determinative features in detecting levels of an emotion is variation in the distribution of specific tokens for different levels of that emotion. To evaluate the effectiveness of terms distribution, four dictionaries per emotion are developed using train data set (one for each level) and all of the terms appeared in a level are added into their corresponding dictionary along with the total number of their occurrence. We call this set a "self-dictionary". Table 3.4 shows part of the developed dictionary for anger emotion. Self-dictionary utilization is similar to

using lexicons. Scores of tokens or zeros[2] form a vector. Hence, each tweet has a feature vector of size $4 \times l$. With respect to a self-dictionary, the percentage that is looked up for a token is a deterministic parameter. Basically, a self-dictionary contains all the terms that exist during training. However, some terms exist rarely and ignoring them can promote more effective terms. Therefore, top $p\%$ of terms after sorting them from the most common to the rarest ones, is kept. Deciding on the optimal percentage (p) is similar to the best tweet length determination. Common practice is to check the dictionary performance for different percentages using the development data set or applying cross-validation technique. Finally, the self-dictionary feature set vector similar to other feature sets appends to the previous feature vector which increases the feature dimension to $(15 \times l)+4+4+(4 \times l)$.

3.5.5 Query Terms

Earlier in section 3.2, query terms were discussed as a list of related terms to an emotion and were used to retrieve tweets using twitter API. Here the idea is to use these terms to test whether a query term can discriminate between different levels of

Table 3.4: Part of developed self-dictionary for different levels of anger

Level 0		**Level 1**		**Level 2**		**Level 3**	
term	**score**	**term**	**score**	**term**	**score**	**term**	**score**
the	177	the	157	the	207	the	159
a	136	to	111	to	179	to	159
I	128	and	100	a	158	I	149
to	127	a	98	I	153	and	131
and	102	I	83	and	136	a	127
is	81	of	67	is	109	you	95
you	78	is	64	of	109	is	95
?	73	you	59	?	84	me	89
of	72	in	56	you	78	my	74
in	70	it	49	_	78	that	73
my	60	that	47	my	74	of	70

2 Depends on the existence of a token in the corresponding dictionary.

inferred emotion from a tweet since it was effective in selecting the set of relevant tweets. To examine this phenomenon during the training phase, tweets are checked after tokenization for appearance of query terms. If a query term appears in it, the emotional level of that tweet appends to the corresponding vector of that query term. Thus, each query term has a vector of levels that it showed up in. This vector is then put into a vector with four elements where each element shows the percentage of times the query term belongs to each level. The following example shows the level-wise occurrence vector of term "sparkling", a query term from emotion joy (Eq. 3.3). Based on this vector term "sparkling" appeared in two tweets with level '2' of joy, or in nine tweets with level '1'. The percentage of its occurrence in each level is equal to the vector on the right side of equation.

$$\begin{aligned} \text{Sparkling: [} & \text{'2', '0', '3', '0', '0', '1', '0', '1', '1', '1',} \\ & \text{'1', '3', '0', '0', '0', '0', '0', '0', '2', '1',} \\ & \text{'1', '0', '0', '1', '0', '0', '1', '0', '0', '0']} \\ & \equiv [0.567, 0.3, 0.067, 0.067] \end{aligned} \tag{Eq. 3.3}$$

In model development and model testing, each tweet is assigned a four-element vector (one element per class) as a feature set where each element keeps the highest score in corresponding level among all occurred query terms in that tweet. Aggregation of query terms feature set with already extracted features, increases vector dimensionality by 4 which equals $(15 \times l)+4+4+(4 \times l)+4$.

3.5.6 Symbol Effect

It is expected that the usage of certain symbols such as "!" or "?" has association with the intensity of emotion the speaker is experiencing. Table 3.5 shows entries of the sixth feature set (NRC Emoticon Lexicon-v1.0). Comparing entries and their

Table 3.5: Sample entries from lexicon NRC Emoticon Lexicon-v1.0.

Term	Score	N.Pos	N.Neg
???	-0.826	5	12
???!	-0.056	9	10
???!!	0.86	9	4
???!!!	-0.521	13	23
???!!!!	-0.356	4	6
.!!	0.392	31	22
.!!!	-0.151	18	22
??!!	-0.188	56	71
??!!!	-0.526	9	16
??!?	0.742	6	3
!!!?	-1.491	3	14

scores recommends symbols combination and their count as a determinative factor. Although such differences may not seen relevant, lexicons that are built over collected data from Twitter reveals various combinations of such symbols practiced by individuals to convey different levels of emotions. For instance, "!!!?" has negative emotion since 14 out 17 times, it has been observed in tweets with negative sentiments (N.Pos). Hence, combination and number of question and exclamation marks (i.e. two widely used symbols) are considered as a new feature set. This feature set assigns each tweet a vector with four elements, where the first and third elements represent the number of tokens with exclamation and question marks, respectively. The second and fourth elements of the vector keep respectively the total number of exclamation and question marks in the entire tweet. For the given tweet in section 3.4, the feature set vector equals to [3.0, 2.0, 7.0, 8.0]. By appending this features set to already extracted features, the new dimension will be $(15 \times l)+4+4+(4 \times l)+4+4$.

3.6 Normalization

Standardization is one of the normalization techniques used in data preprocessing, which is briefly discussed in section 2.8. Although generally there is no guarantee

that normalization improves performance, it can reduce both training time and estimation error. Regarding SVM classifier, considering that it uses combination of features and creates a hyperplane to separate classes, data should not be skewed too much. For this reason, the extracted train feature set is standardized and mapped into a new range with mean 0 and standard deviation 1 before being used to train the model. Normalization is the last step before model construction. Following normalization, a model which uses every single feature set is developed to test feature set performance independently in classification. Besides, separate models for combination of lexicons and combination of all feature sets are constructed. For model construction, LinearSVC, from SVM library of Scikit-learn software [104] is used in the "one vs. rest" mode. Most of the parameters of this function are kept as default values except for tolerance, random-state and maximum iterations. Tolerance which defines the stopping criteria (tol) is set to 10^{-6}. Random state (random_state) used to determine the seed of the pseudo random number generator is set to "None" which means that the system selects seeds randomly; and, finally maximum iteration (max_iter) set to 10^5 to make sure that the training model converges. The remaining of model parameters are set as follows:

LinearSVC(penalty='l2', loss='squared_hinge', dual=True, tol=0.000001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=100000)

We use training data to develop the models for classification. However, achieving the best performance requires optimizing the parameters such as tweet length and using an effective size of self-dictionary. The trained models are validated then with the development data for parameters optimization as well as evaluating the effectiveness

of feature sets before testing the models on actual test data. Thus, the models' performances are checked over extracted features out of the development samples and by modifying parameters in each iteration, the optimal values are determined.

Chapter 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The fundamentals of an automatic emotion detection system were discussed extensively in Chapter 3. In this chapter for each of anger, joy, fear, and sadness emotions construction of models using every feature set in Table 3.3 is discussed. This means development of four sets of 19 classifiers. The levels or classes of emotion intensities also ranges between 0 and 3 where 0 stands for the lowest and 3 for the highest level of inferred emotion.

Model construction is done using the train and development (validation) sets, first to determine the best percentage of self-dictionaries to be used for different emotions, and then to determine the optimal tweet length by using the combination of all feature sets. Throughout the study, precision, recall, and F-score are the metrics used and to compare the performance of classifiers micro and macro averages over all intensity levels are calculated. Section 4.3 considers model development with optimal parameters on different feature sets where combinations and performances are compared. Finally, to improve results, feature and classifier selection techniques are applied. Throughout this work, train and development data sets are used to train, optimize, and validate the models and the test data set is used to compare the performance of classifiers and their combinations.

4.2 Choosing Self-Dictionary Size and Optimal Tweet Length

The idea of developing dictionaries is to test whether terms and their frequencies can effectively help in deciding on the intensity level of the inferred emotion in a tweet. Table 4.1 shows the total number of entries for every emotion. Remember that we have seen in Table 3.4 that majority of the terms occur rarely and are common among different intensity levels. Therefore, pruning the dictionary and trying to keep discriminative entries may increase performance.

As discussed in Section 3.5, each tweet is assigned four vectors of scores of size l. It was also mentioned that the optimal tweet length (l) is not known and performance of feature sets such as lexicons is directly affected by the tweet length. Therefore, the best tweet length has to be set in accordance with the performance of combination of all feature sets for which optimal percentage of self-dictionary is needed. To choose the optimal dictionary size the initial value of l for each emotion is set as the average length of tweets. After calculation of best dictionary size, the optimal lengths are found by training and validating the classifiers on combination of all feature sets. Table 4.1 provides basic statistics on the tweets length for train and development data sets. By setting l to 19 (the average length), series of SVM models are trained using the train data set with different percentages of dictionaries, ranging

Table 4.1: Dictionary sizes and tweet lengths

	Dictionary size				**Tweet length**					
	intensity level				**train**			**validation**		
	0	**1**	**2**	**3**	**max.**	**min.**	**avg.**	**max.**	**min.**	**avg.**
Anger	2619	2340	3281	2368	61	1	18.51	41	2	18.89
Joy	3327	2285	2151	1973	41	2	18.59	37	2	18.56
Fear	6696	2337	1717	1223	43	2	18.67	41	1	18.83
Sadness	3819	2054	2595	1772	159	2	19.26	41	2	18.64

from 5 to 95 percent. Models are then tested using the development data set to check classifiers performance for each percentage. Figure 4.1 shows fluctuations of micro F-scores for each emotion. Comparing plots, reveals that same ratios do not work well for all emotions. Therefore, a different dictionary size should be decided for each emotion. Deciding on the best fraction is merely based on the F-score of trained models on f_{15}. For the anger emotion, the developed classifier by considering 70% of dictionary and F-score of 0.343 outperforms other models. Classifiers for joy, fear, and sadness emotions achieve their best performance on 35%, 20%, and 35% of their corresponding dictionary respectively with F-scores equal 0.276, 0.648, and 0.418 (Table 4.2).

Figure 4.1 also shows variations in micro F-scores when all features are used in combination. It can be seen that similar fluctuations occur when the classifier is trained only with the self-dictionary feature set (f_{15}). Unlike self-dictionary, performance of nearly all feature sets (15 out of 19) is under effect of tweet length. Therefore, the best tweet length is computed based on the performance of the classifiers using all feature sets combination. Since the performance of the SVM classifier may slightly change with the use of new seeds as a part of the random number generator used, model training and validating are repeated 10 times and average of results is considered.

Table 4.2: Models' optimal parameters

	Anger	**Joy**	**Fear**	**Sadness**
opt. percentage	70	35	20	35
opt. length	33	33	36	27
micro F-score	0.343	0.276	0.648	0.418

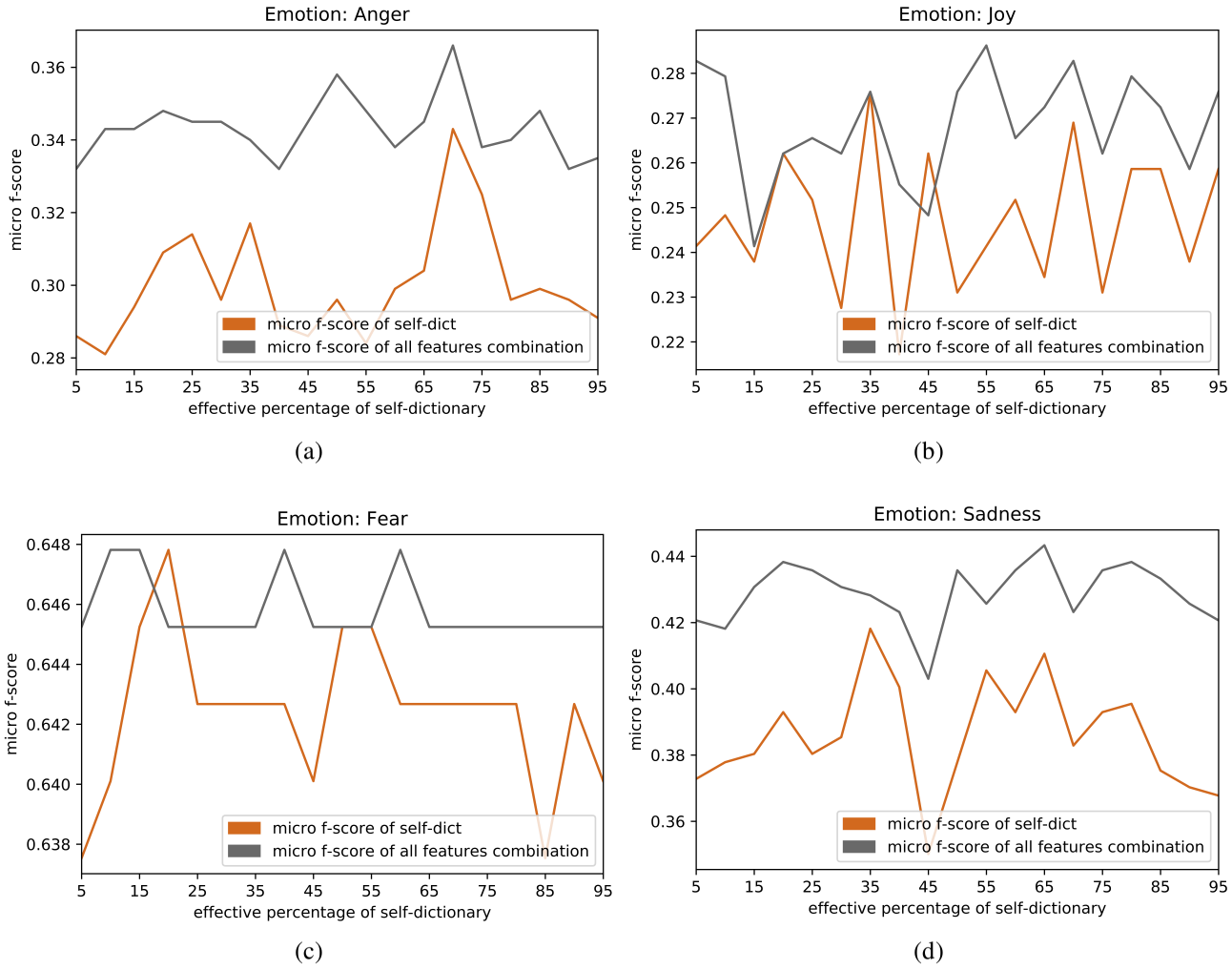


Figure 4.1: Classification performance using different self-dictionary sizes

Figure 4.2 shows averaged micro F-scores of the trained SVM classifiers using different tweet lengths ranging from 3 to 40 terms. Since the lengths considered should be reasonable for majority of the tweets, generality and inclusiveness are reasons for minimum and maximum values of length, respectively. For example the longest tweet for sadness emotion has 159 terms while the second longest one has only 43 terms which makes the former an outlier. 95^{th} percentile of tweets length, the point that indicates 95 percent of tweets have fewer terms than, is shown in Figure 4.2 with dashed red (R) and green (S) lines respectively for train and development sets.

Comparing plots and micro F-score curves reveal that the best length varies through emotions. Therefore, similar to the best percentage of self-dictionaries, each emotion has its optimal tweet length. For anger emotion, the highest F-score of trained model on combination of all features occurs at length 33. For joy, fear, and sadness emotions optimal lengths are respectively 33, 36, and 27 which are marked with blue rectangles. Furthermore, micro F-scores on the optimal lengths for anger, joy, fear, and sadness emotions are 0.373, 0.281, 0.648, and 0.443 (Table 4.2). Closeness of the optimal lengths to 95^{th} percentile shows that consideration of almost full tweet length in training good classifiers is more preferable than dropping terms.

In Figure 4.2, variations in the average micro F-score of combination of all lexicons (i.e. first 14 feature sets) are displayed. For almost all emotions there is a gap between all lexicons curve and the curve for all features. This gap suggests that subset of feature sets can work better than the combination of all feature sets as discussed later in this chapter.

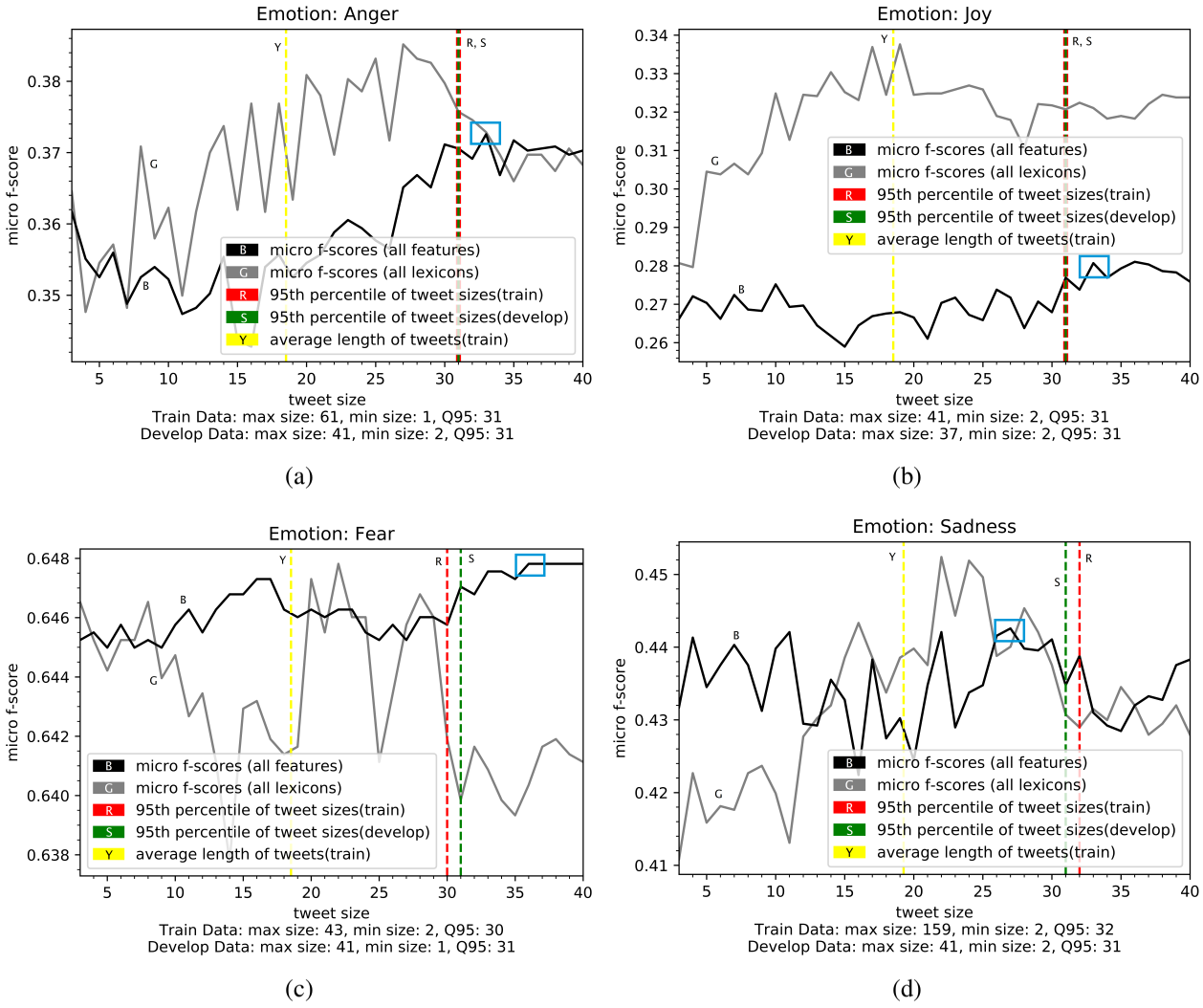


Figure 4.2: Performance of models using different tweet lengths

4.3 Classifier Construction

The 19 feature sets used in this study were introduced in Chapter 3. A classifier is trained for each feature set to measure the success in classifying tweets into four different intensity levels. Table 4.3 shows micro F-score of trained models that are validated on the development set. Scores under the first column of each emotion are performance of classifiers with the optimal tweet length, while the next two columns show scores when models are trained on average and maximum tweet lengths. Maximum and average lengths are studied as benchmarks for later comparisons. Figure 4.3 confirms that deciding to continue with the optimal length, in comparison with the average and maximum lengths, is a valid decision for majority of trained classifiers.

Comparing scores under three different lengths shows that classifiers that are developed using feature sets such as f_{17} and f_{18} have micro F-scores independent of tweet length. For the rest of feature sets, comparing micro F-scores for different tweet sizes reveals that neither of the lengths used improve performance for all classifiers. Nevertheless, when all feature sets are used in combination, the best performance is achieved for optimal length, l.

Feature sets 11 to 13 in Table 4.3, as introduced in Table 3.3, belong to the same lexicon set that covers three different dimensions of emotions i.e. valance, arousal, and dominance. It is expected that their combination performance will be better than their individual result. However, classifiers generated using their combination in comparison to individual ones show improvement only in classification for joy emotion with F-score 0.255. For anger, fear, and sadness emotion the performances

Table 4.3: Micro F-scores of trained classifiers on different tweet lengths

tweet length	Anger (self-dict.: 70%)			Joy (self-dict.: 35%)			Fear (self-dict.: 20%)			Sadness (self-dict.: 35%)			Avg. on opt. len.
	33 (opt.)	**19 (avg.)**	**61 (max.)**	**33 (opt.)**	**19 (avg.)**	**41 (max.)**	**36 (opt.)**	**19 (avg.)**	**43 (max.)**	**27 (opt.)**	**19 (avg.)**	**44 (max.)**	
f_1	0.415	0.399	0.415	0.224	0.217	0.228	0.640	0.645	0.640	0.433	0.431	0.428	0.428
f_2	0.389	0.340	0.389	0.241	0.248	0.252	0.643	0.645	0.643	0.428	0.426	0.411	0.425
f_3	**0.433**	0.438	0.430	0.207	0.172	0.203	0.640	0.645	0.640	0.416	0.421	0.413	0.424
f_4	0.376	0.369	0.394	0.290	0.286	0.307	0.645	0.645	0.645	0.436	0.421	0.408	0.437
f_5	0.366	0.376	0.369	**0.307**	0.307	0.293	0.645	0.645	0.645	0.428	0.426	0.421	0.437
f_6	0.348	0.366	0.345	0.252	0.252	0.255	0.643	0.645	0.645	0.380	0.408	0.360	0.406
f_7	0.348	0.345	0.348	0.272	0.269	0.269	0.648	0.645	0.648	0.431	0.418	0.398	0.425
f_8	0.348	0.353	0.345	0.272	0.300	0.286	0.645	0.645	0.648	0.406	0.431	0.406	0.418
f_9	0.374	0.345	0.376	0.241	0.259	0.252	0.645	0.645	0.643	0.398	0.395	0.403	0.415
f_{10}	0.376	0.330	0.366	0.238	0.248	0.238	0.645	0.645	0.645	0.438	0.428	0.416	0.424
f_{11}	0.307	0.307	0.299	0.214	0.228	0.210	0.645	0.645	0.645	0.411	0.398	0.383	0.394
f_{12}	0.320	0.320	0.330	0.224	0.221	0.224	0.645	0.645	0.645	0.393	0.411	0.393	0.396
f_{13}	0.309	0.304	0.317	0.210	0.221	0.224	0.645	0.645	0.645	0.408	0.403	0.388	0.393
f_{14}	0.356	0.387	0.371	0.231	0.221	0.238	0.643	0.645	0.643	0.426	0.431	0.426	0.414
f_{15}	0.335	0.317	0.338	0.272	0.272	0.272	0.645	0.638	0.645	0.403	0.393	0.416	0.414
f_{16}	0.222	0.222	0.222	0.186	0.186	0.186	**0.650**	0.650	0.650	**0.461**	0.461	0.461	0.380
f_{17}	0.307	0.307	0.307	0.207	0.207	0.207	0.645	0.645	0.645	0.428	0.428	0.428	0.397
f_{18}	0.363	0.363	0.363	0.266	0.266	0.266	0.645	0.645	0.645	0.443	0.443	0.443	0.429
f_{19}	0.374	0.361	0.361	0.266	0.262	0.255	0.645	0.645	0.645	0.355	0.363	0.363	0.410
f_{11}, f_{12}, f_{13}	0.320	0.325	0.335	0.255	0.255	0.248	0.640	0.645	0.64	0.378	0.390	0.368	0.398
f_1 to f_{14} (all lexicons)	0.371	0.361	0.366	0.324	0.307	0.324	0.638	0.645	0.643	0.446	0.433	0.428	0.445
all features	**0.379**	**0.335**	**0.371**	**0.283**	**0.255**	**0.276**	**0.648**	**0.648**	**0.648**	**0.446**	**0.431**	**0.438**	**0.439**

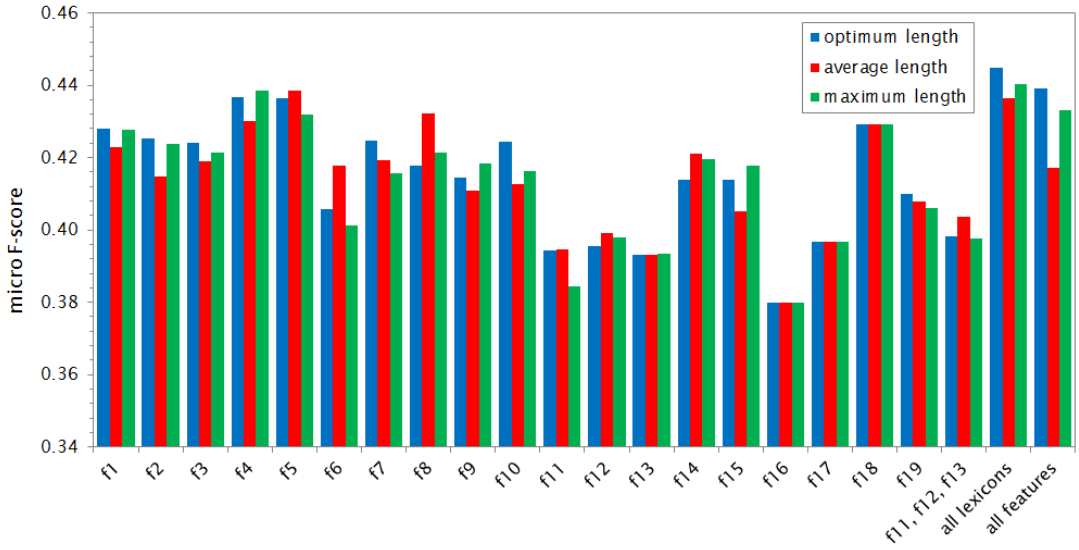


Figure 4.3: Average of micro F-scores of trained classifiers over all emotions

are either the same or worse. Combination of all lexicons, a subset of all feature sets combination, is judged as a single classifier as well and for an emotion such as joy, outperforms the combination of all feature sets with 14% lager F-score.

Table 4.4 reports macro F-scores for classifiers developed. By comparing results, it is clear that the trained classifiers have better performance with optimum tweet lengths on all emotions (except for fear which is almost negligible). Moreover, macro F-scores are significantly smaller than micro F-scores. This is due to the fact that in micro average contribution of all classes is considered. Hence, when a classifier performs very well for a particular class, the micro F-score is important. Macro average F-score, in contrast, gives a better understanding of a classifier's performance on average by considering equal contribution for all classes. Therefore, if the classifier performs poorly for a particular class, this reduces the average considerably. For example, majority of the trained classifiers for fear emotion failed to predict samples from levels 1 and 2 (Table 4.5). However, they were widely

Table 4.4: Macro F-scores of trained classifiers on different tweet lengths

	Anger (self-dict.: 70%)			**Joy (self-dict.: 35%)**			**Fear (self-dict.: 20%)**			**Sadness (self-dict.: 35%)**		
tweet length	**33 (opt.)**	**19 (avg.)**	**61 (max.)**	**33 (opt.)**	**19 (avg.)**	**41 (max.)**	**36 (opt.**	**19 (avg.)**	**43 (max.)**	**27 (opt.)**	**19 (avg.)**	**44 (max.)**
f_1	0.276	0.261	0.275	0.207	0.195	0.210	0.196	0.196	0.196	0.264	0.248	0.269
f_2	0.288	0.261	0.283	0.212	0.213	0.216	0.196	0.196	0.196	0.258	0.258	0.244
f_3	0.269	0.271	0.266	0.171	0.138	0.168	0.196	0.196	0.196	0.222	0.244	0.256
f_4	0.282	0.327	0.290	0.262	0.254	0.281	0.196	0.196	0.196	0.287	0.241	0.269
f_5	0.276	0.327	0.278	0.290	0.284	0.271	0.196	0.196	0.196	0.264	0.286	0.277
f_6	0.273	0.294	0.262	0.223	0.228	0.229	0.196	0.196	0.196	0.243	0.272	0.226
f_7	0.268	0.300	0.270	0.248	0.246	0.243	0.216	0.196	0.216	0.292	0.264	0.247
f_8	0.270	0.296	0.276	0.252	0.281	0.268	0.196	0.196	0.205	0.257	0.279	0.252
f_9	0.272	0.241	0.277	0.215	0.244	0.226	0.196	0.196	0.196	0.239	0.242	0.261
f_{10}	0.265	0.246	0.259	0.219	0.212	0.221	0.196	0.196	0.196	0.224	0.210	0.213
f_{11}	0.227	0.261	0.224	0.188	0.202	0.187	0.196	0.196	0.196	0.221	0.169	0.195
f_{12}	0.240	0.249	0.247	0.199	0.187	0.200	0.196	0.196	0.196	0.208	0.176	0.211
f_{13}	0.236	0.253	0.243	0.178	0.190	0.200	0.196	0.196	0.196	0.210	0.185	0.204
f_{14}	0.252	0.327	0.271	0.187	0.158	0.195	0.204	0.196	0.204	0.272	0.238	0.266
f_{15}	0.281	0.318	0.283	0.235	0.247	0.235	0.196	0.195	0.196	0.263	0.258	0.269
f_{16}	0.180	0.289	0.180	0.114	0.114	0.114	0.272	0.272	0.272	0.273	0.273	0.273
f_{17}	0.214	0.348	0.214	0.151	0.151	0.151	0.196	0.196	0.196	0.150	0.150	0.150
f_{18}	0.310	0.299	0.310	0.240	0.240	0.240	0.215	0.215	0.215	0.331	0.331	0.331
f_{19}	0.291	0.316	0.289	0.249	0.242	0.232	0.196	0.196	0.196	0.253	0.276	0.276
f_{11}, f_{12}, f_{13}	0.250	0.297	0.263	0.228	0.236	0.222	0.196	0.196	0.196	0.246	0.236	0.219
f_1 to f_{14} (all lexicons)	0.295	0.278	0.29	0.314	0.290	0.314	0.252	0.271	0.268	0.338	0.321	0.324
all features	**0.330**	**0.280**	**0.324**	**0.260**	**0.241**	**0.251**	**0.224**	**0.225**	**0.224**	**0.325**	**0.316**	**0.324**

successful in detection of class 0 entities. Therefore, macro average, by giving equal share to all classes has smaller value than micro average, which highlights the performance of the classifier model for class 0. Table 4.5 shows detailed information for the trained classifiers performance using individual feature sets and the optimal parameters. Considering the first 14 feature sets, for the anger emotion, classifier of f_6 achieves the highest precision score on average for detection of classes when macro precision value equals 0.379. However, classifier of f_5 performs the best in terms of recall with macro recall value equals 0.314. Regarding F-score, classifier trained using f_3 has the best performance with micro F-score equals 0.433. On the other hand, classifier of f_2 performs the highest average of F-scores over different classes.

Among the rest of single feature sets, classifier of f_{16} has the highest macro precision and classifier using f_{18} has the highest macro recall and F-score. The last rows of the table give dictionary sizes and effective percentages in number of terms as well as the number of training and development samples for each intensity level.

Overview of results show that classification of emotions under the second and the third levels of intensity is not as easy as the first and the fourth levels since majority of zeros belong to these two levels . For the anger emotion, scores show that almost all of the considered feature sets, when they are used individually, fail in training classifiers that can predict the first level of intensity. Furthermore, none of the feature sets can train classifiers to classify correctly levels 1, 2, and 3 for the fear emotion. Through the rest of the emotions, classifier of f_5 for joy and classifier of f_{16} for fear and sadness surpass other single feature sets' classifiers with highest micro F-scores.

Table 4.5: Precision, recall, and F-score values for all trained classifiers tested on development data set

	Anger emotion (optimal length = 33, self-dictionary percentage = 70%)														
	precision					**recall**					**F-score**				
intensity levels	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**
f_1	0.543	0	0.305	0.183	0.258	0.640	0	0.299	0.255	0.298	0.588	0	0.302	0.213	0.276
f_2	0.560	0.083	0.312	0.194	0.287	0.500	0.019	0.443	0.275	0.309	0.528	0.030	0.366	0.228	**0.288**
f_3	0.516	0	0.343	0.167	0.257	0.688	0	0.351	0.118	0.289	0.590	0	0.347	0.138	0.269
f_4	0.662	0.250	0.263	0.168	0.336	0.473	0.019	0.423	0.314	0.307	0.552	0.034	0.324	0.219	0.282
f_5	0.623	0	0.312	0.170	0.276	0.409	0	0.495	0.353	**0.314**	0.494	0	0.382	0.229	0.276
f_6	0.591	0.500	0.259	0.168	**0.379**	0.435	0.037	0.309	0.431	0.303	0.502	0.069	0.282	0.242	0.273
f_7	0.577	0.333	0.278	0.147	0.334	0.462	0.037	0.278	0.392	0.292	0.513	0.067	0.278	0.214	0.268
f_8	0.545	0.500	0.246	0.209	0.375	0.425	0.019	0.320	0.471	0.308	0.477	0.036	0.278	0.289	0.270
f_9	0.571	0	0.307	0.162	0.260	0.478	0	0.402	0.333	0.303	0.520	0	0.348	0.218	0.272
f_{10}	0.539	0	0.321	0.167	0.257	0.409	0	0.629	0.176	0.303	0.465	0	0.425	0.171	0.265
f_{11}	0.492	0	0.260	0.143	0.224	0.344	0	0.454	0.216	0.253	0.405	0	0.331	0.172	0.227
f_{12}	0.500	0	0.260	0.179	0.235	0.339	0	0.485	0.275	0.274	0.404	0	0.338	0.217	0.240
f_{13}	0.470	0	0.244	0.203	0.229	0.339	0	0.443	0.275	0.264	0.394	0	0.315	0.233	0.236
f_{14}	0.535	0	0.246	0.170	0.238	0.495	0	0.320	0.294	0.277	0.514	0	0.278	0.216	0.252
f_{15}	0.518	0.200	0.297	0.208	0.306	0.312	0.056	0.505	0.392	0.316	0.389	0.087	0.374	0.272	0.281
f_{16}	0.818	0.167	0.378	0.159	**0.380**	0.097	0.019	0.175	0.980	0.318	0.173	0.033	0.239	0.273	0.180
f_{17}	0.596	0	0.271	0.200	0.267	0.167	0	0.814	0.176	0.289	0.261	0	0.407	0.188	0.214
f_{18}	0.551	0.136	0.278	0.295	0.315	0.376	0.056	0.412	0.549	**0.348**	0.447	0.079	0.332	0.384	**0.310**
f_{19}	0.609	0.154	0.319	0.133	0.304	0.495	0.074	0.371	0.255	0.299	0.546	0.100	0.343	0.174	0.291
f_{11}, f_{12}, f_{13}	0.503	0.133	0.252	0.161	0.262	0.387	0.037	0.371	0.275	0.267	0.438	0.058	0.300	0.203	0.250
f_1 to f_{14} (all lexicons)	0.614	0.167	0.298	0.174	0.313	0.462	0.056	0.371	0.373	0.315	0.528	0.083	0.330	0.238	0.295
all features	0.606	0.235	0.298	0.288	0.357	0.355	0.074	0.464	0.627	0.380	0.447	0.113	0.363	0.395	0.330

					total
size of dictionary	**2619**	**2340**	**3281**	**2368**	
no. of taken terms	**1840**	**1638**	**2297**	**1658**	
no. of development samples	**186**	**54**	**97**	**51**	**388**
no. of train samples	**445**	**322**	**507**	**427**	**1701**

Table 4.5 (cont.): Precision, recall, and F-score values for all trained classifiers tested on development data set

	Joy emotion (optimal length = 33, self-dictionary percentage = 35%)														
	precision					**recall**					**F-score**				
intensity levels	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**
f_1	0.184	0.286	0.414	0.222	0.276	0.636	0.105	0.135	0.157	0.258	0.286	0.154	0.203	0.184	0.207
f_2	0.213	0.250	0.500	0.281	0.311	0.764	0.063	0.067	0.314	0.302	0.333	0.101	0.119	0.296	0.212
f_3	0.176	0.286	0.625	0.286	0.343	0.745	0.021	0.056	0.235	0.264	0.285	0.039	0.103	0.258	0.171
f_4	0.275	0.353	0.321	0.291	0.310	0.709	0.063	0.101	0.588	0.365	0.396	0.107	0.154	0.390	0.262
f_5	0.282	0.348	0.425	0.282	0.334	0.727	0.084	0.191	0.471	0.368	0.406	0.136	0.264	0.353	0.290
f_6	0.233	0.412	0.368	0.182	0.299	0.764	0.074	0.157	0.196	0.298	0.357	0.125	0.220	0.189	0.223
f_7	0.243	0.444	0.400	0.233	0.330	0.782	0.084	0.157	0.275	0.324	0.371	0.142	0.226	0.252	0.248
f_8	0.261	0.282	0.357	0.258	0.290	0.764	0.116	0.112	0.314	0.326	0.389	0.164	0.171	0.283	0.252
f_9	0.215	0.333	0.333	0.244	0.281	0.745	0.063	0.135	0.216	0.290	0.333	0.106	0.192	0.229	0.215
f_{10}	0.211	0.303	0.367	0.212	0.273	0.673	0.105	0.124	0.216	0.279	0.322	0.156	0.185	0.214	0.219
f_{11}	0.211	0.387	0.333	0.136	0.267	0.636	0.126	0.045	0.216	0.256	0.317	0.190	0.079	0.167	0.188
f_{12}	0.222	0.292	0.267	0.169	0.237	0.655	0.147	0.045	0.216	0.266	0.332	0.196	0.077	0.190	0.199
f_{13}	0.213	0.311	0.100	0.152	0.194	0.655	0.147	0.011	0.196	0.252	0.321	0.200	0.020	0.171	0.178
f_{14}	0.223	0.160	0.429	0.277	0.272	0.855	0.042	0.034	0.255	0.296	0.353	0.067	0.063	0.265	0.187
f_{15}	0.243	0.400	0.353	0.305	0.325	0.800	0.042	0.067	0.490	0.350	0.373	0.076	0.113	0.376	0.235
f_{16}	0.250	0	0.200	0.182	0.158	0.073	0	0.022	0.941	0.259	0.113	0	0.040	0.305	0.114
f_{17}	0.209	0	0	0.205	0.103	0.600	0	0	0.529	0.282	0.310	0	0	0.295	0.151
f_{18}	0.248	0.278	0.267	0.287	0.27	0.636	0.053	0.090	0.569	0.337	0.357	0.088	0.134	0.382	0.240
f_{19}	0.241	0.318	0.306	0.274	0.285	0.600	0.074	0.124	0.510	0.327	0.344	0.120	0.176	0.356	0.249
f_{11}, f_{12}, f_{13}	0.276	0.286	0.278	0.207	0.261	0.727	0.105	0.056	0.373	0.315	0.400	0.154	0.093	0.266	0.228
f_1 to f_{14} (all lexicons)	0.275	0.344	0.561	0.280	0.365	0.709	0.116	0.258	0.412	0.374	0.396	0.173	0.354	0.333	0.314
all features	0.255	0.400	0.324	0.288	0.317	0.636	0.063	0.124	0.588	0.353	0.365	0.109	0.179	0.387	0.260

					total
size of dictionary	**3327**	**2285**	**2151**	**1973**	
no. of taken terms	**1164**	**800**	**753**	**691**	
no. of development samples	**55**	**95**	**89**	**51**	**290**
no. of train samples	**363**	**346**	**359**	**548**	**1616**

Table 4.5 (cont.): Precision, recall, and F-score values for all trained classifiers tested on development data set

	Fear emotion (optimal length = 36, self-dictionary percentage = 20%)													
	precision					**recall**					**F-score**				
intensity levels	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**
f_1	0.648	0	0	0	0.162	0.992	0	0	0	0.248	0.784	0	0	0	0.196
f_2	0.644	0	0	0	0.161	0.996	0	0	0	0.249	0.782	0	0	0	0.196
f_3	0.650	0	0	0	0.163	0.992	0	0	0	0.248	0.785	0	0	0	0.196
f_4	0.645	0	0	0	0.161	1	0	0	0	0.250	0.784	0	0	0	0.196
f_5	0.645	0	0	0	0.161	1	0	0	0	0.250	0.784	0	0	0	0.196
f_6	0.644	0	0	0	0.161	0.996	0	0	0	0.249	0.782	0	0	0	0.196
f_7	0.647	0	0	1	0.412	1	0	0	0.042	0.260	0.786	0	0	0.080	0.216
f_8	0.645	0	0	0	0.161	1	0	0	0	0.250	0.784	0	0	0	0.196
f_9	0.645	0	0	0	0.161	1	0	0	0	0.250	0.784	0	0	0	0.196
f_{10}	0.647	0	0	0	0.162	1	0	0	0	0.250	0.786	0	0	0	0.196
f_{11}	0.645	0	0	0	0.161	1	0	0	0	0.250	0.784	0	0	0	0.196
f_{12}	0.645	0	0	0	0.161	1	0	0	0	0.250	0.784	0	0	0	0.196
f_{13}	0.645	0	0	0	0.161	1	0	0	0	0.250	0.784	0	0	0	0.196
f_{14}	0.647	0.500	0	0	0.287	0.992	0.018	0	0	0.252	0.783	0.034	0	0	0.204
f_{15}	0.647	0	0	0	0.162	1	0	0	0	0.250	0.786	0	0	0	0.196
f_{16}	0.661	0.125	0	0.667	0.363	0.988	0.018	0	0.167	0.293	0.792	0.031	0	0.267	0.272
f_{17}	0.645	0	0	0	0.161	1	0	0	0	0.250	0.784	0	0	0	0.196
f_{18}	0.649	0	0	0.333	0.246	0.996	0	0	0.042	0.259	0.786	0	0	0.074	0.215
f_{19}	0.645	0	0	0	0.161	1	0	0	0	0.250	0.784	0	0	0	0.196
f_{11}, f_{12}, f_{13}	0.647	0	0	0	0.162	0.992	0	0	0	0.248	0.783	0	0	0	0.196
f_1 to f_{14} (all lexicons)	0.670	0.118	0.500	0.200	0.372	0.964	0.035	0.053	0.042	0.273	0.791	0.054	0.095	0.069	0.252
all features	0.649	1	0	0.500	0.537	0.996	0.018	0	0.042	0.264	0.786	0.034	0	0.077	0.224

					total
size of dictionary	**6696**	**2337**	**1717**	**1223**	
no. of taken terms	**1339**	**467**	**343**	**245**	
no. of development samples	**251**	**57**	**57**	**24**	**389**
no. of train samples	**320**	**249**	**193**	**1490**	**2252**

Table 4.5 (cont.): Precision, recall, and F-score values for all trained classifiers tested on development data set

	Sadness emotion (optimal length = 27, self-dictionary percentage = 35%)														
	precision					**recall**					**F-score**				
intensity levels	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**
f_1	0.482	0.429	0.381	0.140	0.358	0.865	0.034	0.184	0.115	0.300	0.619	0.063	0.248	0.126	0.264
f_2	0.490	0	0.250	0.234	0.243	0.859	0	0.149	0.212	0.305	0.624	0	0.187	0.222	0.258
f_3	0.452	0.308	0.375	0.071	0.301	0.882	0.045	0.103	0.038	0.267	0.598	0.079	0.162	0.050	0.222
f_4	0.460	0	0.371	0.333	0.291	0.812	0	0.299	0.173	0.321	0.587	0	0.331	0.228	0.287
f_5	0.471	0.500	0.313	0.222	0.376	0.818	0.023	0.287	0.077	0.301	0.598	0.043	0.299	0.114	0.264
f_6	0.484	0	0.203	0.185	0.218	0.729	0	0.172	0.231	0.283	0.582	0	0.186	0.205	0.243
f_7	0.518	0	0.329	0.217	0.266	0.776	0	0.276	0.288	0.335	0.621	0	0.300	0.248	0.292
f_8	0.500	0	0.255	0.190	0.236	0.782	0	0.149	0.288	0.305	0.610	0	0.188	0.229	0.257
f_9	0.463	0.375	0.180	0.195	0.303	0.812	0.034	0.103	0.154	0.276	0.590	0.063	0.131	0.172	0.239
f_{10}	0.476	0	0.280	0.143	0.225	0.929	0	0.161	0.038	0.282	0.629	0	0.204	0.061	0.224
f_{11}	0.445	0.600	0.292	0.056	0.348	0.853	0.034	0.161	0.019	0.267	0.585	0.065	0.207	0.029	0.221
f_{12}	0.438	0.167	0.293	0.071	0.242	0.829	0.011	0.138	0.038	0.254	0.573	0.021	0.188	0.050	0.208
f_{13}	0.445	0.200	0.349	0	0.248	0.859	0.011	0.172	0	0.261	0.586	0.022	0.231	0	0.210
f_{14}	0.483	0.429	0.290	0.161	0.341	0.824	0.068	0.207	0.096	0.299	0.609	0.118	0.242	0.120	0.272
f_{15}	0.465	0.571	0.209	0.230	0.369	0.782	0.045	0.103	0.269	0.300	0.583	0.084	0.138	0.248	0.263
f_{16}	0.460	0	0.563	0.414	0.359	0.953	0	0.103	0.231	0.322	0.621	0	0.175	0.296	0.273
f_{17}	0.428	0	0	0	0.107	1	0	0	0	0.250	0.600	0	0	0	0.150
f_{18}	0.529	0.333	0.282	0.329	0.368	0.759	0.034	0.230	0.462	0.371	0.623	0.062	0.253	0.384	0.331
f_{19}	0.440	0.250	0.247	0.170	0.277	0.629	0.045	0.241	0.173	0.272	0.518	0.077	0.244	0.171	0.253
f_{11}, f_{12}, f_{13}	0.448	0.238	0.288	0.128	0.275	0.729	0.057	0.172	0.115	0.269	0.555	0.092	0.216	0.121	0.246
f_1 to f_{14} (all lexicons)	0.554	0.407	0.318	0.204	0.371	0.753	0.125	0.310	0.212	0.350	0.638	0.191	0.314	0.208	0.338
all features	0.541	0.214	0.308	0.292	0.339	0.782	0.034	0.230	0.404	0.363	0.639	0.059	0.263	0.339	0.325

					total
size of dictionary	**3819**	**2054**	**2595**	**1772**	
no. of taken terms	**1337**	**719**	**908**	**620**	
no. of development samples	**170**	**88**	**87**	**52**	**397**
no. of train samples	**260**	**364**	**315**	**594**	**1533**

In summary, results in Tables 4.3 and 4.5 state that, there is not any single feature set classifier that performs well for all emotions or intensity levels. For a more detailed analysis of feature sets for each level of emotions, the performance of feature sets are sorted and tabulated in Tables 4.6 through 4.10. In these tables feature sets are sorted in ascending order on the basis of micro F-scores achieved by their corresponding classifier.

As stated before and as can be seen from Table 4.6, f_3 is the best feature set with the highest micro F-score for the anger emotion classification. However, it does not rank amongst the top ten well performing feature sets for the rest of the emotions. As another sample consider f_{16} that its classifier achieves the worst performance for anger and joy emotions. The classifier performance dominates other feature sets in classifying fear and sadness emotions intensity levels. Moreover, rankings under fear emotion indicate that majority of feature sets perform almost the same, even though this performance is very different for other emotions. Such inconsistencies in performance of feature sets and their trained classifiers for different emotions proves that feature sets, similar to parameters, should be selected based on emotions. Tables 4.7 through 4.10 give ranked feature sets per emotion on a class based performance respectively for anger, joy, fear, and sadness emotions.

Comparing ranks within tables makes it evident that similar attributes do not perform well for all classes similar to the case of emotions. For instance, classifier of f_3 that ranked first in overall intensity classification for the anger emotion (Table 4.6), ranks first only in predicting samples with the lowest intensity level and its performance places it at the bottom of the list as the last feature set for other levels (Table 4.7).

Table 4.6: Sorted and ranked feature sets according to their performance on development data

Anger		Joy		Fear		Sadness		Micro F-scores			
attr.	**rank**	**attr.**	**rank**	**attr.**	**rank**	**attr.**	**rank**	**anger**	**joy**	**fear**	**sadness**
f_3	1	f_5	1	f_{16}	1	f_{16}	1	0.433	0.307	0.650	0.461
f_1	2	f_4	2	f_7	2	f_{18}	2	0.415	0.290	0.648	0.443
f_2	3	f_7	3	f_4	3	f_{10}	3	0.389	0.272	0.645	0.438
f_4	4	f_8	3	f_5	3	f_4	4	0.376	0.272	0.645	0.436
f_{10}	4	f_{15}	3	f_8	3	f_1	5	0.376	0.272	0.645	0.433
f_9	6	f_{18}	6	f_9	3	f_7	6	0.374	0.266	0.645	0.431
f_{19}	6	f_{19}	6	f_{10}	3	f_2	7	0.374	0.266	0.645	0.428
f_5	8	f_6	8	f_{11}	3	f_5	7	0.366	0.252	0.645	0.428
f_{18}	9	f_2	9	f_{12}	3	f_{17}	7	0.363	0.241	0.645	0.428
f_{14}	10	f_9	9	f_{13}	3	f_{14}	10	0.356	0.241	0.645	0.426
f_6	11	f_{10}	11	f_{15}	3	f_3	11	0.348	0.238	0.645	0.416
f_7	11	f_{14}	12	f_{17}	3	f_{11}	12	0.348	0.231	0.645	0.411
f_8	11	f_1	13	f_{18}	3	f_{13}	13	0.348	0.224	0.645	0.408
f_{15}	14	f_{12}	13	f_{19}	3	f_8	14	0.335	0.224	0.645	0.406
f_{12}	15	f_{11}	15	f_2	15	f_{15}	15	0.320	0.214	0.643	0.403
f_{13}	16	f_{13}	16	f_6	15	f_9	16	0.309	0.210	0.643	0.398
f_{11}	17	f_3	17	f_{14}	15	f_{12}	17	0.307	0.207	0.643	0.393
f_{17}	17	f_{17}	17	f_1	18	f_6	18	0.307	0.207	0.640	0.380
f_{16}	19	f_{16}	19	f_3	18	f_{19}	19	0.222	0.186	0.640	0.355
f_{11}, f_{12}, f_{13}								0.320	0.255	0.640	0.378
f_1 to f_{14} (all lexicons)								0.371	0.324	0.638	0.446
all features								**0.379**	**0.283**	**0.648**	**0.446**

Moreover, this feature set is totally ineffective in detection of samples with the first level of intensity. In fact there are disagreements about the feature set that works better for recognition of intensity level for every emotion. Unlike emotions, choosing a subset of feature sets which work well for each intensity level dose not seem feasible. Hence, results on performance of feature sets and selection of features are based on micro F-score, which takes into account contribution for all classes.

Last three rows of Tables 4.7 through 4.10 show micro F-scores for combination of features. Comparison of scores shows that all feature sets and all lexicons alternatively have the best performance on different levels of emotions. However, combination of feature sets 11 through 13 mostly performs the worse and rarely shows better performance.

Table 4.7: Level-wise sorted and ranked feature sets for anger emotion

Level 0		Level 1		Level 2		Level 3		Micro F-scores			
attr.	**rank**	**attr.**	**rank**	**attr.**	**rank**	**attr.**	**rank**	**level 0**	**level 1**	**level 2**	**level 3**
f_3	1	f_{19}	1	f_{10}	1	f_{18}	1	0.590	0.100	0.425	0.384
f_1	2	f_{15}	2	f_{17}	2	f_8	2	0.588	0.087	0.407	0.289
f_4	3	f_{18}	3	f_5	3	f_{16}	3	0.552	0.079	0.382	0.273
f_{19}	4	f_6	4	f_{15}	4	f_{15}	4	0.546	0.069	0.374	0.272
f_2	5	f_7	5	f_2	5	f_6	5	0.528	0.067	0.366	0.242
f_9	6	f_8	6	f_9	6	f_{13}	6	0.520	0.036	0.348	0.233
f_{14}	7	f_4	7	f_3	7	f_5	7	0.514	0.034	0.347	0.229
f_7	8	f_{16}	8	f_{19}	8	f_2	8	0.513	0.033	0.343	0.228
f_6	9	f_2	9	f_{12}	9	f_4	9	0.502	0.030	0.338	0.219
f_5	10	f_1	10	f_{18}	10	f_9	10	0.494	0	0.332	0.218
f_8	11	f_3	10	f_{11}	11	f_{12}	11	0.477	0	0.331	0.217
f_{10}	12	f_5	10	f_4	12	f_{14}	12	0.465	0	0.324	0.216
f_{18}	13	f_9	10	f_{13}	13	f_7	13	0.447	0	0.315	0.214
f_{11}	14	f_{10}	10	f_1	14	f_1	14	0.405	0	0.302	0.213
f_{12}	15	f_{11}	10	f_6	15	f_{17}	15	0.404	0	0.282	0.188
f_{13}	16	f_{12}	10	f_7	16	f_{19}	16	0.394	0	0.278	0.174
f_{15}	17	f_{13}	10	f_8	16	f_{11}	17	0.389	0	0.278	0.172
f_{17}	18	f_{14}	10	f_{14}	16	f_{10}	18	0.261	0	0.278	0.171
f_{16}	19	f_{17}	10	f_{16}	19	f_3	19	0.173	0	0.239	0.138
f_{11}, f_{12}, f_{13}								0.438	0.058	0.300	0.203
f_1 to f_{14} (all lexicons)								0.528	0.083	0.330	0.238
all features								**0.447**	**0.113**	**0.363**	**0.395**

Table 4.8: Level-wise sorted and ranked feature sets for joy emotion

Level 0		Level 1		Level 2		Level 3		Micro F-scores			
attr.	**rank**	**attr.**	**rank**	**attr.**	**rank**	**attr.**	**rank**	**level 0**	**level 1**	**level 2**	**level 3**
f_5	1	f_{13}	1	f_5	1	f_4	1	0.406	0.200	0.264	0.390
f_4	2	f_{12}	2	f_7	2	f_{18}	2	0.396	0.196	0.226	0.382
f_8	3	f_{11}	3	f_6	3	f_{15}	3	0.389	0.190	0.220	0.376
f_{15}	4	f_8	4	f_1	4	f_{19}	4	0.373	0.164	0.203	0.356
f_7	5	f_{10}	5	f_9	5	f_5	5	0.371	0.156	0.192	0.353
f_6	6	f_1	6	f_{10}	6	f_{16}	6	0.357	0.154	0.185	0.305
f_{18}	6	f_7	7	f_{19}	7	f_2	7	0.357	0.142	0.176	0.296
f_{14}	8	f_5	8	f_8	8	f_{17}	8	0.353	0.136	0.171	0.295
f_{19}	9	f_6	9	f_4	9	f_8	9	0.344	0.125	0.154	0.283
f_2	10	f_{19}	10	f_{18}	10	f_{14}	10	0.333	0.120	0.134	0.265
f_9	10	f_4	11	f_2	11	f_3	11	0.333	0.107	0.119	0.258
f_{12}	12	f_9	12	f_{15}	12	f_7	12	0.332	0.106	0.113	0.252
f_{10}	13	f_2	13	f_3	13	f_9	13	0.322	0.101	0.103	0.229
f_{13}	14	f_{18}	14	f_{11}	14	f_{10}	14	0.321	0.088	0.079	0.214
f_{11}	15	f_{15}	15	f_{12}	15	f_{12}	15	0.317	0.076	0.077	0.190
f_{17}	16	f_{14}	16	f_{14}	16	f_6	16	0.310	0.067	0.063	0.189
f_1	17	f_3	17	f_{16}	17	f_1	17	0.286	0.039	0.040	0.184
f_3	18	f_{16}	18	f_{13}	18	f_{13}	18	0.285	0	0.020	0.171
f_{16}	19	f_{17}	19	f_{17}	19	f_{11}	19	0.113	0	0	0.167
f_{11}, f_{12}, f_{13}								0.400	0.154	0.093	0.266
f_1 to f_{14} (all lexicons)								0.396	0.173	0.354	0.333
all features								**0.365**	**0.109**	**0.179**	**0.387**

Table 4.9: Level-wise sorted and ranked feature sets for fear emotion

Level 0		Level 1		Level 2		Level 3		Micro F-scores			
attr.	**rank**	**attr.**	**rank**	**attr.**	**rank**	**attr.**	**rank**	**level 0**	**level 1**	**level 2**	**level 3**
f_{16}	1	f_{14}	1	f_1	1	f_{16}	1	0.792	0.034	0	0.267
f_7	2	f_{16}	2	f_2	2	f_7	2	0.786	0.031	0	0.080
f_{10}	2	f_1	3	f_3	3	f_{18}	3	0.786	0	0	0.074
f_{15}	2	f_2	3	f_4	4	f_1	4	0.786	0	0	0
f_{18}	2	f_3	3	f_5	5	f_2	4	0.786	0	0	0
f_3	6	f_4	3	f_6	6	f_3	4	0.785	0	0	0
f_1	7	f_5	3	f_7	7	f_4	4	0.784	0	0	0
f_4	7	f_6	3	f_8	8	f_5	4	0.784	0	0	0
f_5	7	f_7	3	f_9	9	f_6	4	0.784	0	0	0
f_8	7	f_8	3	f_{10}	10	f_8	4	0.784	0	0	0
f_9	7	f_9	3	f_{11}	11	f_9	4	0.784	0	0	0
f_{11}	7	f_{10}	3	f_{12}	12	f_{10}	4	0.784	0	0	0
f_{12}	7	f_{11}	3	f_{13}	13	f_{11}	4	0.784	0	0	0
f_{13}	7	f_{12}	3	f_{14}	14	f_{12}	4	0.784	0	0	0
f_{17}	7	f_{13}	3	f_{15}	15	f_{13}	4	0.784	0	0	0
f_{19}	7	f_{15}	3	f_{16}	16	f_{14}	4	0.784	0	0	0
f_{14}	17	f_{17}	3	f_{17}	17	f_{15}	4	0.783	0	0	0
f_2	18	f_{18}	3	f_{18}	18	f_{17}	4	0.782	0	0	0
f_6	18	f_{19}	3	f_{19}	19	f_{19}	4	0.782	0	0	0
f_{11}, f_{12}, f_{13}								0.783	0	0	0
f_1 to f_{14} (all lexicons)								0.791	0.054	0.095	0.069
all features								**0.786**	**0.034**	**0**	**0.077**

Table 4.10: Level-wise sorted and ranked feature sets for sadness emotion

Level 0		Level 1		Level 2		Level 3		Micro F-scores			
attr.	**rank**	**attr.**	**rank**	**attr.**	**rank**	**attr.**	**rank**	**level 0**	**level 1**	**level 2**	**level 3**
f_{10}	1	f_{14}	1	f_4	1	f_{18}	1	0.629	0.118	0.331	0.384
f_2	2	f_{15}	2	f_7	2	f_{16}	2	0.624	0.084	0.300	0.296
f_{18}	3	f_3	3	f_5	3	f_7	3	0.623	0.079	0.299	0.248
f_7	4	f_{19}	4	f_{18}	4	f_{15}	3	0.621	0.077	0.253	0.248
f_{16}	4	f_{11}	5	f_1	5	f_8	5	0.621	0.065	0.248	0.229
f_1	6	f_1	6	f_{19}	6	f_4	6	0.619	0.063	0.244	0.228
f_8	7	f_9	6	f_{14}	7	f_2	7	0.610	0.063	0.242	0.222
f_{14}	8	f_{18}	8	f_{13}	8	f_6	8	0.609	0.062	0.231	0.205
f_{17}	9	f_5	9	f_{11}	9	f_9	9	0.600	0.043	0.207	0.172
f_3	10	f_{13}	10	f_{10}	10	f_{19}	10	0.598	0.022	0.204	0.171
f_5	10	f_{12}	11	f_8	11	f_1	11	0.598	0.021	0.188	0.126
f_9	12	f_2	12	f_{12}	11	f_{14}	12	0.590	0	0.188	0.120
f_4	13	f_4	12	f_2	13	f_5	13	0.587	0	0.187	0.114
f_{13}	14	f_6	12	f_6	14	f_{10}	14	0.586	0	0.186	0.061
f_{11}	15	f_7	12	f_{16}	15	f_3	15	0.585	0	0.175	0.050
f_{15}	16	f_8	12	f_3	16	f_{12}	15	0.583	0	0.162	0.050
f_6	17	f_{10}	12	f_{15}	17	f_{11}	17	0.582	0	0.138	0.029
f_{12}	18	f_{16}	12	f_9	18	f_{13}	18	0.573	0	0.131	0
f_{19}	19	f_{17}	12	f_{17}	19	f_{17}	18	0.518	0	0	0
f_{11}, f_{12}, f_{13}								0.555	0.092	0.216	0.121
f_1 to f_{14} (all lexicons)								0.638	0.191	0.314	0.208
all features								**0.639**	**0.059**	**0.263**	**0.339**

Results discussed provide a general overview for the performance of features generated using the development set and accordingly feature sets with the highest precision, recall, and F-score values were determined based on different emotions and levels of intensity. In order to check the validity of the conclusions, the classifiers developed using train data are tested with the test set, that are not seen before by the classifiers. The expectation is that the performances be similar to those of the development data.

Micro F-scores of classifiers tested using test data are given in Table 4.11. In general, except for joy emotion, F-scores have slightly decreased, which was already expected. Regarding the best feature sets for every emotion, similar to the results for development set, f_3 and f_{16} have the highest performances for anger and fear emotions respectively. However, the best feature set for joy and sadness emotions has changed to f_{18}. By sorting feature sets in ascending order according to the micro

Table 4.11: Micro and macro F-scores of classifiers on test data

	Micro F-score				**Macro F-score**				**Average**
	anger	**joy**	**fear**	**sadness**	**anger**	**joy**	**fear**	**sadness**	**micro F.**
f_1	0.393	0.245	0.636	0.374	0.261	0.224	0.201	0.226	0.412
f_2	0.366	0.249	0.640	0.395	0.302	0.220	0.195	0.249	0.413
f_3	0.400	0.198	0.635	0.392	0.249	0.150	0.201	0.234	0.406
f_4	0.372	0.305	0.642	0.409	0.287	0.280	0.195	0.257	0.432
f_5	0.379	0.304	0.642	0.407	0.296	0.282	0.195	0.259	0.433
f_6	0.358	0.281	0.643	0.405	0.273	0.265	0.200	0.273	0.422
f_7	0.346	0.264	0.643	0.412	0.266	0.244	0.200	0.285	0.416
f_8	0.326	0.271	0.641	0.401	0.254	0.255	0.206	0.274	0.410
f_9	0.342	0.239	0.641	0.402	0.259	0.213	0.195	0.264	0.406
f_{10}	0.311	0.239	0.643	0.388	0.219	0.217	0.203	0.212	0.395
f_{11}	0.291	0.209	0.642	0.374	0.226	0.187	0.195	0.217	0.379
f_{12}	0.298	0.211	0.642	0.385	0.231	0.190	0.195	0.212	0.384
f_{13}	0.298	0.211	0.642	0.385	0.234	0.190	0.195	0.222	0.384
f_{14}	0.366	0.234	0.640	0.413	0.270	0.206	0.206	0.263	0.413
f_{15}	0.320	0.277	0.641	0.391	0.263	0.252	0.202	0.255	0.407
f_{16}	0.256	0.220	0.645	0.412	0.221	0.151	0.254	0.218	0.383
f_{17}	0.240	0.240	0.642	0.408	0.150	0.180	0.195	0.145	0.383
f_{18}	0.368	0.328	0.642	0.439	0.323	0.308	0.219	0.333	0.444
f_{19}	0.391	0.283	0.642	0.371	0.320	0.266	0.195	0.278	0.422
f_{11}, f_{12}, f_{13}	0.340	0.232	0.639	0.376	0.272	0.215	0.199	0.243	0.397
f_1 to f_{14} (all lexicons)	0.375	0.322	0.638	0.405	0.301	0.310	0.256	0.317	0.435
all features	**0.344**	**0.323**	**0.642**	**0.412**	**0.296**	**0.305**	**0.216**	**0.314**	**0.430**

Table 4.12: Sorted and ranked feature sets according to their performance on test data

Anger		**Joy**		**Fear**		**Sadness**		**Micro average**			
attr.	**rank**	**attr.**	**rank**	**attr.**	**rank**	**attr.**	**rank**	**anger**	**joy**	**fear**	**sadness**
f_3	1	f_{18}	1	f_{16}	1	f_{18}	1	0.400	0.328	0.645	0.439
f_1	2	f_4	2	f_6	2	f_{14}	2	0.393	0.305	0.643	0.413
f_{19}	3	$\mathbf{f_5}$	**3**	f_7	2	f_7	3	0.391	0.304	0.643	0.412
f_5	4	f_{19}	4	f_{10}	2	$\mathbf{f_{16}}$	**3**	0.379	0.283	0.643	0.412
f_4	5	f_6	5	f_4	5	f_4	5	0.372	0.281	0.642	0.409
f_{18}	6	f_{15}	6	f_5	5	f_{17}	6	0.368	0.277	0.642	0.408
f_2	7	f_8	7	f_{11}	5	f_5	7	0.366	0.271	0.642	0.407
f_{14}	7	f_7	8	f_{12}	5	f_6	8	0.366	0.264	0.642	0.405
f_6	9	f_2	9	f_{13}	5	f_9	9	0.358	0.249	0.642	0.402
f_7	10	f_1	10	f_{17}	5	f_8	10	0.346	0.245	0.642	0.401
f_9	11	f_{17}	11	f_{18}	5	f_2	11	0.342	0.240	0.642	0.395
f_8	12	f_9	12	f_{19}	5	f_3	12	0.326	0.239	0.642	0.392
f_{15}	13	f_{10}	12	f_8	13	f_{15}	13	0.320	0.239	0.641	0.391
f_{10}	14	f_{14}	14	f_9	13	f_{10}	14	0.311	0.234	0.641	0.388
f_{12}	15	f_{16}	15	f_{15}	13	f_{12}	15	0.298	0.220	0.641	0.385
f_{13}	15	f_{12}	16	f_2	16	f_{13}	15	0.298	0.211	0.640	0.385
f_{11}	17	f_{13}	16	f_{14}	16	f_1	17	0.291	0.211	0.640	0.374
f_{16}	18	f_{11}	18	f_1	18	f_{11}	17	0.256	0.209	0.636	0.374
f_{17}	19	f_3	19	f_3	19	f_{19}	19	0.240	0.198	0.635	0.371
f_{11}, f_{12}, f_{13}								0.340	0.232	0.639	0.376
f_1 to f_{14} (all lexicons)								0.375	0.322	0.638	0.405
all features								**0.344**	**0.323**	**0.642**	**0.412**

F-score of their corresponding classifiers (Table 4.12), changes in ranking of feature sets can be seen more clearly. Considering joy emotion, f_5 that trained the best performing classifier with F-score 0.307 in validation experiments, ranks third in the test phase with F-score 0.304. Indeed, the performance of f_5 has not changed significantly, but micro F-score of f_{18} has increased by 23% from 0.266 in validation to 0.328 in test. Therefore, f_5 is still not the best performing feature set during testing stage. Analogous to joy emotion, for sadness emotion, f_{18} with F-score 0.443 trained the second best classifier in validation and its performance has not decreased dramatically during testing. However, performance of f_{16}, the best feature set in validation has reduced around 11% from 0.461 to 0.412 and resulted in a change of rankings. Even though slight decreases in performance during the test phase can be tolerated and performance improvements as signs of well-trained classifiers are desirable, changes in rankings and top performing feature sets are not preferred. In

fact in real world cases, all features are not evaluated again on the test set and only the best features from validation stage are kept. Dramatic fluctuations in performance of selected features using the validation data set may result in missing the best performance when the model is used. Therefore, it is aimed to select and retrieve the best and the most reliable features using a representative development set.

Last column of Table 4.11 gives average of micro F-scores for every trained classifier using feature sets for all emotions. Comparing these scores with the ones from validation stage (last column of Table 4.3) shows that except for f_{18}, the average performance of other feature sets has decreased. Improvement to result for f_{18} may mean that train data set has similar characteristics to test data set and training has helped the model to learn more about data. Besides, as the variance of averages is low, increase in them indicates better performance. Among individual feature sets, lexicons f_{11}, f_{12}, and f_{13} achieve the lowest average score, while other feature sets obtain almost similar performances. Regarding combination of features, the performance of the combination f_{11} through f_{13} on the test data has not changed in comparison with validation. Unfortunately, this combination achieves the lowest performance. Average performance of the combination of all features and all lexicons has also decreased by 2% using test data. However, the combination of all lexicons is still the best after f_{18}.

In table 4.13 precision, recall, and F-score values of trained classifiers using the test data are provided. For anger emotion, f_{18} is still performs as the best feature set with the highest macro recall and F-score. However, the performance of f_2 surpasses

Table 4.13: Precision, recall, and F-score values for developed models using test data

	Anger emotion														
	precision					**recall**					**F-score**				
intensity levels	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**
f_1	0.517	0	0.258	0.196	0.243	0.630	0	0.284	0.219	0.283	0.568	0	0.271	0.207	0.261
f_2	0.561	0.526	0.279	0.216	**0.396**	0.413	0.068	0.486	0.322	0.322	0.476	0.120	0.354	0.258	0.302
f_3	0.505	0	0.268	0.165	0.235	0.673	0	0.288	0.123	0.271	0.577	0	0.278	0.141	0.249
f_4	0.630	0.143	0.260	0.225	0.314	0.447	0.007	0.444	0.384	0.321	0.523	0.013	0.328	0.284	0.287
f_5	0.607	0.182	0.300	0.217	0.327	0.439	0.014	0.473	0.404	0.332	0.509	0.025	0.367	0.282	0.296
f_6	0.580	0.077	0.280	0.185	0.281	0.458	0.007	0.362	0.390	0.304	0.512	0.012	0.316	0.251	0.273
f_7	0.576	0.077	0.264	0.186	0.276	0.439	0.007	0.333	0.418	0.299	0.498	0.012	0.295	0.257	0.266
f_8	0.570	0	0.228	0.208	0.252	0.378	0	0.337	0.473	0.297	0.455	0	0.272	0.289	0.254
f_9	0.567	0	0.269	0.177	0.253	0.402	0	0.449	0.322	0.293	0.470	0	0.336	0.228	0.259
f_{10}	0.491	0	0.222	0.148	0.215	0.394	0	0.449	0.137	0.245	0.437	0	0.297	0.142	0.219
f_{11}	0.479	0.125	0.253	0.139	0.249	0.301	0.020	0.490	0.205	0.254	0.370	0.035	0.334	0.166	0.226
f_{12}	0.477	0.231	0.264	0.135	0.277	0.310	0.020	0.498	0.212	0.260	0.375	0.037	0.345	0.165	0.231
f_{13}	0.479	0.250	0.261	0.145	0.284	0.299	0.027	0.510	0.219	0.264	0.368	0.049	0.345	0.174	0.234
f_{14}	0.518	0.100	0.268	0.210	0.274	0.505	0.007	0.346	0.322	0.295	0.511	0.013	0.302	0.254	0.270
f_{15}	0.546	0.080	0.237	0.227	0.272	0.357	0.027	0.399	0.370	0.288	0.432	0.040	0.297	0.281	0.263
f_{16}	0.703	0.182	0.281	0.177	0.336	0.168	0.041	0.235	0.795	0.309	0.271	0.066	0.256	0.290	0.221
f_{17}	0	0	0.238	0.248	0.122	0	0	0.860	0.212	0.268	0	0	0.373	0.229	0.150
f_{18}	0.630	0.181	0.264	0.283	0.339	0.396	0.101	0.416	0.473	**0.346**	0.486	0.130	0.323	0.354	**0.323**
f_{19}	0.607	0.193	0.288	0.241	0.332	0.499	0.074	0.337	0.459	0.342	0.548	0.107	0.311	0.316	0.320
f_{11}, f_{12}, f_{13}	0.533	0.088	0.267	0.218	0.277	0.398	0.020	0.395	0.390	0.301	0.456	0.033	0.319	0.279	0.272
f_1 to f_{14} (all lexicons)	0.628	0.182	0.264	0.223	0.324	0.465	0.041	0.391	0.404	0.325	0.534	0.066	0.315	0.287	0.301
all features	0.626	0.120	0.244	0.266	0.314	0.368	0.061	0.399	0.466	0.323	0.463	0.081	0.303	0.338	0.296

					total
no. of test samples					**1002**
no. of train samples	**445**	**322**	**507**	**427**	**1701**

Table 4.13 (cont.): Precision, recall, and F-score values for developed models using test data

	Joy emotion														
	precision					recall					F-score				
intensity levels	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**
f_1	0.200	0.327	0.413	0.317	0.314	0.789	0.108	0.106	0.202	0.301	0.319	0.163	0.168	0.246	0.224
f_2	0.213	0.244	0.278	0.337	0.268	0.758	0.066	0.042	0.417	0.321	0.332	0.104	0.072	0.373	0.220
f_3	0.183	0.273	0.357	0.241	0.263	0.830	0.018	0.028	0.193	0.267	0.300	0.034	0.052	0.214	0.150
f_4	0.248	0.351	0.475	0.327	0.350	0.701	0.060	0.156	0.573	0.373	0.367	0.103	0.234	0.417	0.280
f_5	0.271	0.360	0.398	0.305	0.334	0.701	0.081	0.142	0.560	0.371	0.391	0.132	0.209	0.395	0.282
f_6	0.238	0.258	0.396	0.318	0.303	0.727	0.069	0.175	0.385	0.339	0.358	0.109	0.243	0.349	0.265
f_7	0.236	0.228	0.347	0.299	0.278	0.747	0.063	0.139	0.349	0.324	0.358	0.099	0.198	0.322	0.244
f_8	0.247	0.263	0.398	0.275	0.296	0.747	0.132	0.119	0.307	0.327	0.372	0.176	0.184	0.290	0.255
f_9	0.217	0.214	0.315	0.279	0.256	0.784	0.054	0.108	0.252	0.300	0.340	0.086	0.161	0.265	0.213
f_{10}	0.205	0.370	0.339	0.232	0.287	0.753	0.120	0.103	0.188	0.291	0.323	0.181	0.158	0.208	0.217
f_{11}	0.184	0.343	0.340	0.195	0.265	0.665	0.108	0.050	0.220	0.261	0.288	0.164	0.087	0.207	0.187
f_{12}	0.186	0.321	0.328	0.184	0.255	0.660	0.135	0.056	0.183	0.258	0.291	0.190	0.095	0.184	0.190
f_{13}	0.180	0.336	0.383	0.186	0.271	0.655	0.150	0.050	0.174	0.257	0.283	0.207	0.088	0.180	0.190
f_{14}	0.212	0.264	0.246	0.306	0.257	0.814	0.111	0.042	0.225	0.298	0.337	0.156	0.071	0.259	0.206
f_{15}	0.225	0.414	0.382	0.346	0.342	0.804	0.072	0.094	0.422	0.348	0.352	0.123	0.151	0.380	0.252
f_{16}	0.329	0.444	0.355	0.205	0.333	0.134	0.012	0.031	0.927	0.276	0.190	0.023	0.056	0.336	0.151
f_{17}	0.205	0	0	0.349	0.138	0.881	0	0	0.431	0.328	0.332	0	0	0.386	0.180
f_{18}	0.253	0.350	0.448	0.411	0.366	0.784	0.063	0.203	0.532	0.395	0.383	0.107	0.279	0.464	0.308
f_{19}	0.235	0.410	0.371	0.299	0.329	0.660	0.102	0.119	0.495	0.344	0.346	0.163	0.181	0.373	0.266
f_{11}, f_{12}, f_{13}	0.220	0.287	0.306	0.203	0.254	0.655	0.105	0.094	0.275	0.282	0.329	0.154	0.144	0.234	0.215
f_1 to f_{14} (all lexicons)	0.278	0.400	0.407	0.323	0.352	0.758	0.132	0.194	0.436	0.380	0.407	0.199	0.263	0.371	0.310
all features	0.262	0.338	0.414	0.382	0.349	0.758	0.075	0.200	0.518	0.388	0.389	0.123	0.270	0.440	0.305

					total
no. of test samples					**1105**
no. of train samples	**363**	**346**	**359**	**548**	**1616**

Table 4.13 (cont.): Precision, recall, and F-score values for developed models using test data

	Fear emotion													
	precision					recall					F-score				
intensity levels	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**
f_1	0.643	0	0	0.143	0.196	0.989	0	0	0.014	0.251	0.779	0	0	0.026	0.201
f_2	0.641	0	0	0	0.160	0.997	0	0	0	0.249	0.780	0	0	0	0.195
f_3	0.642	0	0	0.111	0.188	0.987	0	0	0.014	0.250	0.778	0	0	0.025	0.201
f_4	0.642	0	0	0	0.160	1	0	0	0	0.250	0.782	0	0	0	0.195
f_5	0.642	0	0	0	0.160	1	0	0	0	0.250	0.782	0	0	0	0.195
f_6	0.645	0.333	0	0	0.245	1	0.008	0	0	0.252	0.784	0.016	0	0	0.200
f_7	0.644	1	0	0	0.411	1	0.008	0	0	0.252	0.783	0.016	0	0	0.200
f_8	0.643	1	0	0.200	0.461	0.995	0.008	0	0.014	0.254	0.781	0.016	0	0.026	0.206
f_9	0.642	0	0	0	0.160	0.998	0	0	0	0.250	0.781	0	0	0	0.195
f_{10}	0.643	0	0	1	0.411	1	0	0	0.014	0.254	0.782	0	0	0.028	0.203
f_{11}	0.642	0	0	0	0.160	1	0	0	0	0.250	0.782	0	0	0	0.195
f_{12}	0.642	0	0	0	0.160	1	0	0	0	0.250	0.782	0	0	0	0.195
f_{13}	0.642	0	0	0	0.160	1	0	0	0	0.250	0.782	0	0	0	0.195
f_{14}	0.643	0.200	0	0.333	0.294	0.994	0.008	0	0.014	0.254	0.781	0.016	0	0.027	0.206
f_{15}	0.643	0	0	0.250	0.223	0.997	0	0	0.014	0.253	0.782	0	0	0.027	0.202
f_{16}	0.658	0.250	0.500	0.368	0.444	0.986	0.032	0.006	0.099	0.281	0.789	0.057	0.013	0.156	0.254
f_{17}	0.642	0	0	0	0.160	1	0	0	0	0.250	0.782	0	0	0	0.195
f_{18}	0.651	0.200	0.222	0.286	0.340	0.992	0.008	0.013	0.028	0.260	0.786	0.016	0.024	0.051	0.219
f_{19}	0.642	0	0	0	0.160	1	0	0	0	0.250	0.782	0	0	0	0.195
f_{11}, f_{12}, f_{13}	0.643	0.143	0	0	0.197	0.994	0.008	0	0	0.250	0.781	0.015	0	0	0.199
f_1 to f_{14} (all lexicons)	0.664	0.120	0.333	0.333	0.363	0.970	0.024	0.044	0.070	0.277	0.788	0.040	0.078	0.116	0.256
all features	0.648	0.250	0.167	0.400	0.366	0.994	0.008	0.006	0.028	0.259	0.784	0.016	0.012	0.053	0.216

					total
no. of test samples					**986**
no. of train samples	**320**	**249**	**193**	**1490**	**2252**

Table 4.13 (cont.): Precision, recall, and F-score values for developed models using test data

	Sadness emotion														
	precision					recall					F-score				
intensity levels	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**	**0**	**1**	**2**	**3**	**macro avg.**
f_1	0.428	0.160	0.242	0.192	0.255	0.794	0.021	0.086	0.178	0.270	0.556	0.037	0.127	0.185	0.226
f_2	0.433	0	0.351	0.212	0.249	0.791	0	0.180	0.186	0.289	0.560	0	0.238	0.198	0.249
f_3	0.418	0.379	0.292	0.208	0.324	0.842	0.057	0.082	0.116	0.274	0.558	0.099	0.128	0.149	0.234
f_4	0.452	0.500	0.308	0.250	0.377	0.799	0.010	0.255	0.109	0.293	0.577	0.020	0.279	0.151	0.257
f_5	0.455	0.200	0.300	0.271	0.307	0.786	0.005	0.263	0.124	0.295	0.576	0.010	0.280	0.170	0.259
f_6	0.482	0	0.331	0.213	0.257	0.764	0	0.204	0.302	0.318	0.591	0	0.252	0.250	0.273
f_7	0.489	0.333	0.329	0.225	0.344	0.766	0.010	0.216	0.310	0.326	0.597	0.020	0.261	0.261	0.285
f_8	0.475	0.286	0.328	0.222	0.328	0.756	0.010	0.169	0.349	0.321	0.583	0.020	0.223	0.271	0.274
f_9	0.468	0.316	0.327	0.169	0.320	0.791	0.031	0.188	0.178	0.297	0.588	0.057	0.239	0.174	0.264
f_10	0.420	0	0.295	0.160	0.219	0.832	0	0.153	0.062	0.262	0.558	0	0.202	0.089	0.212
f_11	0.403	0.300	0.313	0.182	0.300	0.804	0.016	0.102	0.124	0.261	0.537	0.030	0.154	0.147	0.217
f_12	0.410	0.190	0.338	0.169	0.277	0.847	0.021	0.090	0.085	0.261	0.553	0.037	0.142	0.113	0.212
f_13	0.407	0.167	0.356	0.203	0.283	0.822	0.005	0.122	0.124	0.268	0.545	0.010	0.181	0.154	0.222
f_14	0.454	0.235	0.322	0.254	0.316	0.822	0.021	0.220	0.124	0.296	0.584	0.038	0.261	0.167	0.263
f_15	0.450	0.125	0.272	0.240	0.272	0.776	0.005	0.122	0.310	0.303	0.570	0.010	0.168	0.270	0.255
f_16	0.429	0	0.344	0.268	0.260	0.927	0	0.043	0.171	0.285	0.586	0	0.077	0.209	0.218
f_17	0.408	0	0	0	0.102	1	0	0	0	0.250	0.580	0	0	0	0.145
f_18	0.495	0.364	0.399	0.303	0.390	0.731	0.021	0.310	0.419	0.370	0.590	0.039	0.349	0.352	0.333
f_19	0.450	0.069	0.300	0.256	0.269	0.626	0.010	0.278	0.310	0.306	0.524	0.018	0.289	0.281	0.278
f_{11}, f_{12}, f_{13}	0.425	0.200	0.288	0.208	0.280	0.759	0.021	0.141	0.194	0.279	0.545	0.038	0.189	0.201	0.243
f_1 to f_{14} (all lexicons)	0.510	0.239	0.332	0.246	0.332	0.668	0.057	0.271	0.380	0.344	0.578	0.092	0.298	0.299	0.317
all features	0.493	0.357	0.301	0.300	0.363	0.696	0.026	0.247	0.442	0.353	0.577	0.048	0.272	0.357	0.314
					total										
no. of test samples					**975**										
no. of train samples	**260**	**364**	**315**	**594**	**1533**										

that of f_6 by achieving higher macro precision score in detection of correct labels of samples. Moreover, the feature sets considered do not show any improvement in classification of samples for fear emotion in comparison with validation results and class 0 is yet the best predicted label among all emotions. In addition, macro F-scores are given as well in Table 4.11. Comparison of validation and test results reveals that despite of minor changes in macro scores, best feature sets with the highest average performance have not changed.

4.4 Feature selection

Reported micro F-scores for combinations of feature sets in Table 4.3 suggest that subsets of feature sets may improve classification performance. Basically, there are different methods to select the best combinations. Brute force search is one of the techniques that assesses performance of all possible combinations. However, it is not practical on large collections of features and consequently intelligent and faster approaches are required. Wrapper methods (section 2.15) are the applied methods in this study that start with the single best or combination of all features and in a recursive manner tries to expand or decrease the set of features in the combination until no more improvement in performance is attained. In the next subsections, four already discussed wrapper based techniques are used in an effort to select the best subset of feature sets to improve the classification performance.

Random Forward Selection (RFS), one of the variations of forward feature selection technique, starts with the best performing single feature and continues by appending randomly selected single feature to the previously chosen ones at every iteration. Repetition and extending the feature combination continue as long as the

Table 4.14: Results of RFS method on validation set

	Anger		Joy		Fear		Sadness	
	feature set	**micro F-score**	**feature set**	**micro F-score**	**feature set**	**micro F-score**	**feature set**	**micro F-score**
initial step	f_1	0.415	f_1	0.224	f_1	0.640	f_1	0.433
	f_2	0.389	f_2	0.241	f_2	0.643	f_2	0.428
	f_3	0.433	f_3	0.207	f_3	0.640	f_3	0.416
	f_4	0.376	f_4	0.290	f_4	0.645	f_4	0.436
	f_5	0.366	f_5	0.307	f_5	0.645	f_5	0.428
	f_6	0.348	f_6	0.252	f_6	0.643	f_6	0.380
	f_7	0.348	f_7	0.272	f_7	0.648	f_7	0.431
	f_8	0.348	f_8	0.272	f_8	0.645	f_8	0.406
	f_9	0.374	f_9	0.241	f_9	0.645	f_9	0.398
	f_{10}	0.376	f_{10}	0.238	f_{10}	0.645	f_{10}	0.438
	f_{11}	0.307	f_{11}	0.214	f_{11}	0.645	f_{11}	0.411
	f_{12}	0.320	f_{12}	0.224	f_{12}	0.645	f_{12}	0.393
	f_{13}	0.309	f_{13}	0.210	f_{13}	0.645	f_{13}	0.408
	f_{14}	0.356	f_{14}	0.231	f_{14}	0.643	f_{14}	0.426
	f_{15}	0.335	f_{15}	0.272	f_{15}	0.645	f_{15}	0.403
	f_{16}	0.222	f_{16}	0.186	f_{16}	0.650	f_{16}	0.461
	f_{17}	0.307	f_{17}	0.207	f_{17}	0.645	f_{17}	0.428
	f_{18}	0.363	f_{18}	0.266	f_{18}	0.645	f_{18}	0.443
	f_{19}	0.374	f_{19}	0.266	f_{19}	0.645	f_{19}	0.355
1^{st} iter.	f_3, f_1	0.430	f_5, f_{10}	0.245	f_9, f_{16}	0.648	f_4, f_{16}	0.476
2^{nd} iter.							f_{12}, f_4, f_{16}	0.466

performance of the new combination is better than the previous one. Table 4.14 reports the results of RFS technique on the four emotions. Considering the sadness emotion as an example, f_{16} with the highest micro F-score is selected in the initial step and is passed on to the next iteration.

In the first iteration a randomly selected feature set from the remaining feature sets is selected (i.e. f_4) and is appended to f_{16}. The combination of f_{16} and f_4 is used to train a new classifier and its performance is compared to that of f_{16}. The micro F-score of this new combination (0.476) outperforms that of f_{16}. Thus, the iteration continues and a new random feature set (i.e. f_{12}) from the remaining ones is combined to f_{16} and f_4. The micro F-score of the developed classifier using the combination of f_{16}, f_4, and f_{12} equals 0.466 which is less than the performance of the combination before adding f_{12}. Hence, repetition stops and the best subset of feature sets is determined as the combination of f_{16} and f_4. Using this method, the

generated subset in comparison to the best single feature set and the combination of all features has improved performance by respectively 3% and 6%. The same method is applied to all emotions and f_3, f_5, and f_{16} were selected for anger, joy, and fear emotions respectively.

Forward Selection (FS) works almost similar to RFS method. However, after initial selection of the best feature set, iterations continue by measuring the performance of the combination with each feature set. The best combination is selected and passed on to the next iteration. Table 4.15 shows FS iterations for the sadness emotion. f_{16} is the feature set passed on to the first iteration from the initial step with the highest micro F-score that is equal to 0.460. In the first iteration, the performance of the developed classifiers, that are combination of f_{16} with each of the remaining feature sets, is measured and the best performing pair is selected. (f_{16}, f_4) is the best set that is passed on to the next iteration with micro F-score 0.469. Since in the second iteration there is no combination surpasses the performance of (f_4, f_{16}), repetition terminates. Results of FS technique for all emotions are summarized in Table 4.16.

The FS method in comparison with the RFS method achieved better results for all emotions. However, improvements are not very significant except for the joy emotion. It is worth mentioning that, since the FS method is a greedy algorithm, in every iteration combinations with equal F-scores to that of last selected subset are all considered as possible candidates for the next repetition. However, the smallest subset with similar performance is considered as the best set if no improvement is achieved in the next iterations.

Table 4.15: FS technique iterations for sadness emotion

Initial step		1^{st} iter.		2^{nd} iter.	
feature set	**micro F-scores**	**feature set**	**micro F-scores**	**feature set**	**micro F-scores**
f_1	0.433	f_1, f_{16}	0.418	f_1, f_{16}, f_4	0.456
f_2	0.428	f_2, f_{16}	0.423	f_2, f_{16}, f_4	0.456
f_3	0.416	f_3, f_{16}	0.426	f_3, f_{16}, f_4	0.453
f_4	0.436	f_4, f_{16}	0.469	– – – –	– – –
f_5	0.428	f_5, f_{16}	0.463	f_5, f_{16}, f_4	0.461
f_6	0.380	f_6, f_{16}	0.411	f_6, f_{16}, f_4	0.428
f_7	0.431	f_7, f_{16}	0.413	f_7, f_{16}, f_4	0.433
f_8	0.406	f_8, f_{16}	0.413	f_8, f_{16}, f_4	0.446
f_9	0.398	f_9, f_{16}	0.418	f_9, f_{16}, f_4	0.438
f_{10}	0.438	f_{10}, f_{16}	0.423	f_{10}, f_{16}, f_4	0.458
f_{11}	0.411	f_{11}, f_{16}	0.388	f_{11}, f_{16}, f_4	0.431
f_{12}	0.393	f_{12}, f_{16}	0.390	f_{12}, f_{16}, f_4	0.448
f_{13}	0.408	f_{13}, f_{16}	0.385	f_{13}, f_{16}, f_4	0.428
f_{14}	0.426	f_{14}, f_{16}	0.443	f_{14}, f_{16}, f_4	0.436
f_{15}	0.403	f_{15}, f_{16}	0.403	f_{15}, f_{16}, f_4	0.403
f_{16}	0.461	– – –	– – –	– – – –	– – –
f_{17}	0.428	f_{17}, f_{16}	0.453	f_{17}, f_{16}, f_4	0.453
f_{18}	0.443	f_{18}, f_{16}	0.443	f_{18}, f_{16}, f_4	0.443
f_{19}	0.355	f_{19}, f_{16}	0.358	f_{19}, f_{16}, f_4	0.416

Simplified Forward feature Selection (SFS), similar to the FS method, begins with the single best feature and in every iteration appends the next best single feature from the initial step to the combination set. Table 4.17 gives results for the SFS technique for all emotions. As an example, for sadness emotion the initial step starts with f_{16} i.e. the best trained classifier. In the first iteration f_{18}, the second best feature, is appended to f_{16}. Since performance of the generated set decreases, no more selection is required and the best subset is f_{16}.

Backward Selection (BS) unlike different versions of forward feature selection method, starts with the combination of all features and a feature is randomly

Table 4.16: Feature sets selected by FS technique

	Feature set	**Micro F-score**
Anger	f_3	0.433
Joy	$f_2, f_5, f_8, f_9, f_{11}, f_{12}, f_{19}$	0.379
Fear	$f_4, f_{11}, f_{13}, f_{14}, f_{16}$	0.656
Sadness	f_4, f_{16}	0.469

Table 4.17: Iterations and results for SFS technique

	Anger		Joy		Fear		Sadness	
	feature set	**micro F-score**	**feature set**	**micro F-score**	**feature set**	**micro F-score**	**feature set**	**micro F-score**
initial step	f_1	0.415	f_1	0.224	f_1	0.640	f_1	0.433
	f_2	0.389	f_2	0.241	f_2	0.643	f_2	0.428
	f_3	0.433	f_3	0.207	f_3	0.640	f_3	0.416
	f_4	0.376	f_4	0.290	f_4	0.645	f_4	0.436
	f_5	0.366	f_5	0.307	f_5	0.645	f_5	0.428
	f_6	0.348	f_6	0.252	f_6	0.643	f_6	0.380
	f_7	0.348	f_7	0.272	f_7	0.648	f_7	0.431
	f_8	0.348	f_8	0.272	f_8	0.645	f_8	0.406
	f_9	0.374	f_9	0.241	f_9	0.645	f_9	0.398
	f_{10}	0.376	f_{10}	0.238	f_{10}	0.645	f_{10}	0.438
	f_{11}	0.307	f_{11}	0.214	f_{11}	0.645	f_{11}	0.411
	f_{12}	0.320	f_{12}	0.224	f_{12}	0.645	f_{12}	0.393
	f_{13}	0.309	f_{13}	0.210	f_{13}	0.645	f_{13}	0.408
	f_{14}	0.356	f_{14}	0.231	f_{14}	0.643	f_{14}	0.426
	f_{15}	0.335	f_{15}	0.272	f_{15}	0.645	f_{15}	0.403
	f_{16}	0.222	f_{16}	0.186	f_{16}	0.650	f_{16}	0.461
	f_{17}	0.307	f_{17}	0.207	f_{17}	0.645	f_{17}	0.428
	f_{18}	0.363	f_{18}	0.266	f_{18}	0.645	f_{18}	0.443
	f_{19}	0.374	f_{19}	0.266	f_{19}	0.645	f_{19}	0.355
1^{st} iter.	f_3, f_1	0.430	f_5, f_4	0.307	f_7, f_{16}	0.650	f_{16}, f_{18}	0.443
2^{nd} iter.			f_5, f_4, f_7	0.314	f_7, f_{11}, f_{16}	0.648		
3^{rd} iter.			f_5, f_4, f_7, f_{15}	0.272				

removed at every iteration until no more improvement in performance is achieved or a single feature remains. Table 4.18 summarizes the results of the BS technique for anger, joy, fear, and sadness emotions. All emotions start with classifiers trained using the combination of 19 feature sets. However, for an emotion such as sadness 5 feature sets remain at the end. Indeed, elimination of 14 randomly selected feature sets does not worsen the performance of the developed classifiers. Nevertheless by removing them and hence reducing the feature dimension, training and testing times decrease.

Table 4.18: Feature sets selected by BS method

	Anger	**Joy**	**Fear**	**Sadness**
micro F-score	0.379	0.283	0.648	0.446
selected features	all except f_{11}	all except $f_1, f_3, f_4, f_9, f_{10}, f_{12}, f_{13}$	all except $f_3, f_6, f_{10}, f_{13}, f_{16}, f_{17}$	$f_3, f_5, f_9, f_{15}, f_{18}$

Table 4.19: Summary of feature subsets performance on development data set

	Anger		Joy		Fear		Sadness	
	feature set	**micro F-score**	**feature set**	**micro F-score**	**feature set**	**micro F-score**	**feature set**	**micro F-score**
RFS	f_3	0.433	f_5	0.307	f_{16}	0.650	f_4, f_{16}	0.476
FS	f_3	0.433	$f_2, f_5, f_8, f_9, f_{11}, f_{12}, f_{19}$	0.379	$f_4, f_{11}, f_{13}, f_{14}, f_{16}$	0.656	f_4, f_{16}	0.476
SFS	f_3	0.433	f_4, f_5, f_7	0.314	f_{16}	0.650	f_{16}	0.461
BS	all except f_{11}	0.379	all except $f_1, f_3, f_4, f_9, f_{10}, f_{12}, f_{13}$	0.283	all except $f_3, f_6, f_{10}, f_{13}, f_{16}, f_{17}$	0.648	$f_3, f_5, f_9, f_{15}, f_{18}$	0.446
single best	f_3	0.433	f_5	0.307	f_{16}	0.650	f_{16}	0.461
f_{11}, f_{12}, f_{13}	-	0.320	-	0.255	-	0.640	-	0.378
all lexicons	f_1 to f_{14}	0.371	f_1 to f_{14}	0.324	f_1 to f_{14}	0.638	f_1 to f_{14}	0.446
all features	–	0.379	–	0.283	–	0.648	–	0.446

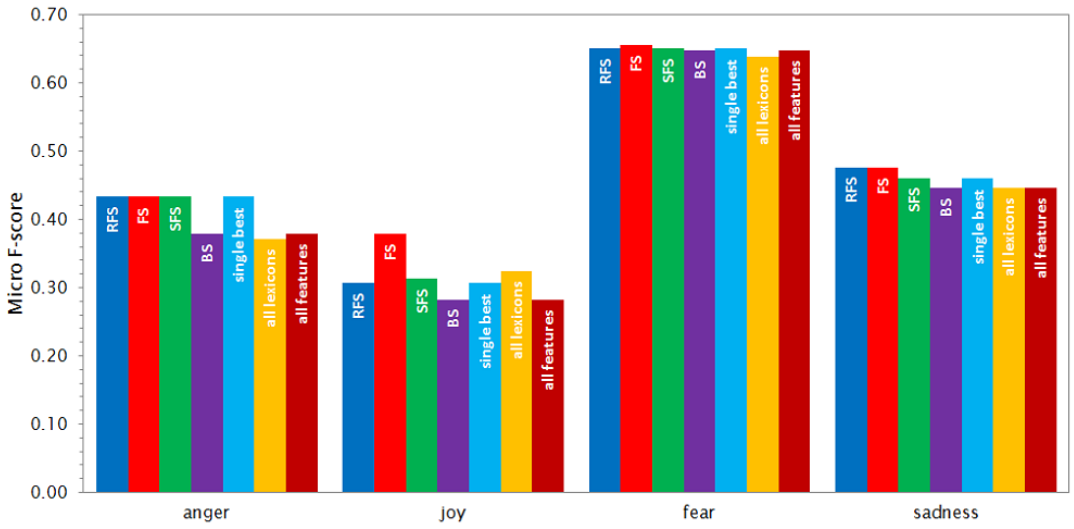


Figure 4.4: Comparison of feature subsets performance on development data set

A summary of the discussed feature selection techniques is given in Table 4.19 and Figure 4.4. For anger emotion, RFS, FS, and SFS approaches give similar result and select f_3 as the best performing subset with micro F-score 0.433. In fact the selected

subset is a single feature set that its combination with any other feature set reduces classification performance. For joy emotion, $(f_2, f_5, f_8, f_9, f_{11}, f_{12}, f_{19})$ is the selected subset by the FS technique that forms the best combination with micro F-score 0.379 which is around 34% more than the micro F-score of the combination of all features. Moreover, by using the BS method since feature elimination is random-based, the removed features did not improve the performance similar to the case of FS and the performance is still far below that of FS. For fear emotion, similar to joy, the FS technique outperforms other methods and selects $(f_4, f_{11}, f_{13}, f_{14}, f_{16})$ as the best feature subset which is working slightly better than the result of the rest of discussed selection methods. Indeed in the case of fear emotion, since none of the combinations of feature sets similar to the single feature sets are successful in classification of samples for classes 1 through 3, they demonstrates a similar performance. (f_4, f_{16}) is the best selected combination by RFS and FS techniques for sadness emotion with micro F-score of 0.476 which is around 6% higher than the performance of the combination of all features.

In general the selected best subset in comparison to the single best feature set, combination of all lexicons, and combination of all features shows improvement in performance for all emotions. In fact, except for anger emotion, where the best subset is a single feature set, selection techniques help by detecting subsets of feature sets that perform better than the combination of all feature sets with a lower feature dimensionality.

By comparing different feature selection methods for four studied emotions, the forward selection and its variations outperform other selection techniques. These

two observations prove that any similarity among the best selected subsets for different emotions cannot be found, and the combination of a large number of features does not always guarantee better predictions.

Feature selection techniques try to choose the best subset of attributes that perform the best classification. Bearing in mind that test labels are not available in practice and decisions are merely made on the basis of train and development data, this study continues to focus on the selected subsets of feature sets from earlier feature selection discussions and checks whether the results can be generalized to the test sets.

Table 4.20 and Figure 4.5 show micro F-scores of classifiers of the selected feature subsets using the test data. The expectation is that the recommended subsets by the selection techniques perform well with the test data as well. During validation all variations of forward feature selection method recommended f_3, individually, as the best subset for anger emotion. Comparing F-score of generated classifier for f_3 on the test data proves that it generally performs better in comparison to classifier from BS technique. Indeed trained classifier using f_3 with F-score 0.401 is more successful on average in classification of samples than the combination of features from BS method with F-score 0.383. Same holds for joy emotion and the best subset from FS method with F-score 0.379 on validation achieves the best F-score on the test data as well. Moreover, the recommended subset by BS method works better on the test data and achieves almost equal performance to that of the FS subset. For fear and sadness emotions, unlike the other two emotions, the recommended subsets do not work well on the test data. The feature subset given by the FS technique for fear emotion is not the best for the test data and the recommended combination by FS or

Table 4.20: Summary of feature subsets performance on test data using different feature selection methods

	Anger		Joy		Fear		Sadness	
	feature set	**micro F-score**	**feature set**	**micro F-score**	**feature set**	**micro F-score**	**feature set**	**micro F-score**
RFS	f_3	0.401	f_5	0.304	f_{16}	0.645	f_4, f_{16}	0.410
FS	f_3	0.401	$f_2, f_5, f_8, f_9, f_{11}, f_{12}, f_{19}$	0.329	$f_4, f_{11}, f_{13}, f_{14}, f_{16}$	0.644	f_4, f_{16}	0.410
SFS	f_3	0.401	f_4, f_5, f_7	0.319	f_{16}	0.645	f_{16}	0.412
BS	all except f_{11}	0.383	all except $f_1, f_3, f_4, f_9, f_{10}, f_{12}, f_{13}$	0.328	all except $f_3, f_6, f_{10}, f_{13}, f_{16}, f_{17}$	0.642	$f_3, f_5, f_9, f_{15}, f_{18}$	0.417
signle best	f_3	0.401	f_{18}	0.328	f_{16}	0.645	f_{18}	0.439
f_{11}, f_{12}, f_{13}	-	0.340	-	0.232	-	0.639	-	0.376
all lexicons	f_1 to f_{14}	0.375	f_1 to f_{14}	0.322	f_1 to f_{14}	0.638	f_1 to f_{14}	0.405
all features	–	0.344	–	0.323	–	0.642	–	0.412

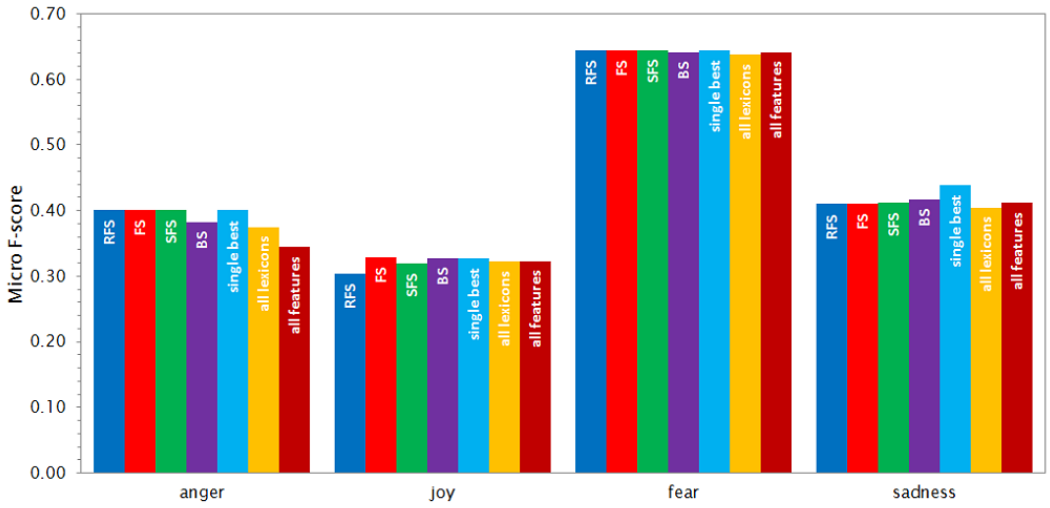


Figure 4.5: Comparison of feature subsets performance on test data

SFS technique ranks first with the highest micro average of F-scores. However, the difference is insignificant. For sadness emotion, the given subset by RFS technique, despite of being the best using validation data with F-score 0.476, shows the lowest

performance using the test set with F-score 0.410 and the subset for BS performs better with F-score equals 0.417.

Single best feature sets for all emotions have equal or better performances in comparison to the recommended subsets by selection techniques. Generally speaking limited differences in performance of any classifier on the test data is expected due to small overfitting by parameters tuning. For joy emotion the difference in performance of the single best and the selected subset is insignificant and can be ignored. For fear and sadness emotions F-scores of the best subsets from validation phase are respectively 1% and 1.7% less than the F-scores of the best subsets in the test phase. Moreover, as mentioned in the study for ordered feature sets (Tables 4.6 and 4.12), changes in rank of single feature sets is unavoidable. Therefore, by ignoring such differences almost same acceptable levels of performances can be obtained with suggested subsets during the validation phase.

4.5 Classifier selection

The discussions on performance of individual feature sets and their combinations has indeed focused on finding out how well single classifiers can perform. An alternative technique to achieve better classification performance is to focus on the outputs of classifiers (predictions) instead of the inputs. That is combining classifiers' predictions with the aim of achieving better results. In Section 2.16 better performance of some classifiers on some subspaces of input domain was addressed as the main intention of applying classifier selection methods. Moreover, majority voting was introduced as the used selection technique in this study. In its simplest form, majority voting which is applied on the predictions of each trained classifier,

gives a positive 1 mark (vote) to the predicted class of a sample and finally votes for each class are summed up over classifiers and the label with majority of the votes determines the predicted class for that sample. Predicted classes in comparison to real labels, define precision, recall, and F-score metrics. In other variations of this technique, positive votes are weighted. In this study both the classifier subset selection method and its variations are investigated to check if better results than subset of feature sets can be attained. It is worth mentioning that since random numbers are used in SVM classifiers, predictions and accordingly precision, recall, and F-score values between trials may change. Therefore, rank of close classifiers can be different for similar selection techniques. In this thesis we apply classifier selection among the 19 base classifiers generated using the 19 feature sets discussed earlier and combine them using majority voting.

Table 4.21 summarizes results for unweighted and weighted combination of all classifiers for different emotions. First four rows show precision, recall, micro, and macro F-scores of combination of classifiers' votes without weighting. Next rows show results of voting with six different weighting schemes. In the second, third, and fourth rows, micro, macro and level-wise F-scores of each trained classifier is used, respectively, as weights. Last three rows use the same weights as rows two through four. However, they are normalized before being applied. As an example, in the normalized macro F-score weighting, macro F-scores for each classifier is divided by the summation of macro F-scores over 19 classifiers. In normalized level-wise weights, the denominator in normalization is the summation of F-scores over classes within each classifier.

Table 4.21: Results for different weighting schemes in voting for classifier selection

		Anger					Joy				
		0	**1**	**2**	**3**	**macro F.**	**0**	**1**	**2**	**3**	**macro F.**
unweighted classifier voting	**precision**	0.632	0	0.315	0.238	0.296	0.216	0	0.500	0.353	0.267
	recall	0.554	0	0.526	0.294	0.343	0.927	0	0.011	0.353	0.323
	F-score	0.590	0	0.394	0.263	0.312	0.351	0	0.022	0.353	0.181
	no. of instances	186	54	97	51		55	95	89	51	
	micro F-score			0.436					0.241		
micro F-score weighted classifier voting	**precision**	0.618	0	0.331	0.254	0.301	0.218	0	0.667	0.365	0.313
	recall	0.591	0	0.515	0.294	0.350	0.927	0	0.022	0.373	0.331
	F-score	0.604	0	0.403	0.273	0.320	0.353	0	0.043	0.369	0.191
	no. of instances	186	54	97	51		55	95	89	51	
	micro F-score			0.451					0.248		
macro F-score weighted classifier voting	**precision**	0.620	0	0.314	0.250	0.296	0.217	0	0.667	0.373	0.314
	recall	0.570	0	0.495	0.314	0.345	0.927	0	0.022	0.373	0.331
	F-score	0.594	0	0.384	0.278	0.314	0.352	0	0.043	0.373	0.192
	no. of instances	186	54	97	51		55	95	89	51	
	micro F-score			0.438					0.248		
level-wise F-score classifier voting	**precision**	0.562	0	0.343	0.368	0.318	0.210	0	0	0.381	0.148
	recall	0.710	0	0.474	0.137	0.330	0.945	0	0	0.314	0.315
	F-score	0.627	0	0.398	0.200	0.306	0.343	0	0	0.344	0.172
	no. of instances	186	54	97	51		55	95	89	51	
	micro F-score			0.477					0.234		
N. micro F-score weighted classifier voting	**precision**	0.618	0	0.331	0.254	0.301	0.218	0	0.667	0.365	0.313
	recall	0.591	0	0.515	0.294	0.350	0.927	0	0.022	0.373	0.331
	F-score	0.604	0	0.403	0.273	0.320	0.353	0	0.043	0.369	0.191
	no. of instances	186	54	97	51		55	95	89	51	
	micro F-score			0.451					0.248		
N. macro F-score weighted classifier voting	**precision**	0.620	0	0.314	0.250	0.296	0.217	0	0.667	0.373	0.314
	recall	0.570	0	0.495	0.314	0.345	0.927	0	0.022	0.373	0.331
	F-score	0.594	0	0.384	0.278	0.314	0.352	0	0.043	0.373	0.192
	no. of instances	186	54	97	51		55	95	89	51	
	micro F-score			0.438					0.248		
N. level-wise F-score classifier voting	**precision**	0.567	0	0.360	0.389	0.329	0.214	0	0	0.383	0.149
	recall	0.704	0	0.515	0.137	0.339	0.945	0	0	0.353	0.325
	F-score	0.628	0	0.424	0.203	0.314	0.349	0	0	0.367	0.179
	no. of instances	186	54	97	51		55	95	89	51	
	micro F-score			0.485					0.241		

Table 4.21 (cont.): Results for different weighting schemes in voting for classifier selection

		Fear					Sadness				
		0	**1**	**2**	**3**	**macro F.**	**0**	**1**	**2**	**3**	**macro F.**
unweighted classifiervoting	**precision**	0.645	0	0	0	0.161	0.441	0	0.333	0.600	0.344
	recall	1	0	0	0	0.250	0.994	0	0.034	0.058	0.272
	F-score	0.784	0	0	0	0.196	0.611	0	0.063	0.105	0.195
	no. of instances	251	57	57	24		170	88	87	52	
	micro F-score			0.645					0.441		
micro F-score weighted classifiervoting	**precision**	0.645	0	0	0	0.161	0.441	0	0.333	0.600	0.344
	recall	1	0	0	0	0.250	0.994	0	0.034	0.058	0.272
	F-score	0.784	0	0	0	0.196	0.611	0	0.063	0.105	0.195
	no. of instances	251	57	57	24		170	88	87	52	
	micro F-score			0.645					0.441		
macro F-score weighted classifiervoting	**precision**	0.645	0	0	0	0.161	0.442	0	0.333	0.667	0.361
	recall	1	0	0	0	0.250	0.994	0	0.034	0.077	0.276
	F-score	0.784	0	0	0	0.196	0.612	0	0.063	0.138	0.203
	no. of instances	251	57	57	24		170	88	87	52	
	micro F-score			0.645					0.443		
level-wise F-score classifier voting	**precision**	0.645	0	0	0	0.161	0.428	0	0	0	0.107
	recall	1	0	0	0	0.250	1	0	0	0	0.250
	F-score	0.784	0	0	0	0.196	0.600	0	0	0	0.150
	no. of instances	251	57	57	24		170	88	87	52	
	micro F-score			0.645					0.428		
N. micro F-score weighted classifier voting	**precision**	0.645	0	0	0	0.161	0.441	0	0.333	0.600	0.344
	recall	1	0	0	0	0.250	0.994	0	0.034	0.058	0.272
	F-score	0.784	0	0	0	0.196	0.611	0	0.063	0.105	0.195
	no. of instances	251	57	57	24		170	88	87	52	
	micro F-score			0.645					0.441		
N. macro F-score weighted classifier voting	**precision**	0.645	0	0	0	0.161	0.442	0	0.333	0.667	0.361
	recall	1	0	0	0	0.250	0.994	0	0.034	0.077	0.276
	F-score	0.784	0	0	0	0.196	0.612	0	0.063	0.138	0.203
	no. of instances	251	57	57	24		170	88	87	52	
	micro F-score			0.645					0.443		
N. level-wise F-score classifier voting	**precision**	0.645	0	0	0	0.161	0.428	0	0	0	0.107
	recall	1	0	0	0	0.250	1	0	0	0	0.250
	F-score	0.784	0	0	0	0.196	0.600	0	0	0	0.150
	no. of instances	251	57	57	24		170	88	87	52	
	micro F-score			0.645					0.428		

Comparing different weighting schemes reveals that normalized level-wise weighting of votes works best for anger emotion and improves classification micro F-score to 0.485 (12% more in comparison to the best subset from feature selection). For joy emotion, micro and macro F-scores regardless of being normalized or not, have similar performances as the best weighting strategies. However, micro F-score of combination of classifiers equals 0.248, that is far below (around 47%) the best subset from feature selection method with micro F-score 0.379. Regarding fear emotion, different weighting strategies are almost alike and micro F-scores are equal to 0.645. This is slightly less than the performance of the best feature selection subset. Normalized and un-normalized macro F-scores are the best weighting schemes for sadness emotion. However, with micro F-score 0.443, it does not perform better than the subset generated by the RFS technique for feature selection.

In general, macro F-score in its normalized or un-normalized form, seems to be the best weighting strategy for joy, fear, and sadness emotions. Here this study continues with the un-normalized form for these emotions. For anger emotion since the combination of classifiers perform better with the normalized level-wise weighting, it is used in continue.

Similar to feature selection techniques, subsets of classifiers may outperform individual classifiers or combination of all classifiers. Using the best weighting scheme for each emotion, the same selection approaches to the used ones for feature selection are applied on classifiers to test whether any subset of classifiers exists that performs better than the combination of all classifiers.

Table 4.22: Results for RFS technique for classifier subset selection

	Anger		Joy		Fear		Sadness	
	classifier set	**micro F-score**	**classifier set**	**micro F-score**	**classifier set**	**micro F-score**	**classifier set**	**micro F-score**
initial step	c_1	0.420	c_1	0.224	c_1	0.640	c_1	0.433
	c_2	0.389	c_2	0.241	c_2	0.643	c_2	0.428
	c_3	0.436	c_3	0.207	c_3	0.640	c_3	0.416
	c_4	0.376	c_4	0.290	c_4	0.645	c_4	0.436
	c_5	0.369	c_5	0.307	c_5	0.645	c_5	0.428
	c_6	0.348	c_6	0.252	c_6	0.643	c_6	0.380
	c_7	0.348	c_7	0.272	c_7	0.648	c_7	0.431
	c_8	0.348	c_8	0.272	c_8	0.645	c_8	0.406
	c_9	0.374	c_9	0.241	c_9	0.645	c_9	0.398
	c_{10}	0.379	c_{10}	0.238	c_{10}	0.645	c_{10}	0.438
	c_{11}	0.314	c_{11}	0.214	c_{11}	0.645	c_{11}	0.411
	c_{12}	0.325	c_{12}	0.224	c_{12}	0.645	c_{12}	0.393
	c_{13}	0.317	c_{13}	0.210	c_{13}	0.645	c_{13}	0.408
	c_{14}	0.358	c_{14}	0.231	c_{14}	0.643	c_{14}	0.426
	c_{15}	0.335	c_{15}	0.272	c_{15}	0.645	c_{15}	0.403
	c_{16}	0.222	c_{16}	0.186	c_{16}	0.650	c_{16}	0.461
	c_{17}	0.307	c_{17}	0.207	c_{17}	0.645	c_{17}	0.428
	c_{18}	0.363	c_{18}	0.266	c_{18}	0.645	c_{18}	0.443
	c_{19}	0.374	c_{19}	0.266	c_{19}	0.645	c_{19}	0.355
1^{st} iter.	c_3, c_{17}	0.461	c_5, c_3	0.307	c_{16}, c_{18}	0.650	c_{16}, c_5	0.461
2^{nd} iter.	c_3, c_{17}, c_{14}	0.428	c_5, c_3, c_{16}	0.307	c_{16}, c_{18}, c_9	0.648	c_{16}, c_5, c_{10}	0.456
3^{rd} iter.			c_5, c_3, c_{16}, c_{10}	0.286				

Table 4.22 gives result of the RFS method using classifiers for four emotions. Note that classifiers have the same index as the feature sets they are trained with. For anger emotion, combination of (c_3, c_{17}) achieves the highest micro F-score which is 5% less than the combination of all classifiers and equals 0.461. Classifier 5 outperforms for joy emotion with a micro F-score of 0.307 which is better than the combination of all classifiers. Finally, c_{16}, individually, achieves the highest micro F-score for both fear and sadness emotions with performance better than the combination of all classifiers. Analysis of the FS technique, shows that combination of c_1 and c_3 for anger emotion is the best subset (Table 4.23). However, for other emotions, there may

Table 4.23: Subset of classifiers using FS technique

	Classifier sets	**Micro F-score**
anger	c_1, c_3	0.482
joy	c_5	0.307
fear	c_{16}	0.650
sadness	$c_2, c_{10}, c_{16}, c_{19}$	0.471

be other scenarios. Assume that a combination of classifiers from a previous iteration has dominant votes in the current iteration since the weights and consequently votes are all in favor of the old set. Thus, new combination will have similar predictions and equal scores. For example, if c_x's votes dominate in combination with other classifiers, combinations will achieve micro F-scores equal to that of c_x.

This condition happens for joy, fear, and sadness emotions which use macro F-scores as a weight. For sadness emotion combination of c_{16} (the best single classifier) with the rest of the classifiers results in a micro F-score of 0.460. Continuing the selection with a randomly chosen pair (c_{16} and c_{19}), all triple combinations achieve a micro F-score of 0.460 in the third repetition. Selecting another combination randomly as the best set (c_{10}, c_{16}, c_{19}), and combining all remained classifiers improves micro F-score to 0.471. In the end, a random subset of four classifiers such as (c_2, c_{10}, c_{16}, c_{19}) is considered as the best subset, since there is no more improvement if iterations continue (Table 4.23).

Results of using SFS method for classifier selection is given in Table 4.24. The results suggest that c_3 for anger emotion works as the best subset and achieves a micro F-score of 0.441. For joy emotion the combination (c_4, c_5, c_7) achieves a higher score than any other combination and c_{16}, individually, improves micro F-score for fear emotion insignificantly. For sadness emotion similar result to that of RFS approach is attained. Recall that inconsistency in the recommended combinations by FS and SFS techniques is due to the effect of randomness in SVM classifiers that results in slight variations in performance of classifiers.

Table 4.24: Simplified forward selection(SFS) iterations for classifier selection

	Anger		Joy		Fear		Sadness	
	classifier set	**micro F-score**	**classifier set**	**micro F-score**	**classifier set**	**micro F-score**	**classifier set**	**micro F-score**
initial step	c_1	0.418	c_1	0.224	c_1	0.640	c_1	0.433
	c_2	0.389	c_2	0.241	c_2	0.643	c_2	0.428
	c_3	0.441	c_3	0.207	c_3	0.640	c_3	0.416
	c_4	0.376	c_4	0.290	c_4	0.645	c_4	0.436
	c_5	0.371	c_5	0.307	c_5	0.645	c_5	0.428
	c_6	0.348	c_6	0.252	c_6	0.643	c_6	0.380
	c_7	0.348	c_7	0.272	c_7	0.648	c_7	0.431
	c_8	0.348	c_8	0.272	c_8	0.645	c_8	0.406
	c_9	0.374	c_9	0.241	c_9	0.645	c_9	0.398
	c_{10}	0.381	c_{10}	0.238	c_{10}	0.645	c_{10}	0.438
	c_{11}	0.312	c_{11}	0.214	c_{11}	0.645	c_{11}	0.411
	c_{12}	0.325	c_{12}	0.224	c_{12}	0.645	c_{12}	0.393
	c_{13}	0.314	c_{13}	0.210	c_{13}	0.645	c_{13}	0.408
	c_{14}	0.358	c_{14}	0.231	c_{14}	0.643	c_{14}	0.426
	c_{15}	0.335	c_{15}	0.272	c_{15}	0.645	c_{15}	0.403
	c_{16}	0.222	c_{16}	0.186	c_{16}	0.650	c_{16}	0.461
	c_{17}	0.307	c_{17}	0.207	c_{17}	0.645	c_{17}	0.428
	c_{18}	0.363	c_{18}	0.266	c_{18}	0.645	c_{18}	0.443
	c_{19}	0.374	c_{19}	0.266	c_{19}	0.645	c_{19}	0.355
1^{st} iter.	c_1, c_3	0.438	c_4, c_5	0.307	c_7, c_{16}	0.650	c_{16}, c_{18}	0.443
2^{nd} iter.			c_4, c_5, c_7	0.310	c_4, c_7, c_{16}	0.648		
3^{rd} iter.			c_4, c_5, c_7, c_8	0.293				

Table 4.25 presents results of using BS method for different emotions. Comparing micro F-scores of classifiers combination shows that except for anger emotion, performance for others emotions decreases or remains unchanged. Moreover, for fear emotion c_2 individually has similar performance to combination of all classifiers.

Table 4.26 provides a comparison of results using feature and classifier selection combinations. Results reveal that except for anger emotion, combinations of features achieve better performances, and results from feature selection are generally more

Table 4.25: Subset of classifiers selected using BS technique

	Feature set	**Micro F-score**
anger	all classifiers except c_7	0.487
joy	all classifiers except c_7, c_{13}, c_{14}	0.262
fear	c_2	0.645
sadness	all classifier	0.443

Table 4.26: Summary of feature and classifier selection methods on development data

		Anger	Joy	Fear	Sadness
Feature selection	**micro F-score**	0.433	0.379	0.656	0.476
	selection technique	RFS / FS / SFS	FS	FS	RFS
	feature set	f_3	$f_2, f_5, f_8, f_9, f_{11}, f_{12}, f_{19}$	$f_4, f_{11}, f_{13}, f_{14}$	f_4, f_{16}
Classifier selection	**micro F-score**	0.487	0.310	0.650	0.471
	selection technique	BS	SFS	SFS / FS / RFS	FS
	feature set	all except c_7	c_4, c_5, c_7	c_{16}	$c_2, c_{10}, c_{16}, c_{19}$
All classifiers	**micro F-score**	0.485	0.248	0.645	0.443
	weighting method	N. level-wise	macro F-score	macro F-score	macro F-score

satisfactory than classifiers combination. However, for anger emotion, subset of classifiers selected using the BS technique increases micro F-score obtained using the best subset of feature sets by 10%.

Generally, it maybe concluded that, subsets of classifiers work better than combination of all classifiers. Among different selection techniques used for classifier selection BS, SFS, and FS methods respectively, give the best subsets for anger, joy, and sadness emotions and both FS and SFS methods are equal regarding fear emotion where c_{16}, individually, is selected as the best subset. In order to test whether results obtained using validation data sets can be generalized, similar experiments are conducted using the test data. The results are provided in Table 4.27. For anger emotion, standard level-wise weighting scheme is applied with different selection techniques. Based on the validation results, BS method is expected to have the best performance. Results prove that, as expected, BS technique achieves the highest micro F-score on the test data that is equal to 0.437. For the remaining emotions, validation outcomes recommended macro F-scores as the best weighting strategy. Therefore, macro F-score weighting along with four different selection techniques are experimented on the joy emotion data set and the

combination obtained using SFS, (c_4, c_5, c_7), outperforms with F-score of 0.309. Regarding fear emotion, evaluation is indeed a comparison between recommended subset by various forward selection techniques with the BS method's subset that are c_{16} and c_2, respectively. Results show that c_{16} performs better on the test data in comparison to c_2. Thus, the subset recommended by validation experiment is valid. For sadness emotion, unlike other emotions, combination of (c_2, c_{10}, c_{16}, c_{19}) that had performed well in the validation experiment, does not achieve the best result on the test data and combination of classifiers obtained using BS method achieves higher micro F-score.

Table 4.28 and Figure 4.6 give a summary of feature and classifier selection obtained using the test data. The subset of classifier for anger emotion improves performance in comparison to the feature selection from 0.401 to 0.437 as it was expected according to the validation results. For joy emotion, the performance of the selected subset of classifiers from validation and test phases is almost similar, and classifier selection does not improve performance. Thus, subset of feature sets with around 5% higher micro F-score is preferred. For fear emotion, performances are not much different for feature and classifier subsets and no improvement is achieved. For sadness emotion in contrast to other emotions, although the selected subset of classifiers does not result in higher micro F-scores than the subset of feature sets

Table 4.27: Micro F-score of best subset of classifiers using test data

	Anger	**Joy**	**Fear**	**Sadness**
RFS	0.247	0.304	0.645	0.407
FS	0.404	0.304	0.645	0.403
SFS	0.403	0.309	0.645	0.412
BS	0.437	0.267	0.642	0.423

Table 4.28: Summary of feature and classifier selection methods on test data

		Anger	Joy	Fear	Sadness
Feature selection	**micro F-score**	0.401	0.329	0.645	0.417
	selection technique	FS / RFS / SFS	FS	RFS / SFS	BS
	feature set	f_3	$f_2, f_5, f_8, f_9, f_{11}, f_{12}, f_{19}$	f_{16}	$f_3, f_5, f_9, f_{15}, f_{18}$
Classifier selection	**micro F-score**	0.437	0.309	0.645	0.423
	selection technique	BS	SFS	FS / SFS / RFS	BS
	feature set	all exceptc_7	c_4, c_5, c_7	c_{16}	$c_2, c_{10}, c_{16}, c_{19}$
All classifiers	**micro F-score**	0.437	0.267	0.642	0.424
	weighting meth.	N. level-wise	macro F-score	macro F-score	macro F-score

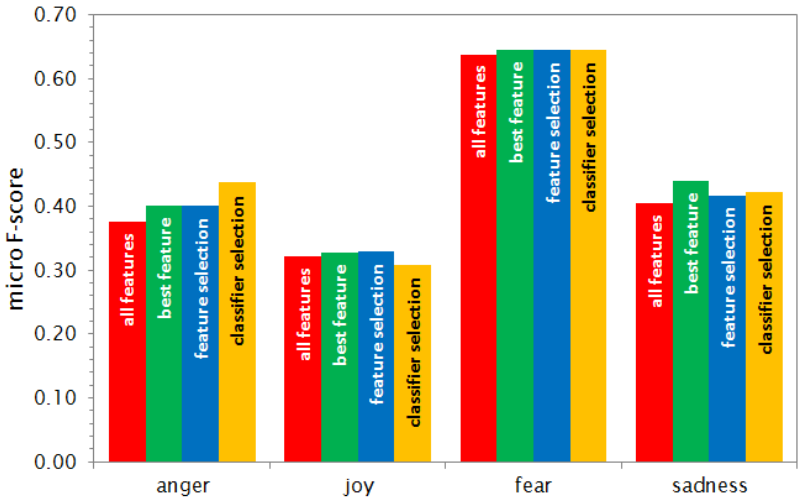


Figure 4.6: Comparison of feature and classifier selection methods on test data

during validation phase, combination of classifiers by BS method improves F-score about 1% from 0.417 to 0.423 during testing which is a minor improvement.

Chapter 5

SUMMARY AND CONCLUSION

In this thesis, a brief discussion on review of psychological studies in field of human emotions and remarking importance of emotions in people interactions was given. Fundamental concepts such as emotions and sentiments were explained extensively and SemEval competition as an effort to explore the nature of meanings and replication of human cognitive processing was introduced with main focus being on textual resources. Among textual resources, Twitter has received extensive attention in recent years due to its characteristics such as briefness of tweets.

Analyzing tweets or generally text data requires techniques of conversion to represent terms and sentences by scores or vector of scores. For this purpose in Chapter 2, different techniques, including word2vec as a measure of terms similarity, tf-idf scoring as a measure of terms importance in a document or corpus, and affect lexicons as a measure of terms relevancy were explained. In the same chapter, linear SVM classifier for model development using extracted features, refinement of sets of features and classifiers, and metrics to measure the performance of classifiers were reviewed. Chapter 3 gave a comprehensive discussion to the data sets used, and all used feature sets such as affect lexicons, tf-idf scoring, word2vec models, query terms, self-dictionary, and symbols. Finally, in Chapter 4 a detailed study on results of the developed classifiers and models for feature and classifier selection was discussed.

For feature extraction and model development, total of 19 feature sets including 14 lexicons from 4 lexicon sets were considered. It was observed that effective use of lexicon scores is only possible by considering tweet lengths. To find the optimal tweet length, performance of models on different tweet lengths was compared and the length with the highest micro F-score was selected. Results revealed that there exists a different optimal length for different emotions. Similar procedure was applied to determine the size of a self-dictionary, as an additional feature source for each emotion. Among individual lexicons the best performing one was "Affirmative Context and Negated Context" lexicon (f_5) from NRC set which achieved the best averaged micro F-score over emotions. Remaining lexicons had similar performances except for Warriner et al. lexicon set that had the lowest average micro F-scores. Regarding performance of lexicons for each emotion word-level NRC hashtag sentiment (f_3), NRC hashtag sentiment v1.0 (f_4), NRC emotion lexicon v1.0 (f_6), and Bing Liu Opinion lexicon (f_{14}) were the most effective ones for anger, joy, fear, and sadness emotions respectively. Among the single feature sets considered, tf-idf scoring showed the best average performance. However, when the performance achieved by combination of all features was compared to the performance achieved by the combination of all lexicons, the latter one performed better.

In order to investigate the effect of feature selection for classification of tweets, we tried wrapper based feature selection techniques. In total four selection strategies including Simplified Forward Selection (SFS), Forward Selection (FS), Randomized Forward Selection (RFS), and Backward Selection (BS) were explained.

Combination of features recommended by these techniques showed improvement in micro F-scores for all emotions. However, the best feature subsets turned out to be different for each emotion. Therefore, it can be concluded that a set of fixed feature subset does not perform well for all emotions and similar to other parameters, specialization in the set of attributes is needed. Due to the possible variations in performance of classifiers on subspaces of input domain and for further improvement in results, wrapper based classifier selection methods were also investigated. Majority voting was applied to combine scores of classifiers in an ensemble. However, to increase accuracy, a weighted voting technique was applied, by considering micro, macro, and level-wise F-scores as weights. Results revealed that subsets of classifiers unlike subsets of feature sets only show better performance for anger and sadness emotions. Moreover, best subset of classifiers differ among emotions.

In conclusion, although selected feature sets were not successful in predicting samples from all levels of emotions, performance of feature sets in terms of average performance was acceptable. One drawback of this study may be the fact that it was modeled as a classical classification problem while a model based on regression may improve the results as showed in the SemEval workshop. Another point may be using of one versus one classification method instead of the one versus rest used in this thesis. These efforts may improve the classification performance when the micro F-score is very low or even zero for some levels of emotions for some emotions.

This study, by considering tf-idf scoring as a source of feature, emphasized the importance of terms appearance frequency and tokenization, as they can directly

affect scoring by lexicons and consequently performance of classifiers. Since tweets are combination of formal and informal ways of writing, development of more intelligent and accurate tokenization systems along with stemming and lemmatization methods can be a topic of future studies. Feature and classifier selection in this area was a novel attempt to reach better combinations. However, as a research area for future studies, more intelligent subset selection algorithms such as neural networks or genetic algorithms may be considered.

REFERENCES

[1] Picard, R. W. (2010). Affective computing. https://affect.media.mit.edu/pdfs/95.picard.pdf.

[2] Morgun, I. (2015). Types of machine learning algorithms.

[3] Garcia-Garcia, J. M., Penichet, V. M., & Lozano, M. D. (2017). Emotion detection: a technology review. *Proceedings of the XVIII International Conference on Human Computer Interaction*, p. 8.

[4] Gosai, D. D., Gohil, H. J., & Jayswal, H. S. (2018). A review on a emotion detection and recognization from text using natural language processing. *International Journal of Applied Engineering Research*, *13*(9), pp. 6745–6750.

[5] Badugu, S., & Suhasini, M. (2017). Emotion detection on twitter data using knowledge base approach. *International Journal of Computer Applications*, *162*(10), pp. 975–978.

[6] De Choudhury, M., Counts, S., & Gamon, M. (2012). Not all moods are created equal! exploring human emotional states in social media. *Sixth international AAAI conference on weblogs and social media*.

[7] Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical*

Informatics Association, *18*(5), pp. 544–551.

[8] Duppada, V., & Hiray, S. (2017). Seernet at emoint-2017: Tweet emotion intensity estimator. *arXiv preprint arXiv:1708.06185*.

[9] Mohammad, S. M., & Bravo-Marquez, F. (2017). Wassa-2017 shared task on emotion intensity. *arXiv preprint arXiv:1708.03700*.

[10] Köper, M., Kim, E., & Klinger, R. (2017). Ims at emoint-2017: emotion intensity prediction with affective norms, automatically extended resources and deep learning. *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 50–57.

[11] Goel, P., Kulshreshtha, D., Jain, P., & Shukla, K. K. (2017). Prayas at emoint 2017: an ensemble of deep neural architectures for emotion intensity prediction in tweets. *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 58–65.

[12] SAS. Machine learning, what it is and why it matters. https://www.sas.com/en_us/insights/analytics/machine-learning.html. Last checked: 04.02.2019.

[13] Shalev-Shwartz, S., & Shai, B.-D. (2014). Understanding machine learning: From theory to algorithms. http://www.cs.huji.ac.il/ shais/UnderstandingMachineLearning.

[14] Semeval-2017 task 4, sentiment analysis in twitter. http://alt.qcri.org/semeval2017/task4/. Last checked: 04.02.2019.

[15] Holzman, L. E., & Pottenger, W. M. (2003). Classification of emotions in internet chat: An application of machine learning using speech phonemes. *Retrieved November*, *27*(2011), p. 50.

[16] Alm, C. O., Roth, D., & Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pp. 579–586.

[17] Brooks, M., Kuksenok, K., Torkildson, M. K., Perry, D., Robinson, J. J., Scott, T. J., Anicello, O., Zukowski, A., Harris, P., & Aragon, C. R. (2013). Statistical affect detection in collaborative chat. *Proceedings of the 2013 conference on Computer supported cooperative work*, pp. 317–328.

[18] Mohammad, S. M. (2012). Emotional tweets. *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pp. 246–255.

[19] Mohammad, S. M. (2015). Sentiment analysis: Detecting valence, emotions, and other affectual states from text. *Emotion measurement*, pp. 201–237.

[20] Pearl, L., & Steyvers, M. (2010). Identifying emotions, intentions, and attitudes in text using a game with a purpose. *Proceedings of the naacl hlt 2010 workshop on computational approaches to analysis and generation of emotion in text*, pp. 71–79.

[21] Wang, Z., et al. (2014). Segment-based fine-grained emotion detection for chinese text. *Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp. 52–60.

[22] Sun, Y., Quan, C., Kang, X., Zhang, Z., & Ren, F. (2015). Customer emotion detection by emotion expression analysis on adverbs. *Information Technology and Management*, *16*(4), pp. 303–311.

[23] Shivhare, S. N., & Khethawat, S. (2012). Emotion detection from text. *arXiv preprint arXiv:1205.4944*.

[24] Shaheen, S., El-Hajj, W., Hajj, H., & Elbassuoni, S. (2014). Emotion recognition from text based on automatically generated rules. *2014 IEEE International Conference on Data Mining Workshop*, pp. 383–392.

[25] Tilakraj, M. M., Shetty, D. D., Nagarathna, M., Shruthi, K., & Narayan, S. Emotion finder: Detecting emotions from text, tweets and audio.

[26] Agrawal, A., & An, A. (2012). Unsupervised emotion detection from text using semantic and syntactic relations. *Proceedings of the The 2012 IEEE/WIC/ACM*

International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01, pp. 346–353.

[27] Cabanac, M. (2002). What is emotion? *Behavioural processes*, *60*, pp. 69–83. doi:10.1016/S0376-6357(02)00078-5.

[28] Kleinginna, P. R., & Kleinginna, A. M. (1981). A categorized list of emotion definitions, with suggestions for a consensual definition. *Motivation and Emotion*, *5*, pp. 345–379. ISSN 1573-6644. doi:10.1007/BF00992553. https://doi.org/10.1007/BF00992553.

[29] Kołakowska, A., Landowska, A., Szwoch, M., Szwoch, W., & Wróbel, M. (2014). Emotion recognition and its applications. *Advances in Intelligent Systems and Computing*, *300*, pp. 51–62.

[30] Ekman, P., Levenson, R., & Friesen, W. (1983). Autonomic nervous system activity distinguishes among emotions. *American Association for the Advancement of Science*, *221*(4616), pp. 1208–1210. ISSN 0036-8075. doi:10.1126/science.6612338. http://science.sciencemag.org/content/221/4616/1208.full.pdf.

[31] Plutchik, R. (1984). Emotions: A general psychoevolutionary theory. *Approaches to emotion*, pp. 197–219.

[32] Bann, E. Y. (2012). Discovering basic emotion sets via semantic clustering on a twitter corpus. *arXiv preprint arXiv:1212.6527*.

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

[34] Frijda, N. H. (1988). The laws of emotion. *American psychologist*, *43*(5), p. 349.

[35] Mejova, Y. (2009). Sentiment analysis: An overview. *University of Iowa, Computer Science Department*.

[36] Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), pp. 1–135.

[37] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86.

[38] Liu, B., Hu, M., & Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the web. *Proceedings of the 14th international conference on World Wide Web*, pp. 342–351.

[39] Melville, P., Gryc, W., & Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1275–1284.

[40] Wiebe, J., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning subjective language. *Computational linguistics*, *30*(3), pp. 277–308.

[41] Liu, B. (2006). Web data mining: exploring hyperlinks, contents, and usage data. *Springer Science & Business Media*.

[42] Gernsbacher, M. A., Goldsmith, H. H., & Robertson, R. R. (1992). Do readers mentally represent characters' emotional states? *Cognition & Emotion*, *6*(2), pp. 89–111.

[43] Gernsbacher, M. A., & Robertson, R. R. (1992). Knowledge activation versus sentence mapping when representing fictional characters' emotional states. *Language and Cognitive Processes*, *7*(3-4), pp. 353–371.

[44] Gygax, P., Garnham, A., & Oakhill, J. (2004). Inferring characters' emotional states: Can readers infer specific emotions? *Language and Cognitive Processes - LANG COGNITIVE PROCESS*, *19*. doi:10.1080/01690960444000016.

[45] Merriam-Webster.com. lexicon. https://www.merriam-webster.com.

[46] Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). The measurement of meaning. *University of Illinois press*, (47).

[47] General inquirer (GI) lexicon. http://www.wjh.harvard.edu/ inquirer/. Accessed: 2019-02-28.

[48] Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

[49] Bradley, M. M., & Lang, P. J. (1999). Affective norms for english words (ANEW): Instruction manual and affective ratings. *Tech. rep.*, Citeseer.

[50] Nielsen, F. Å. (2011). AFINN. http://www2.imm.dtu.dk/pubdb/p.php?6010.

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

[52] Liu, B. Opinion mining, sentiment analysis, and opinion spam detection. https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html. Last checked: 16.03.2019.

[53] Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the 10th ACM SIGKDD international conference on*

Knowledge discovery and data mining, pp. 168–177.

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

[55] Miller, G. (1998). Wordnet: An electronic lexical database. *MIT press*.

[56] Princeton University (2010). About WordNet. https://wordnet.princeton.edu/.

[57] Esuli, A., & Sebastiani, F. (2006). SentiWordNet: A publicly available lexical resource for opinion mining. *LREC*, *6*, pp. 417–422.

[58] Strapparava, C., Valitutti, A., et al. (2004). WordNet affect: an affective extension of WordNet. *LREC*, *4*(1083-1086), p. 40.

[59] Mohammad, S. M., & Yang, T. W. (2011). Tracking sentiment in mail: How genders differ on emotional axes. *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*, pp. 70–79.

[60] Mohammad, S. M., & Turney, P. D. (2010). Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pp. 26–34.

[61] Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, *29*(3), pp. 436–465.

[62] Warriner, A. B., Kuperman, V., & Brysbaert, M. (2013). Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, *45*(4), pp. 1191–1207.

[63] Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, *50*, pp. 723–762.

[64] Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the European chapter of the association for computational linguistics*, pp. 174–181.

[65] Turney, P. D., & Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, *21*(4), pp. 315–346.

[66] Mohammad, S., Dunne, C., & Dorr, B. (2009). Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pp. 599–608.

[67] Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint*

arXiv:1308.6242.

[68] Hentschel, M., & Alonso, O. (2014). Follow the money: A study of cashtags on twitter. *First Monday*, *19*(8).

[69] Jahanbakhsh, K., & Moon, Y. (2014). The predictive power of social media: On the predictability of us presidential elections using twitter. *arXiv preprint arXiv:1407.0622*.

[70] Shi, L., Agarwal, N., Agrawal, A., Garg, R., & Spoelstra, J. (2012). Predicting US primary elections with twitter. http://snap. stanford. edu/social2012/papers/shi.pdf.

[71] Wright, D. K., & Hinson, M. D. (2013). An updated examination of social and emerging media use in public relations practice: A longitudinal analysis between 2006 and 2014. *Public relations journal*, *7*(3), pp. 1–39.

[72] McCorkindale, T., DiStaso, M. W., & Carroll, C. (2013). The power of social media and its influence on corporate reputation. *The handbook of communication and corporate reputation*, pp. 497–512.

[73] Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Data preprocessing for supervised learning. *International Journal of Computer Science*, *1*, pp. 111–117.

[74] What does tf-idf mean? http://www.tfidf.com/. Last checked: 28.02.2019.

[75] Mogotsi, I. (2010). Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval.

[76] Kauchak, D. (2009). Tf-idf. Lecture Notes, http://www.stanford.edu/class/cs276/handouts/lecture6-tfidf.ppt.

[77] Kenter, T. (2014). Word2Vec. Lecture Notes, IR Reading Group.

[78] Rong, X. (2014). Word2Vec parameter learning explained. *arXiv:1411.2738*. Provided by the SAO/NASA Astrophysics Data System, https://ui.adsabs.harvard.edu/#abs/2014arXiv1411.2738R.

[79] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pp. 3111–3119. http://dl.acm.org/citation.cfm?id=2999792.2999959.

[80] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[81] Mohammad, S. (2012). Portable features for classifying emotional text. *Proceedings of the 2012 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pp. 587–591.

[82] Roggeman, K. (2017). *Emotion Detection On Twitter, A Corpus Analysis On The Reputation Of Donald Trump During The American Elections*. Master's thesis. https://lib.ugent.be/fulltxt/RUG01/002/348/977/RUG01-002348977_2017_0001_AC.pdf.

[83] Adreevskaia, A., & Bergler, S. (2006). Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from WordNet glosses. *11th conference of the European chapter of the Association for Computational Linguistics*.

[84] Esuli, A., & Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 617–624.

[85] Esuli, A., & Sebastiani, F. (2007). Pageranking wordnet synsets: An application to opinion mining. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 424–431.

[86] Kamps, J., Marx, M., Mokken, R. J., De Rijke, M., et al. (2004). Using WordNet to measure semantic orientations of adjectives. *LREC*, *4*, pp. 1115–1118.

[87] Van Asch, V. (september 2013). Macro- and micro-averaged evaluation measures. Basic draft.

[88] Trevor, H., Robert, T., & JH, F. (2009). The elements of statistical learning: data mining, inference, and prediction.

[89] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, *3*, pp. 1157–1182. ISSN 1532-4435. http://dl.acm.org/citation.cfm?id=944919.944968.

[90] Li, J., Cheng, K., Wang, S., Morstatter, F., P. Trevino, R., Tang, J., & Liu, H. (2016). Feature selection: A data perspective. *ACM Computing Surveys*, *50*. doi:10.1145/3136625.

[91] Hamel, L. (2015). An introduction to artificial intelligence with AI game development. *University of Rhode Island*. Lecture Notes.

[92] Ver Hoeve, N., Martek, C., & Gardner, B. (2010). Classifier selection. *Rochester Institute of Technology*.

[93] Sharkey, A. J. (2012). Combining artificial neural nets: ensemble and modular multi-net systems. *Springer Science & Business Media*.

[94] Sharkey, A. J. (1999). Linear and order statistics combiners for pattern classification. *Combining artificial neural nets*, pp. 127–161.

[95] Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10), pp. 993–1001.

[96] Giacinto, G., & Roli, F. (2000). Dynamic classifier selection. *International Workshop on Multiple Classifier Systems*, pp. 177–189.

[97] Byeon, B., & Rasheed, K. (2010). Selection of classifier and feature selection method for microarray data. *2010 Ninth International Conference on Machine Learning and Applications*, pp. 534–539.

[98] Fletcher, T. (2008). Support Vector Machines explained. Www.cs.ucl.ac.uk/staff/T.Fletcher/.

[99] Boswell, D. (2002). Introduction to support vector machines. *Departement of Computer Science and Engineering University of California San Diego*.

[100] Kecman, V. (2005). Support vector machines – an introduction. *Support Vector Machines: Theory and Applications*, *177*, pp. 605–605.

[101] R. Berwick, V. I. (2011). An idiot's guide to support vector machines (SVMs). Lecture notes.

[102] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), pp. 273–297.

[103] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org, http://tensorflow.org/.

[104] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, pp. 2825–2830.

[105] Python. https://www.python.org. Last checked: 16.03.2019.

[106] Zhu, X., Kiritchenko, S., & Mohammad, S. (2014). NRC-Canada-2014: Recent improvements in the sentiment analysis of tweets. *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 443–447.

[107] Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010*

Workshop on New Challenges for NLP Frameworks, pp. 45–50. http://is.muni.cz/publication/884893/en.

[108] Pandas, python data analysis library. https://pandas.pydata.org. Last checked: 16.03.2019.

[109] Numpy. http://www.numpy.org. Last checked: 16.03.2019.