NANYANG TECHNOLOGICAL UNIVERSITY

Modelling Public Sentiment in Twitter

By

Prerna Chikersal

School of Computer Engineering

2015

NANYANG TECHNOLOGICAL UNIVERSITY

SCE14-0075
Modelling Public Sentiment in Twitter

Submitted to the School of Computer Engineering,
in partial fulfilment of the requirements of the degree of
Bachelor of Engineering (B.Eng.) in Computer Science
at Nanyang Technological University, Singapore

By
Prerna Chikersal

School of Computer Engineering
2015

In loving memory of my grandmother, Late Ms. Indu Chikersal.

I can't thank her enough for the sacrifices she made for my upbringing and education.

Acknowledgements

I would like to thank Prof. Erik Cambria as I have benefited tremendously from my interactions with him and am grateful for his invaluable advice and guidance. I am deeply inspired by his passion and enthusiasm in pursuing research, and appreciate the many times he has encouraged me. I would also like to thank Prof. Chng Eng Siong for giving me the opportunity to work on this project and for supervising my previous URECA project, which prepared me for this project and others I took up in between. I am also grateful to Soujanya Poria for his useful insights and the fruitful collaboration and discussions we had throughout the project.

I believe my Final Year Project is the culmination of my undergraduate education in computer science. Yet, considering that I will be pursuing graduate studies, it only marks the beginning of my academic career. These past 4 years in Singapore have been the most defining years of my life so far, and several people have contributed to it both personally and academically. Working on projects and hanging out with my friends and colleagues, Nishtha, Rinsha, Budhaditya and others has not only enriched my education, but has also helped me evolve as a human being. Moreover, I cannot thank my academic mentor Prof. Anwitaman Datta enough, for his patience and kindness, and timely advice and encouragement, which helped me thrive in every aspect of life and also played a major role in my decision of pursuing research and graduate studies.

Lastly but most importantly, I am grateful to my grandparents, Chaman Lal and Late Indu Chikersal for the many pains they took for my upbringing and education. I don't know where I would've been had they not been an integral part of my life. Also, I cannot deny that my father, Ajay Chikersal's amusing attempts to teach me science and maths, and how to construct circuits and program lego robots at a very early age, not only sparked my interest in Computer Science, but also helped me acquire fundamental critical thinking skills. Through the toughest times in my life I was lucky to be supported and loved by my godmother, Rachna Kumar, from whom I learnt that family isn't always about blood, rather it's about who holds your hand when you're in need. Together, these four people have been the most loving anchors in my life, and whatever I've achieved till now and will achieve in future will be largely because of them.

Finally, I wish to acknowledge the funding for this project from Nanyang Technological University under the Undergraduate Research Experience on CAmpus (URECA) programme.

Abstract

People often use social media as an outlet for their emotions and opinions. Analysing social media text to extract sentiment can help reveal the thoughts and opinions people have about the world they live in. This thesis contributes to the field of Sentiment Analysis, which aims to understand how people convey sentiment in order to ultimately deduce their emotions and opinions. While several sentiment classification methods have been devised, the increasing magnitude and complexity of social data calls for scrutiny and advancement of these methods. The scope of this project is to improve traditional supervised learning methods for Twitter polarity detection by using rule-based classifiers, linguistic patterns, and common-sense knowledge based information.

This thesis begins by introducing some terminologies and challenges pertaining to sentiment analysis, followed by sub-tasks or goals of sentiment analysis and a survey of commonly used approaches. In the first phase of the project, we propose a sentiment analysis system that combines a rule-based classifier with supervised learning to classify tweets into positive, negative and neutral using the training set provided by [1] and test sets provided by [2]. We find that the average positive and negative f-measure improves by 0.5 units when we add a rule-based classification layer to the supervised learning classifier. This demonstrates that combining high-confidence linguistic rules with supervised learning can improve classification. In the second phase of this project, we extend our work further by proposing a sentiment analysis system that leverages on complex linguistic rules and common-sense based sentic computing resources to enhance supervised learning, and classify tweets into positive and negative. We train our classifier on the training set provided by [3] and test it on positive and negative tweets from the test sets provided by [2] and [4]. We find that our system achieves an average positive and negative f-measure that is 4.47 units and 3.32 units more than the standard n-grams model for the two datasets respectively.

Supervised learning classifiers often misclassify tweets containing conjunctions like "but" and conditionals like “if”, due to their special linguistic characteristics. These classifiers also assign a decision score very close to the decision boundary for a large number tweets, which suggests that they are simply unsure instead of being completely wrong about these tweets. The second system proposed in this thesis attempts to enhance supervised classification by countering these two challenges. An online real-time system (http://www.twitter.gelbukh.com/) is also implemented to demonstrate the results obtained, however it is still primitive and a work-in-progress.

Key words: Sentiment Analysis, Opinion Mining, Sentic Computing, Social Media

The work presented in this thesis will be published as [5] and [6], that is:

- Chikersal, P., Poria, S., Cambria, E.: *SeNTU: Sentiment analysis of tweets by combining a rule-based classifier with supervised learning.* In: Proceedings of the International Workshop on Semantic Evaluation, SemEval 2015.
- Chikersal, P., Poria, S., Cambria, E., Gelbukh, A., Siong, C. E. *Modelling Public Sentiment in Twitter: Using Linguistic Patterns to Enhance Supervised Learning.* In: Proceedings of the 16th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2015, Part II, LNCS 9042, pp. 49–65.

Contents

List of Figures

List of Tables

1 Introduction

"Fundamentals, fundamentals, fundamentals. You've got to get the fundamentals down because otherwise the fancy stuff isn't going to work."

– Randy Pausch, *The Last Lecture*

1.1 Terminology

Sentiment analysis is a field that infers and analyses people's opinions, sentiments, assessments, and emotions towards other people, products, issues, etc. It comprises of several slightly different sub-tasks such as polarity detection, subjectivity detection, opinion mining, and emotion analysis. In this thesis, we will be focussing on polarity detection, that is predicting if a tweet conveys positive, negative or neutral sentiment, or just positive and negative sentiment. Sentiment analysis can be said to be a sub-field of *affective computing* [8] that is a field which focuses of building computers that can detect, express, and "have" emotions. *Sentic Computing* [9] is a new field at the intersection of affective computing and common-sense computing that exploits both common-sense knowledge and emotional knowledge for sentiment analysis.

In this thesis, we will leverage on sentiment lexicons (containing words that express emotions), linguistic patterns (rules based on how parts-of-speech like conditionals and conjunctions are used in language) and common-sense knowledge base(s) to enhance sentiment analysis techniques based on supervised learning.

1.2 Importance of Sentiment Analysis

Our opinions and the opinions of others play a very important role in our decision-making process and therefore influence behaviour. In recent times, the use of social networking sites has become increasingly ubiquitous. This has led an abundance of unstructured yet insightful “social" big data. Considering this, it has become even more essential to devise techniques for the distillation of opinions and trends from unstructured data like microblogging text. This is because, being able to analyse this data and extract opinions about a plethora of topics, can help us make informed choices and predictions regarding those topics. This is why, sentiment analysis of tweets is gaining importance across a number of domains; some examples of which are given below.

- Sentiment analysis for **products and services** has been welcomed by the industry and the consumers. Knowing the percentage of positive and negative reviews, can help consumers make good choices while purchasing products and services. It can also alert companies selling these product and services, when the public opinion towards their goods becomes too negative. Detecting mood states [10] such as anger can also help a company understand how people are responding to their products or services. Using this information, the company can make better decisions regarding their brand development. Sentiment analysis has also been used to predict **success of movies** [11] and tv-shows, and for **crowd validation** of healthcare patient opinions [12].
- In recent times, sentiment analysis has been used to make predictions regarding **politics and polls**, such as predicting public sentiment towards a particular politician [13], predicting election results and how events and news affect public opinion during elections [14, 15].
- Some research has also been done to understand how socio-economic factors affect public mood and discover if sentiment analysis can help us predict **socio-economic** events. For example, in [16], Bollen et al. are able to find some correlation between Twitter sentiment and socio-economic phenomenons, and in [17] they experiment with making stock market predictions.

- Sentiment analysis of discourses has also helped in online **dispute detection**[18], and analysing tweets from different users has helped gather **mental health insights** [19] and **detect depression** [20]. It has also helped us explore and corroborate psychological hypothesis such as the correlation between weather and mood [21].
- Interestingly, in June 2014 there have been reports of the secret service seeking to buy a special software that can detect sarcasm on Twitter. While sarcasm detection is still a very challenging problem, this incident and works such as [22] show that sentiment analysis of microblogging and other text is valuable even for **intelligence and surveillance**.
- Sentiment analysis is also used for analysing and extracting affective information from **multiple modalities** [23] such as video, audio and text.

These are just a few examples taken from recent and not-so-recent literature. Ultimately, the possible applications of sentiment analysis are limitless.

1.3 Thesis organisation

Chapter 2 provides a general background of sentiment analysis. We begin by describing the various sub-tasks of sentiment analysis such as polarity detection, subjectivity detection, topic-based sentiment analysis, aspect-based sentiment analysis, deciphering perspectives, and document-level sentiment analysis. Then, we give a brief overview of some commonly used approaches for sentiment analysis, namely methods based on supervised learning, unsupervised learning and linguistic rules, bag-of-concepts model and sentic computing. Finally, we review literature that focuses of sentiment analysis of microblogging text.

In **Chapter 3**, we describe a Twitter sentiment analysis system developed by combining a rule-based classifier with supervised learning. We submitted our results for the message-level sub-task in SemEval 2015 Task 10, and achieved a F[1]-score of 57.06%.

[1]We average the positive and negative F-measures to get the F-score, which is the evaluation metric for this task.

The rule-based classifier is based on rules that are dependent on the occurrences of emoticons and opinion words in tweets. Whereas, the Support Vector Machine (SVM) is trained on semantic, dependency, and sentiment lexicon based features. The tweets are classified as *positive*, *negative* or *unknown* by the rule-based classifier, and as *positive*, *negative* or *neutral* by the SVM. The hold-out cross-validation method is used to test the system on the test set provided by [2]. On this data, we obtain an average positive and negative F-score of 66.2 units using supervised learning alone, and 66.7 units using the combined supervised learning and rule-based classifier. Hence, these results show that rules can help refine the SVM's predictions.

Chapter 4 and 5 describe a Twitter sentiment analysis system that classifies a tweet as positive or negative based on its overall tweet-level polarity. Supervised learning classifiers often misclassify tweets containing conjunctions like "but" and conditionals like "if", due to their special linguistic characteristics. These classifiers also assign a decision score very close to the decision boundary for a large number tweets, which suggests that they are simply unsure instead of being completely wrong about these tweets. To counter these two challenges, we proposes a system that enhances supervised learning for polarity classification by leveraging on linguistic rules and sentic computing resources. The system is trained on $\approx$ 1.6 million positive and negative tweets provided by [3] and evaluated on positive and negative tweets from the test sets provided by [2] and [4]. We find that our system achieves an average positive and negative f-measure that is 4.47 units and 3.32 units more than the standard n-grams model for the two datasets respectively.

Chapted 6 concludes the thesis by clearly elucidating the inferences derived from our experiments, and summarising the principal contributions of this work.

Chapted 7 describes the real-time Twitter sentiment analysis system currently being developed, and gives a glimpse into the future work for the project.

2 Background

"The most beautiful experience we can have is the mysterious – the fundamental emotion which stands at the cradle of true art and true science."

– Albert Einstein, *The World As I See It*

2.1 Sub-tasks

Based on real-world problems, researchers have been working on various sub-tasks of sentiment analysis. This section gives a brief overview of some of the most common sub-tasks of sentiment analysis, as described in greater detail in [24] and [25].

2.1.1 Polarity Detection and Degrees of Polarity

A text can either be objective or subjective. Subjective text is also known as opinionated text that is text that conveys some opinion. Polarity detection is the task of classifying a given opinionated text as one of two opposing sentiment polarities. At times, it also involves detecting where the text lies between these polarities, which could for example mean detecting whether the text is neutral (neither positive not negative, or equidistant from both poles), detecting the degree of positivity or negativity such as "very positive", "slightly positive", "positive score = 4 on 5", etc. Degree of polarity is a particularly useful metric for sentiment analysis of conditional text [26].

Sentiment polarity classification usually involves labelling a text as positive or negative (binary classification), or positive, negative, or neutral (multi-class classification). Sometimes, instead of training a classifier to discriminate between positive and negative polarities, it may be trained to discriminate between other labels such as "like" vs "dislike", "likely to win" vs "unlikely to win" [27], and possessing a certain orientation (E.g. political [28]). Sentiment expressed in a text can be explicit (E.g. This semester, my grades were excellent!) or implicit (E.g. I got all straight As this semester.). In order to detect implicit sentiments in text, the machine needs to be able to understand factual or common-sense based information, and deduce the sentiment it implies. Hence, implicit sentiment in text further adds to the complexity to the sentiment analysis problem.

Tasks involving prediction of degrees of polarity such as predicting the star rating of a movie review, can be modelled as either a multi-class classification problem (e.g. 5 classes for assigning movie reviews scores of 1,2,3,4, and 5) or a regression problem (e.g. regressor outputs float values between 0 and 5 as scores for movie reviews).

In some papers, the label "neutral" is used as a label for the objective class. However, neutrality and objectivity are two different concepts. An objective text is one that conveys no opinions or expresses a complete lack of opinion; whereas, a neutral text is usually opinionated but lies between the positive and negative classes, thereby expressing a neutral opinion. Thus, due to the unequivocal position of neutral text, carrying out neutral vs polar classification considerably increases the difficulty of the polarity detection task.

2.1.2 Subjectivity Detection

The task of polarity detection often assumes the input text to be opinionated or subjective. Hence, there is a need to perform subjectivity detection before polarity detection in a typical sentiment analysis pipeline. Studies such as [29] have shown that the task of subjectivity detection is often more difficult than the task of polarity detection.

There also exists the task of opinion strength detection that is different from the task of detecting the degrees of polarity. Note that, a neutral text is assumed to have 0 degrees of polarity, however it need not have 0 opinion strength. For example, you can have a very strong opinion that a certain student is neither good nor bad, rather just average.

2.1.3 Topic-based and Aspect-based Sentiment Analysis

In certain situations, it is necessary to detect the sentiment of a text towards a particular topic. Lets consider the problem of detecting the public's overall sentiment towards *IPhone 6* and *Galaxy S5*, given access to all relevant public tweets. We can have three types of tweets – (i) tweets about *IPhone 6* and not *Galaxy S5*, (ii) tweets about *Galaxy S5* and not *IPhone 6*, (iii) tweets about both *IPhone 6* and *Galaxy S5* (such as comparisons and analogies). For types (i) and (ii), we can simply add a topic-based filter preceding the sentiment analysis pipeline tweets, and process and analyse both types of tweets separately. However, to derive any inferences from type (iii) tweets, we will need to build a sentiment analysis system that can detect the sentiment of a tweet towards different topics. Therefore, in cases such as this, it is important identify topics in a text and separate the opinions associated with each topic.

A text can also discuss different aspects of an entity, such as the *battery life* and *display* of a certain smartphone. In cases such as this, it is important to identify the aspects of entities and separate the opinions associated with each aspect of an entity.

Thus, both topic-based and aspect-based sentiment analysis basically involve two steps – (1) identifying the *opinion target(s)* in a text, and (2) determining the sentiment of the text towards that target. Methods may be supervised [30], rule-based [31], or other.

2.1.4 Document and sentence level Sentiment Classification

A "document" is a collection of sentences or texts (e.g. tweets). Often, there is a need to determine the overall sentiment of a document, instead of just the sentiment of the individual sentences it contains. Lets take the following as an example in the context of Twitter – we are given a collection of tweets by "n" different authors pertaining to a certain topic, and we need to determine the overall sentiment of each author towards that topic. How can we do that? We can make use of a number of statistical techniques to do this. One such example is the bayes rule method recently proposed by Volkova et al. [28] for classifying political orientation (analogous to polarity classification) of Twitter users. Moreover, studies such as [32] devise techniques to aggregate sentence-level polarities in order to calculate document-level polarities.

2.1.5 Other Sub-tasks

Another sub-task of sentiment analysis is detecting emotions such as happiness, sadness, fear, anger, disgust and surprise. Cambria et al. [33] have proposed an “hourglass” model for computational modelling of human emotions. Subsequently, the effectiveness of this model for sentiment analysis has been evaluated in studies such as [34] and [35].

Multimodal sentiment analysis [36, 37] which involves using information from multiple modalities such as audio, video and text, is also an interesting topic of research.

Finally, note that the list of sentiment analysis sub-tasks described in this section is not exhaustive. Several sentiment analysis frameworks have been delineated in literature. One such example is the Concept-Level Sentiment Analysis model proposed by Cambria et al. [7] and shown in figure 2.1.

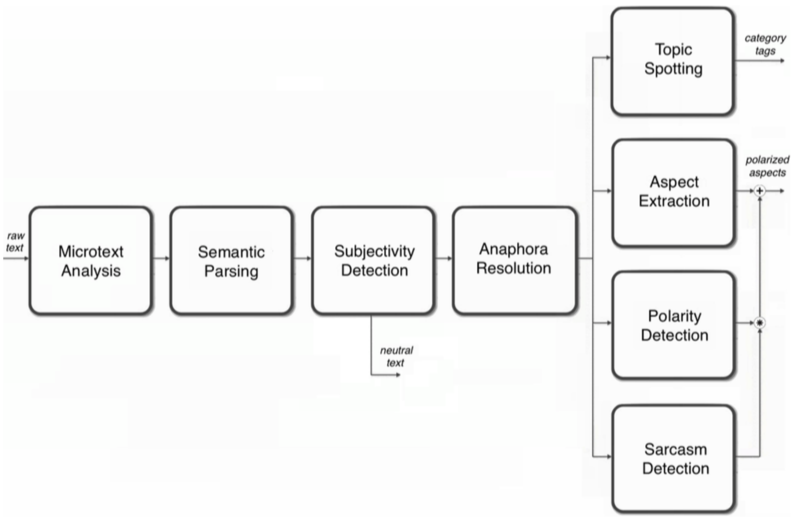

Figure 2.1 – The CLSA model proposed by Cambria et al. [7]

2.2 Approaches

This section briefly describes some of the commonly used methods for sentiment analysis.

2.2.1 Supervised Learning For Sentiment Analysis

Supervised learning methods involve converting a textual sample into a feature vector, and passing the feature vectors of different texts as input to a machine learning (ML) algorithm. Each sample must be labeled as the category to which it belongs. The ML algorithm uses the feature vectors and these labels to approximate a classification function $f(x)$, such that $y = f(x)$ that is given a feature vector x, function f outputs a label y. The key to supervised learning is feature engineering. We must extract pragmatic features from text for good performance.

In their respective books and surveys, Pang and Lee [24] Liu and Zhang [38, 25] describe and assess some of these features. For one of the datasets used by us, features in [39] give close to state-of-the-art performance. Below, we briefly explain what some of the commonly used features mean:

- **N-grams:** These features are word n-grams and their frequency counts. Character n-grams can also be used, however it is unclear if they add value, since they are usually unable to represent semantics of the text. The TF-IDF weighting scheme is often applied to the frequency counts. Word n-grams have proved to be very effective for sentiment analysis.

- **Parts-of-speech tags:** Parts-of-speech such as adjectives are often important indicators of opinions. Moreover, adverbs such as “not" often negate sentiment expressed by the subsequent words, while adverbs such as “exceedingly", and “immensely" greatly intensify sentiment and adverbs such as “barely" and “slightly" greatly weaken sentiment. These adverbs that negate/minise, intensify, weaken, affirm or cast doubt on sentiment, are called adverbs of degree. In [40], Benamara et al. show that using adverbs and adjectives gives better accuracy than adjectives alone.

- **Negation:** Negations have been found to be very crucial, since they invert the sentiment polarity of the sentence. However, negation is still tricky, because some occurrences of negation such as “not only"... “but also" do not invert polarity.

Moreover, [41] highlights some challenges with regard to negation such as negation within words (e.g. harm*less*), differences in polarity strength between words like “bad" and “not good", and the use of figurative language such as irony. Moreover, the use of sarcasm also causes polarity inversion. While there have been attempts such as [42] to detect sarcasm in text, the accuracy still remains very low. This is not surprising considering that even human coders do not perform well in this task. In [42], *smiley*, *frown* and *ToUser* (whether the tweet was a reply to another tweet) were found to be the most discriminating features for sarcasm detection.

- **Sentiment/opinion words and phrases:** Sentiment/opinion words are words that are commonly used to express positive or negative sentiments. For example, “good", “splendid", etc express positive sentiment, while “horrible", “bad", etc express negative sentiment. Features encoding occurrence and count of these opinion words have been found to be very useful for sentiment analysis. Some researchers have found opinion words present at the end of a sentence to have more bearing on the sentiment expressed by the sentence, than the opinion words present in between. Hence, sometimes even positions of opinion words are encoded in the feature vector. Sentiment lexicons are often used to find these words and phrases.

- **Dependency Features:** Apart from negation, other dependency features such as subject-noun relations and adverb modifiers are also used as features for sentiment analysis. The effect of modifiers such as adverbs of degree has already been mentioned above.

Considering the informal nature of microblogging text, the following features are also commonly used for social media text:

- **Emoticons:** The occurrence or count of positive or negative emoticons has been found to be a useful feature for sentiment analysis of social media text. Studies such as [39] have also found the position of emoticons in a sentence to be important. It has been hypothesised that emoticons present at the end of a sentence have more bearing on the sentiment expressed by the sentence, than the emoticons present in between.

- **Hashtags:** Occurrence and count of hashtags are often used as features for sentiment analysis. Hashtags can also be expanded and queried in sentiment lexicons to find if the hashtag itself conveys an opinion. For example, #sarcasm conveys sarcasm and usually negative sentiment, #happy conveys positive sentiment, #lonely conveys negative sentiment, and so on. Hashtags can also be one of the most discriminating features for topic-based sentiment analysis.

- **Elongations:** Repeating letters several times in a word can intensify the sentiment expressed by the word. For example, “I am sooooo happyyyyy" conveys a more intense positive sentiment than “I am so happy".

- **Punctuation:** Repeating punctuation marks such as repeated exclamation marks often increases the intensity of the sentiment. For example, “This is so great!!!!!!!!!!" often conveys a more intense positive sentiment than “This is so great!". Also, punctuation marks such as repeated question marks are often used to express worry or sorrow, thereby conveying a negative sentiment. An example of this, would be questioning God when something bad happens – “Why???? God, why did this happen to me???" or expressing anxiety – “I missed the deadline. What should I do now???"

2.2.2 Unsupervised Learning and Linguistic Rules For Sentiment Analysis

Using unsupervised learning usually involves extracting information from text and applying certain rules to determine which category it belongs to. Unsupervised sentiment analysis is usually based on using sentiment lexicons to find sentiment/opinion words in the text. Once the polarities of opinion words in a text have been found, the polarity of the whole phrase or sentence is calculated by using statistical formulae such as the pointwise mutual information (PMI) measure (see equation 2.1) that tells us the amount of information we acquire about the presence of one of the terms when we observe the other. For example, if a phrase contains the terms “wonderful" and “bad" conveying positive and negative sentiment respectively, the polarity of the phrase will be based on its association with the positive reference word “wonderful" and its association with the negative reference word “bad" (see equation 2.2).

$$PMI(term_1, term_2) = log_2\left(\frac{\Pr(term_1 \wedge term_2)}{\Pr(term_1).\Pr(term_2)}\right) \tag{2.1}$$

$$Polarity(phrase) = PMI(phrase, \text{“}wonderful\text{”}) - PMI(phrase, \text{“}bad\text{”}) \tag{2.2}$$

Liu and Zhang describe this method in greater detail in [38]. A comparison of supervised and unsupervised approaches for sentiment analysis of movie reviews can be found in [43]

2.2.3 Concept-level Sentiment Analysis and Sentic Computing

So far sentiment analysis approaches relying on keyword spotting, word co-occurrence frequencies, and bag-of-words have worked fairly well. However, with increase in user-generated content like microblogging text and the epidemic of deception phenomenon like web-trolling and opinion spam, these standard approaches are becoming progressively inefficient. Thus, sentiment analysis systems will eventually stop relying solely on word-level techniques and move onto concept-level techniques. Concepts can be single-word or multi-word expressions extracted from text. Multi-word expressions are often more useful for sentiment analysis as they carry specific *semantics and sentics* [9], which include common-sense knowledge (which people acquire during their early years) and common knowledge (which people gather in their daily lives). The survey in [44] explains how Natural Language Processing research is evolving from methods based on bag-of-words to bag-of-concepts and finally on bag-of-narratives. In this paper, we define linguistic rules which rely on polarity values from a concept-level common-sense knowledge base called SenticNet [45]. Common-sense knowledge has also been shown to be useful for tasks such as personality recognition [46].

Other techniques for sentiment analysis are beyond the scope of this thesis.

3 Combining Supervised Learning with a Rule-based Classifier

"Probability is expectation founded upon partial knowledge. A perfect acquaintance with all the circumstances affecting the occurrence of an event would change expectation into certainty, and leave nether room nor demand for a theory of probabilities."

– George Boole

3.1 Objective

SemEval 2015 Task 10 [1] is an international shared-task competition that aims to promote research in sentiment analysis of tweets by providing annotated tweets for training, development and testing. We created a sentiment analysis system to participate in the message-level task of this competition. The objective of the system is to label the sentiment of each tweet as *positive*, *negative* or *neutral*.

In this chapter, we describe our sentiment analysis system, which is a combined classifier created by integrating a rule-based classification layer with a support vector machine.

3.2 The Proposed Method

Our Sentiment Analysis System consists of two classifiers – (i) Rule-based and (ii) Supervised, integrated together. This section describes both these classifiers and how we combine them.

During pre-processing, all the *@<username>* references are changes to *@USER* and all the *URLs* are changed to *http://URL.com*. Then, we use the CMU Twitter Tokeniser and POS Tagger [47] to tokenise the tweets and assign a parts-of-speech tag to each token. We use the POS tags to remove all emoticons from the pre-processed tweets. Pre-processed tweets **with emoticons** are given as input to the rule-based classifier, whereas the support vector machine takes pre-processed tweets **without emoticons** as an input.

3.2.1 Supervised Learning

For the supervised classifier, we cast the sentiment analysis problem as a multi-class classification problem, where each tweet has to be labeled as *positive*, *negative* or *neutral*. We train a Support Vector Machine (SVM) [48] on the tweets provided for training. For all our experiments, we use a linear kernel with L1-regularisation. The C parameter is chosen by the holdout method of cross-validation. As mentioned above, emoticons have already been removed from tweets given as input to the SVM.

Each tweet is represented as a feature vector, containing the following features:

- **Word N-grams:** Frequencies of contiguous sequences of 1, 2 or 3 tokens. The TF-IDF [49] weighting scheme is applied.
- **Character N-grams:** Frequencies of contiguous sequences of 1, 2 or 3 characters inside each word's boundary. The TF-IDF [49] weighting scheme is applied.
- **POS Tags:** Using CMU Twitter Tagger [47] output, for each tweet we compute – (i) *countAdj* (number of adjectives), (ii) *countAdv* (number of adverbs), (iii) *countNoun* (number of nouns, proper nouns, and proper nouns+possessives), (iv) *countVerb* (number of verbs), and (v) *countIntj* (number of interjections). The sum of these five counts, gives us the *totalPos*. The POS features are: $[\frac{countAdj}{totalPos}, \frac{countAdv}{totalPos}, \frac{countNoun}{totalPos}, \frac{countVerb}{totalPos}, \frac{countIntj}{totalPos}]$.
- **@USER:** A boolean feature that is set to 1 if the tweet contains a @<*username*> reference.
- **Hashtag:** A boolean feature that is set to 1 if the tweet contains a hashtag.

- **URL:** A boolean feature that is set to 1 if the tweet contains a URL.
- **Discourse:** A boolean feature that is set to 1 if the tweet contains a "discourse marker". Examples of discourse markers would be a "RT" followed by a username to indicate that the tweet is a re-tweet, news article headline followed by "..." followed by a URL to the news article, etc. Basically, this feature indicates whether or not the tweet is a part of a discourse.
- **Sentiment140 Lexicon:** The Sentiment140 Lexicon [39] contains unigrams and bigrams along with their polarity scores in the range of -5.00 to $+5.00$. Considering all uni/bi-grams with polarity less than -1.0 to be negative and with polarity greater than $+1.0$ to be positive, we count the number of negative (*negativesCount*) and the number of positive (*positivesCount*) uni/bi-gram occurrences in every tweet. For each tweet,
  - the *polarityMeasure* is based on the *positivesCount* and *negativesCount*, and calculated using Algorithm 1.
  - the maximum polarity value (*maxPolarityValue*) is the most positive or most negative polarity value of all polar uni/bi-gram occurrences in the tweet.

  Both these features are normalised to values between -1 and $+1$.

Algorithm 1 Calculating *polarityMeasure* based on *positivesCount* and *negativesCount*

if positivesCount $>$ negativesCount **then**
  if negativesCount $!=0$ **then**
    polarityMeasure $= \frac{positivesCount}{negativesCount}$
  else
    polarityMeasure = positivesCount
  end if
else if negativesCount $>$ positivesCount **then**
  if positivesCount $!=0$ **then**
    polarityMeasure $= -1 \times \frac{negativesCount}{positivesCount}$
  else
    polarityMeasure $= -1\times$ negativesCount
  end if
end if

- **Bing Liu Lexicon:** The Bing Liu lexicon [50] is a list of positive and negative words. We count the number of positive (*positivesCount*) and negative words (*negativesCount*) in each tweet, and calculate *polarityMeasure* using Algorithm 1. The *polarityMeasure* is appended to the feature vector.

- **NRC Emotion Lexicon:** The NRC Emotion Lexicon [51] contains a list of positive and negative words. The *polarityMeasure* is calculated using the method used for the Bing Liu Lexicon.

- **NRC Hashtag Lexicon:** The NRC Hashtag Lexicon [39] contains unigrams and bigrams along with their polarity scores in the range of -5.00 to $+5.00$. Using the method used for the Sentiment140 Lexicon, we calculate *polarityMeasure* and *maxPolarityValue*, and append them to the feature vector.

- **SentiWordNet:** SentiWordNet [52] assigns to each synset of WordNet [53] three scores: positivity, negativity, objectivity. A word whose positivity score is greater than negativity and objectivity is positive, while a word whose negativity score is greater than positivity and objectivity is negative. For each tweet, we calculate *polarityMeasure* and *maxPolarityValue* using the method used for the Bing Liu Lexicon.

- **SenticNet:** SenticNet [45] contains polarity scores of single and multi-word phrases. We count the number of positive and negative words/phrases in each tweet, and calculate *polarityMeasure* using the method used for the Sentiment140 Lexicon.

- **Negation:** The Stanford Dependency Parser [54] is used to find negation in tweets. Negation is not a feature on its own. Rather, it affects the word n-grams and the lexicons related features. The negated word is appended with a "_NEG" in all n-grams, while the polarity of all negated words is inverted in the lexicon features.

3.2.2 Rule-based Classifier

For the rule-based classifier, we cast the problem as a multi-class classification problem, where each tweet is to be labeled as *positive*, *negative*, or *unknown*. This is an unsupervised classifier, which applies the following rules for predictions:

- **Emoticon-related Rules:** If a tweet contains only positive emoticons and no negative emoticons, it is classified as positive. If a tweet contains only negative emoticons and no positive emoticons, it is classified as negative. If a tweet contains no emoticons, we apply the sentiment lexicon-related rules. The following emoticons are considered to be positive: :) , (: , ;) , :-) , (-: , :D , :-D , :P , :-P . While, the following emoticons are considered to be negative: :(,): , ;(, :-(,)-: , D: , D-: , :'(, :'-(,)': ,)-': .

- **Sentiment Lexicon-related Rules:** The Bing Liu lexicon [50], the NRC Emotion lexicon [51], and SentiWordNet [52] are used as resources for positive and negative opinion words. If a tweet contains **more than two** positive words, and no negation or negative words from either of the lexicons, it is classified as positive. If a tweet contains **more than two** negative words, and no negation or positive words from either of the lexicons, it is classified as negative. If none of the above rules apply, the tweet is classified as unknown.

3.2.3 Combining the Classifiers

After developing the rule-based classifier and training the SVM, we combine the them to refine the SVM's predictions. Since, our goal is to maximise positive and negative precision and recall, we use the rule-based classifier to correct or verify the "neutral" SVM predictions. So, for every tweet labeled as neutral by the SVM, we consider the predictions of the rule-based layer as the final labels.

3.3 Experiments and Results

We trained a Support Vector Machine (SVM) on 9418 tweets allowed by the shared-task organisers to be used for training purposes. The results we submitted to SemEval 2015 were yielded by using all SVM features and emoticon-related rules. The sentiment lexicon-related rules were implemented later, and thus could not be used for the official submission. Table 3.1 shows the official test results for SemEval 2015.

Dataset	Our Score	Best Score
Twitter 2015	57.06	64.84
LiveJournal 2014	68.70	75.34
Twitter 2014	66.85	74.42
Twitter 2013	63.50	72.80
SMS 2013	60.53	68.49
Twitter 2014 Sarcasm	45.18	57.50

Table 3.1 – Average positive and negative F-scores for system with all SVM features and only emoticon rules.

Table 3.2 reports the results of a feature ablation study carried out by testing the SVM classifier on 3204 development tweets (from SemEval 2013) not included in the training data. These are cross-validation results obtained using the hold-out method.This study helps us understand the importance of different features. From the table, we can see that the word and character n-grams features are the most useful, followed by negation and then the rest. All sentiment lexicon related features appear to have similar importance, but we get the best F-score when we append them all to the feature vector.

Features	Positive			Negative			Neutral			F_{pn}
	P	R	F	P	R	F	P	R	F	
All Features	0.824	0.629	0.713	0.612	0.607	0.610	0.679	0.831	0.748	0.662
w/o N-grams	0.671	0.597	0.632	0.430	0.574	0.491	0.645	0.637	0.641	0.562
w/o POS Tags	0.814	0.611	0.698	0.633	0.589	0.610	0.669	0.839	0.744	0.654
w/o @User, Hashtag, URL, Discourse	0.821	0.616	0.704	0.602	0.607	0.605	0.672	0.826	0.741	0.654
w/o Sentiment140	0.814	0.616	0.701	0.602	0.599	0.600	0.676	0.830	0.745	0.651
w/o Bing Liu	0.821	0.621	0.707	0.616	0.603	0.610	0.676	0.833	0.746	0.658
w/o NRC Emotion + Hashtag	0.816	0.619	0.705	0.609	0.597	0.603	0.676	0.832	0.746	0.654
w/o SentiWordNet	0.821	0.624	0.709	0.610	0.597	0.603	0.674	0.830	0.744	0.656
w/o SenticNet	0.820	0.615	0.703	0.610	0.597	0.603	0.674	0.837	0.747	0.653
w/o Negation	0.811	0.610	0.701	0.598	0.601	0.593	0.674	0.824	0.744	0.647

Table 3.2 – Feature ablation study for the SVM classifier. Each row shows the precision, recall, and F-score for the positive, negative, and neutral classes respectively, followed by the average positive and negative F-score, which is the chosen evaluation metric. All values in the table are between 0 and 1, and are rounded off to 3 decimal places.

Since, using all the previously described features gives the best SVM predictions, we add the rule-based classification layer to a SVM trained on all features. Table 3.3 compares the results obtained using the SVM alone with the results obtained using SVM along with all the rules (emoticon and lexicon-based) specified in section 3.2.2. We observe that the F-score further increases by around half a unit and the classification rate[1] increases by around 0.8.

Features	F_{pn}	**Classification Rate (%)**
All Features	66.2	71.5
All Features and Rules	66.7	72.3

Table 3.3 – Comparison between the results obtained using SVM alone, and using SVM with a rule-based layer.

[1] Classification rate = $\frac{number\ of\ tweets\ classified\ correctly}{total\ number\ of\ tweets}$

4 Using Linguistic Patterns and Sentic Computing to Enhance Supervised Learning

"Although many texters enjoy breaking linguistic rules, they also know they need to be understood."

– David Crystal, *Txtng: The gr8 db8*

4.1 Motivation

Supervised learning classifiers commonly used for polarity classification rely on feature vectors extracted from the text to represent the most important characteristics of the text. Word N-grams, which are denoted by the frequencies of contiguous sequences of 1, 2, or 3 tokens in the text, are the most commonly used features for supervised sentiment analysis. While such classifiers [55, 56, 57] have been shown to perform reasonably well, studies such as [58], [59] and [60] show that using a "one-technique-fits-all" solution for all types of sentences is not good enough due to the diverse types of linguistic patterns found in sentences. That is, the presence of modal verbs like "could" and "should", conjunctions like "but" and "or" and conditionals like "if", "until", "unless", and "in case" in a text substantially worsen the predictions of a supervised classifier.

Furthermore, supervised learning classifiers classify each tweet with a certain probability or decision (confidence) score. For a large number of tweets, the decision score predicted by a typical supervised classifier is very close to the decision boundary. This implies that the classifier is unsure about which class the tweets in question belong to and so cannot assign class labels to them with much confidence. Thus, the class labels assigned to such tweets are either completely incorrect or correct mostly by fluke.

To prove this notion, we train a Support Vector Machine (SVM) classifier using n-grams (n = 1,2,3) as features on ≈ 1.6 million tweets provided by [3] and test it on 1794 positive and negative tweets provided by [2] and plot the decision scores computed by the SVM in figure 4.1. In the graph, we can see that frequency of misclassifications reduce as we move away from the decision boundary ($y = 0$). We find that 341 tweets out of 1794 tweets are misclassified by the SVM, however 239 out of the 341 misclassified tweets have a decision score that lies between -0.5 and $+0.5$. Thus, the SVM is simply unsure instead of being completely wrong about these 239 tweets. If we consider all the predictions of the SVM, we get a misclassification rate[1] of ≈ 19%. But, if we exclude all the predictions whether right (475 tweets) or wrong (239 tweets) with a decision score between -0.5 and $+0.5$, we get a misclassification rate of only ≈ 9.4%. This means that if we consider the classification of only those tweets that the SVM is confident about, we can say that it correctly classifies over 90% of the tweets!

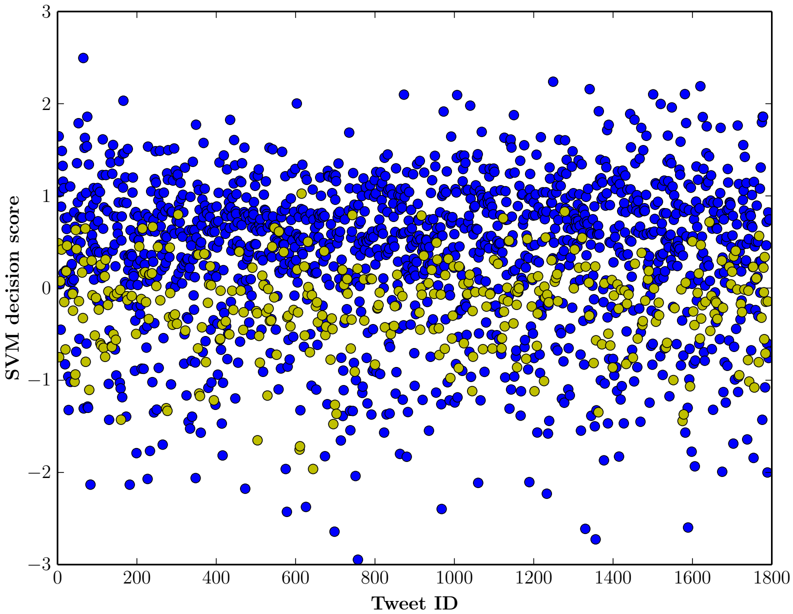

Figure 4.1 – SVM decision scores plotted for the classification of 1794 tweets into positive or negative using n-grams (n=1,2,3) as features. Tweets with decision score above 0 are labelled as positive, while tweets with decision score below 0 are labelled as negative. Blue = correctly classified (1453 tweets), Green = misclassified (341 tweets). 239 out of the 341 misclassified tweets have decision score between -0.5 and +0.5, implying that the SVM is simply unsure about them.

[1]misclassification rate = $\frac{Number\ of\ Incorrect\ Classifications}{Total\ Number\ of\ Classifications}$

So, from the above, we can deduce that it would be beneficial to design a classifier that:

- Can handle special parts-of-speech of grammar like conjunctions and conditionals.
- Uses a secondary (high-confidence) classifier to verify or change the classification labels of the tweets the SVM computes a very low decision or confidence score for.

To handle the special parts-of-speech of grammar, we modify the n-gram features provided as input to the classifier, based on linguistic analysis of how these parts-of-speech are used in sentences. The scope of the method proposed in this thesis is limited to the conjunction “but" and the conditionals “if", “unless", “until" and “in case".

Furthermore, we design an unsupervised rule-based classifier to verify or change the classification labels of the tweets the SVM computes a very low decision score for. The rules used by this classifier are based on our linguistic analysis of tweets, and leverage on sentiment analysis resources that contain polarity values of words and phrases. The primarily resource used for this purpose is SenticNet [45] – a semantic and affective resource for concept-level sentiment analysis, which basically assigns polarity values to concepts taken from a common-sense knowledge base called ConceptNet [61].

As human beings, we are able to understand the meaning of texts and determine the sentiment conveyed by them. Our common-sense plays a very important role in this process by helping us estimate the polarities of commonly used single-word and multi-word expressions or concepts occurring in text, and then use the relationships between words and concepts to ascertain the overall polarity. For example, say a text contains the phrase “good morning"; how do you interpret it and estimate its polarity? Luckily, depending on the context, our common-sense helps us deduce whether the expression “good morning" is used as a wish, as a fact, or as something else. Otherwise, without common-sense, we would need to ask each other questions like...

> *“Do you wish me a good morning, or mean that it is a good morning whether I want it or not; or that you feel good this morning; or that it is a morning to be good on?"* – J.R.R. Tolkien (from The Hobbit)

Moreover, the estimated polarity of the expression "good morning" cannot merely be the sum of the polarities of the words "good" and "morning". Hence, unlike most sentiment analysis methods, we prefer to break tweets into concepts and query those concepts in SenticNet, instead of relying completely on bag-of-words queried in lexicons containing word-level polarities.

4.2 The Proposed Method

Before analysing raw tweets for sentiments, we pre-process them. During pre-processing, all the *@<username>* references are changes to *@USER* and all the *URLs* are changed to *http://URL.com*. Then, we use the CMU Twitter Tokeniser and Parts-of-Speech Tagger [47] to tokenise the tweets and assign a parts-of-speech tag to each token. Apart from nouns, verbs, adjectives and adverbs, this tagger is also able to tag injunctions, and microblogging-specific tokens such as emoticons, hashtags, and URLs.

The proposed sentiment analysis system is illustrated in figure 4.2, and is explained in detail in this section.

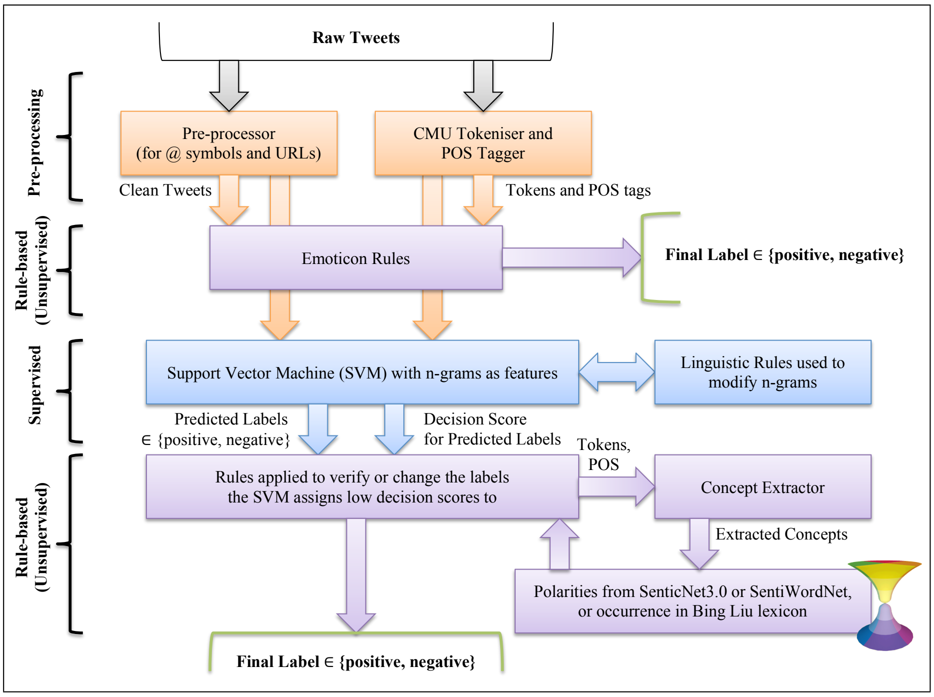

Figure 4.2 – Flowchart of the Proposed Twitter Sentiment Analysis System.

4.2.1 Emoticon Rules

Using the tokens in a tweet and the output of the tagger, we are able to find all the tokens that represent emoticons in the tweet. Since people often repeat certain punctuations to emphasise emoticons, we remove all repeated characters from every emoticon string to obtain the bag-of-emoticons present in the tweet. Table 4.1 is a manually created list of usually polar emoticons along with their semantic orientation (*positive or negative*). We match emoticons in the bag-of-emoticons of the tweet to the list of positive or negative emoticons, and count the number of positive and the number of negative emoticons present in the tweet.

Orientation	**List of Emoticons**
Positive	(-: , (: , =) , :) , :-) , =‘) , :‘) , :‘-) , =-d , =d , ;d , :d , :-d , ^ −^ , ^_^ , :] , ^_- , ^_* , ^^
Negative	)-: ,): , =(,]: , :[, :(, :-(, >;(, >:(, :_(, d’x , :‘(, :"(, =’[, :’(, :’-(, \: , :/ , (~_~) , >__> , <(’-’)> , </3

Table 4.1 – Manually Created List of Positive and Negative Emoticons.

Then, we apply the following rules to classify the tweet:

- If a tweet contains one or more positive emoticons and no negative emoticons, it is labeled as *positive*.
- If a tweet contains one or more negative emoticons and no positive emoticons, it is labeled as *negative*.
- If neither of the two rules above apply, the tweet is labeled as *unknown*.

If these emoticon-based rules label a tweet as *positive* or *negative*, we consider that label to be the final label outputted by our system. However, all tweets labelled as *unknown* by these rules are passed into the next stage in our sentiment analysis pipeline, that is the supervised learning classifier.

4.2.2 Support Vector Machine (SVM) with N-grams as Features

For supervised learning, we represent each tweet as a feature vector of case-sensitive n-grams (unigrams, bigrams, and trigrams). These n-grams are frequencies of sequences of 1, 2 or 3 contiguous tokens in a tweet. The TF-IDF [49] weighting scheme from information retrieval is applied to the frequency counts, and L1 regularisation is used for feature selection and dimensionality reduction. Finally, a Support Vector Machine is trained using the LIBLINEAR library [48].

To account for negation, we append the string "_NEG" to all negated tokens in a tweet. All tokens between certain negation words and the next punctuation mark are considered to be negated, as long as they are either nouns, adjectives, adverbs, or verbs. This is so because negating emoticons, hashtags or URLs would not make sense. Apart, from this, no other feature related to negation is used in the feature vector.

For this purpose, we take into account the following negation words: *never, no, nothing, nowhere, noone, none, not, havent, haven't, hasnt, hasn't, hadnt, hadn't, cant, can't, couldnt, couldn't, shouldnt, shouldn't, wont, won't, wouldnt, wouldn't, dont, don't, doesnt, doesn't, didnt, didn't, isnt, isn't, arent, aren't, aint, ain't.*

In section 4.2.4, the same method will be to find negated tokens in order to invert their polarity values.

4.2.3 Modifying N-grams according to Linguistic Rules

As mentioned in section 4.1, typical supervised learning methods based on n-grams perform badly on sentences containing special parts-of-speech such as conjunctions and conditionals commonly used in grammar, due to their peculiar linguistic characteristics. We theorise that one such characteristic is that a certain part of the sentence either becomes irrelevant for sentiment analysis or possesses a semantic orientation that is opposite to the sentence's overall orientation.

We analyse tweets containing the conjunction "but" and the conditionals "if", "unless", "until", and "in case", and formulate rules that should enable removal of irrelevant or oppositely oriented n-grams from the tweet's feature vector, before it is used for supervised learning.

Below are a few examples of tweets containing "but" at different syntactic positions. In each tweet, the most salient part that is the part that contributes considerably to the overall polarity of the tweet is underlined. In certain tweets however, if no salient part can be found or is ambiguous, nothing is underlined. The overall polarity of the tweet is indicated in parenthesis.

(1) @USER Tell you at our Friday lunch. Sorry for the late reply but yes we can eat somewhere on Marshall tomorrow haha *(positive)*

(2) it may have been against the minnows of FC Gomel, but a great performance from Rodger's Reds at Anfield tonight, and a great start! *(positive)*

(3) SP to support UPA, but oppose anti-people policies: Samajwadi Party on Saturday said it will continue to oppose *(negative)*

(4) Taylor Kitsch may not be a leading man, or able to open a movie, but he was quite good in The Bang Bang Club- Ryan Phillippe as well *(positive)*

(5) S/O to @USER ! I don't really know her but she seems real chill. She may not know how to spell Peyton Siva, but still follow her! *(positive)*

(6) you didnt win ABDC but you won over my heart you may not know me but imma true ICONiac by heart *(positive)*

(7) I laughed a little, but not sharing them out with anyone. Will the weather be good tomorrow for Boris Bikes? *(positive)*

(8) Gutted I'm missing the cardigan match on Saturday! But more important things to do *(negative)*

From the examples above, we observe that the part of the sentence posterior to the word “but" is usually (though not always) a better indicator of the overall polarity of the tweet, as compared to the anterior part. This premise holds true for examples (1) to (6), but does not work for a few examples such as (7) and (8).

In example (7), it is difficult to determine the most salient part of the tweet. This could be because that tweet appears to be only weakly positive, and could even be interpreted as negative if we only consider the posterior part of “but". In example (8), the most salient part of the tweet is anterior to the word “but", perhaps because the polarity of the posterior part is too subtle or even neutral. Nevertheless, in this thesis, we will only focus on formulating rules that work for tweets similar to examples (1) through (6), as handling tweets similar to (7) and (8) is too difficult and requires more complex linguistic analysis which is beyond the scope of this study.

Furthermore, it is difficult to automatically determine the salient part in tweets similar to example (6), due to grammatical errors introduced by the writer of the tweet. That is, example (6) should contain 2 separate sentences, but there is no punctuation mark to separate “...my heart" and “you may not know me...", which makes it very hard for us to pick out the phrases “you won over my heart" and “imma true ICONiac by heart" as the most salient parts. Hence, to best handle such tweets, if there are more than one “but"s in the same sentence of a tweet, only the part posterior to the last occurrence of the word “but" is to be considered as the most salient.

Hence, we propose the following strategy to modify n-grams for tweets containing the conjunction “but":

1. We use the Punkt sentence tokeniser [62] to break a tweet into sentences.
2. In each sentence, we find the location of the last occurrence of the word “but"
3. We remove all tokens except the tokens posterior to (occurring after) that location. So, the modified sentence only contains tokens succeeding the last “but".
4. Once we have processed all the sentences in the tweet, we merge the modified sentences together to obtain the modified tweet.

Moving forwards, below are a few examples of tweets containing “if" at different syntactic positions. In each tweet, the most salient part that is the part that contributes considerably to the overall polarity of the tweet is underlined. In certain tweets however, if no salient part can be found or is ambiguous, nothing is underlined. The overall polarity of the tweet is indicated in parenthesis.

(1) If Gerald Green doesn’t have the most hops in the league then he definitely is a strong 2nd!! *(positive)*

(2) If you’re not coming to the SVSU vs. Wayne State game tomorrow, watch it on CBS College Sports or FSN Detroit. It’s about to be hype! *(positive)*

(3) If the Lakers still had Jordan Farmar,Trevor Ariza,&Shannon Brown I’d be watching them ..I dont like the Lakers but they were entertaining. *(positive)*

(4) if you follow @USER ill make you my famous Oreo brownie on Sunday!!! *(positive)*

(5) Juniors playing powderpuff, if you aren’t at practice tomorrow you will NOT play, it starts at 5:30pm, hope to see you there! *(negative)*

(6) @USER can you please come to SXSW in Austin in March? I’ve wanted to see you for years & it would be amazing if you played a show here! *(positive)*

From the above examples, we can see that as compared to “but", “if" has many more syntactic positions, such as:

(i) if <condition clause> then <consequent clause>

(ii) if <condition clause>, <consequent clause>

(iii) if <condition clause> <missing then/comma, or other> <consequent clause>

(iv) <consequent clause> if <condition clause>

According to syntax, example (1) is of type (i), example (2), (3) and (5) are of type (ii), example (4) is of type (iii), and example (6) is of type (iv). In examples (1) and (2), the most salient part of the tweet is the part that occurs after "then" or after the comma (,). Even in example (3), the part just after the first comma succeeding the "if" includes the most salient part of the tweet. However, example (3) contains both "if" and "but", which makes it harder to automatically determine the most salient part.

Moreover, in examples (4) and (5), the most salient part is not preceded by a "then" or comma, due to grammatical errors introduced by the writer or due to the informal nature of tweets. In example (6), "if" occurs in the middle of the sentence, such that even though the consequent clause usually precedes the "if" in such cases, it is hard to automatically determine the scope of the most salient part. Hence, determining the most salient part in tweets similar to (4), (5), and (6) requires more complex linguistic analysis which is beyond the scope of this study.

In this thesis, we will only focus on tweets similar to (1), (2), and (3). Also, while the examples above are only limited to the conditional "if", we will also handle the conditionals "unless", "until" and "in case". For these conditionals, we consider the most salient part of the tweet to be the part that occurs after the first comma succeeding the conditional, whereas for "if" we consider the part occurring after "then" as well as the comma.

Therefore, we propose the following strategy to modify n-grams for tweets containing the conditionals "if", "unless", "until" and "in case":

1. We use the Punkt sentence tokeniser [62] to break a tweet into sentences.

2. In each sentence, we find the location of the last occurrence of the conditional ("if", "unless", "until" or "in case")

3. Then, we find the location of the first comma (and also "then" in case of "if") that

occurs after the conditional.

4. We remove all tokens between the conditional and the comma/“then" including the conditional and the comma/“then". All the remaining tokens now make up the modified sentence.

5. Once we have processed all the sentences in the tweet, we merge the modified sentences together to obtain the modified tweet.

In case a tweet contains a conditional as well as the conjunction “but", only “but" rules are applied.

Finally, using the modified tweets, we create new feature vectors containing modified unigrams, bigrams, and trigrams for each tweet. These modified n-grams are then provided as input to the Support Vector Machine (SVM) specified in section 4.2.2, instead of the n-grams that are typically used.

4.2.4 Tweaking SVM Predictions using Linguistic Rules and Sentic Computing

During training, a Support Vector Machine (SVM) approximates a hyperplane or decision boundary that best separates data points (feature vectors of samples) belonging to n different classes (feature vectors of samples = n-grams of tweets, n = 2 and class $\in$ {positive, negative} in our case). The data points that “support" this hyperplane on either sides are known as support vectors.

Each trained SVM has a scoring function that computes the decision score for each new sample, based on which the class label is assigned. The SVM decision score for classifying a sample is the signed distance from the sample's feature vector x to the decision boundary, and is given by:

$$SVM\ Decision\ Score\ = \sum_{m}^{i=1} \alpha_i y_i G(x_i, x) + b \tag{4.1}$$

where $\alpha_1, \alpha_2, ...\alpha_n$, and b are the parameters estimated by the SVM, $G(x_i, x)$ is the dot product in the predictor space between x and the support vectors, and m is the number of training samples.

As explained in section 4.1, the decision score for a large number of tweets is too low, implying that the SVM is unsure about the label it assigns to them, because their feature vector lies very close to the decision boundary. Hence, after running the supervised classifier on all the unlabelled tweets, we get the decision score computed by it for each tweet to determine the confidence of the SVM's predictions.

For tweets with an absolute decision score or confidence below 0.5, we discard the class labels assigned by the SVM and instead use an unsupervised classifier to predict their class labels. This unsupervised classification process works as follows:

1. The tweets are modified using the method describes in section 4.2.3, in order to take into account conjunctions and conditionals.
2. Single-word and multi-word concepts are extracted from the tweets in order to fetch their polarities from SenticNet [45]. These concepts are extracted using algorithm 2.
3. Then, we query all these concepts in SenticNet in order to get their polarities. If a single-word concept is not found in SenticNet, it is queried in SentiWordNet [52], and if it is not found in SentiWordNet, it is searched for in the list of positive and negative words from the Bing Liu lexicon [50]. The number of positive and negative concepts, and the polarity of the most polar concept is noted as the tweet's most polar value. The Bing Liu lexicon only contains a list of around 2000 strongly positive and 4800 strongly negative words, and no polarity values. So, the polarity of all positive words in the Bing Liu lexicon is assumed as $+1.0$ while the polarity of all negative words is assumed as -1.0. The polarities of the negated concepts is inverted. Negated concepts are those that precede the negation words specified in section 4.2.2.
4. Based on the number of positive and negative concepts, and the most polar value

[2] "N" = Noun, "V" = Verb, "A" = Adjective, "R" = Adverb, "P" = Preposition

Algorithm 2 Given a list of *tokens* in a tweet and a list of their corresponding POS tags, this algorithm extracts a bag-of-concepts from tweets

```
token1 = []; pos1 = [];
{First, remove all stop words from the tweet tokens}
for each token, tag in tokens, pos do
   if token is NOT a stop word then
      append token to token1 and tag to pos1
   end if
end for
concepts = []
{adjacent tokens with the following POS tags2 are extracted as multi-word concepts}
conceptTagPairs = [("N", "N"), ("N", "V"), ("V", "N"), ("A", "N"), ("R", "N"), ("P", "N"), ("P",
"V")]
for ti in range(0, len(tokens1)) do
   token = tokens1[ti]; tag = pos1[ti];
   prevtoken = tokens1[ti-1]; prevtag = pos1[ti-1];
   token_stem = Stem(token); prevtoken_stem = Stem(prevtoken);
   {raw tokens and stemmed tokens are extracted as single-word concepts}
   append token to concepts
   append token_stem to concepts
   if (prevtag, tag) in conceptTagPairs then
      append prevtoken+" "+token to concepts
      append prevtoken_stem+" "+token_stem to concepts
   end if
end for
```

occurring in the tweet, the following rules are applied to classify it:

- If the number of positive concepts is greater than the number of negative concepts and the most polar value occurring in the tweet is greater than or equal to 0.6, the tweet is labelled as positive.
- If the number of negative concepts is greater than the number of positive concepts and the most polar value occurring in the tweet is less than or equal to –0.6, the tweet is labelled as negative.
- If neither of the two rules stated above apply, the tweet is labeled as unknown by the rule-based classifier, and the SVM's low confidence predictions are taken as the final output of the system.

4.3 Experiments and Results

We train our SVM [48] classifier on around 1.6 million positive and negative tweets provided by [3]. First, the training data is divided into 80% train and 20% validation sets, and the "c" parameter is selected as 0.4 through 10-fold cross-validation. Then, the model is trained on 100% of the training data.

We evaluate our proposed method on two publicly available datasets – SemEval 2013 [2] test set and SemEval 2014 [2] test set. Neutral tweets are removed from each dataset, which leaves 1794 and 3584 positive/negative tweets in the SemEval 2013 and SemEval 2014 datasets respectively. Tables 4.2 and 4.3 show the results obtained on these two datasets. In these tables, each row shows the precision (P), recall (R), and F-score for the positive, and negative classes, followed by the average positive and negative precision, recall, and F-score. All values in the tables are between 0 and 100, and are rounded off to 2 decimal places. This section will focus on discussing and analysing the results shown.

In order to gauge the effectiveness of our method, we consider averaged positive and negative F-score (F_{avg}) as the primary evaluation metric, and the standard n-grams based supervised model as a benchmark. It is important to note that apart from TF-IDF weighed frequency counts of n-grams, this standard n-grams benchmark model also takes negation into account. We use L1 regularisation for feature reduction. The n-grams feature space is reduced from 12548966 features to just 233 significant features.

On comparing the standard n-grams model with the n-grams and emoticon rules model,

Method	**Positive**			**Negative**			**Average**		
	P	R	F	P	R	F	P	R	F
N-grams	90.48	82.67	86.40	61.98	76.45	68.46	76.23	79.56	77.43
N-grams and Emoticon Rules	90.62	83.36	86.84	62.99	76.65	69.15	76.80	80.00	78.00
Modified N-grams	89.95	84.05	86.90	63.33	74.59	68.50	76.64	79.32	77.70
Modified N-grams, and Emoticon Rules	90.10	84.73	87.33	64.41	74.79	69.22	77.26	79.76	78.27
Modified N-grams, Emoticon Rules, and Word-level Unsupervised Rules	91.40	86.79	89.04	68.55	77.89	72.92	79.97	82.34	80.98
Modified N-grams, Emoticon Rules, and Concept-level Unsupervised Rules	92.42	86.56	89.40	68.96	80.79	74.41	80.69	83.68	81.90

Table 4.2 – Results obtained on 1794 positive/negative tweets from the SemEval 2013 dataset.

Method	Positive			Negative			Average		
	P	R	F	P	R	F	P	R	F
N-grams	89.92	81.90	85.72	61.20	75.66	67.67	75.56	78.78	76.69
N-grams and Emoticon Rules	89.74	83.05	86.27	62.50	74.85	68.11	76.12	78.95	77.19
Modified N-grams	89.39	82.90	86.02	62.00	73.93	67.44	75.69	78.41	76.73
Modified N-grams, and Emoticon Rules	89.25	83.97	86.53	63.29	73.22	67.89	76.27	78.60	77.21
Modified N-grams, Emoticon Rules, and Word-level Unsupervised Rules	90.22	86.24	88.19	67.37	75.25	71.09	78.80	80.75	79.64
Modified N-grams, Emoticon Rules, and Concept-level Unsupervised Rules	90.41	86.20	88.25	67.45	75.76	71.37	78.93	80.98	79.81

Table 4.3 – Results obtained on 3584 positive/negative tweets from the SemEval 2014 dataset.

we can see that emoticon rules increase F_{avg} by 0.57 and 0.50 in the 2013 and 2014 datasets respectively. Comparison between the modified n-grams model, and modified n-grams and emoticon rules model also shows that emoticon rules increase F_{avg} by 0.57 and 0.48 in the two datasets respectively. Thus, this shows that the emoticon rules formulated by us significantly improve sentiment analysis.

Modifying n-grams using linguistic rules for conjunctions and conditionals increases F_{avg} by 0.27 and 0.04 in the two datasets respectively. While the increase is not very significant for the 2014 dataset, modified n-grams are still better than standard n-grams as (i) they do increase the overall F_{avg} and the increase is quite significant in the 2013 dataset, (ii) a typical Twitter corpus contains a very small percentage of tweets with such conjunctions and conditionals, and hence even a small improvement is very encouraging.

Next, we observe the results obtained by tweaking the SVM's predictions using the method specified in section 4.2.4. In this, we also compare the results obtained by using a bag-of-concepts model to the results obtained by using a bag-of-words (or single-word concepts only) model. We see that the F_{avg} of the bag-of-concepts model is 0.92 more than the bag-of-words model for the 2013 dataset, and 0.17 more than the bag-of-words model for the 2014 dataset. So, even though the effect of moving to concept-level sentiment analysis from word-level sentiment analysis will vary from one dataset to another, concept-level sentiment features will almost always perform better since they already include word-level sentiment features.

On comparing the results obtained by the modified n-grams and emoticon rules model with the modified n-grams, emoticon rules and concept-level unsupervised rules model, we see that tweaking the SVM's predictions using rules and sentic computing increases the F_{avg} by 3.63 and 2.6 in the two datasets respectively. Hence, this shows that the linguistic rules and sentic computing based secondary classifier proposed by us, substantially improve the result and is thus very beneficial for sentiment analysis.

Overall, our final sentiment analysis system achieves a F_{avg} score that is 4.47 units and 3.12 units higher than the standard n-grams model.

5 Conclusion and Future Work

John Connor: "If by feelings you mean emotions, I'm pretty sure you still don't have any of those. And if by feeling you mean what it feels like to have the wind blow through your toes or your hair, I'm pretty sure you can't feel that either."
Terminator: "I don't think you understand how we work. I have sensation. I feel. I wouldn't be worth much if I couldn't feel."

– Terminator: The Sarah Connor Chronicles

5.1 Summary

Chapter 3 describes a sentiment analysis system developed by combining a SVM with a rule-based classification layer. Even though we do not get the best scores, we find that a rule-based classification layer can indeed refine the SVM's predictions. We also devise creative twitter-specific, negation and lexicon-related features for the SVM, and demonstrate how they improve the sentiment analysis system.

Chapter 4 describes the pipeline of a Twitter sentiment analysis system that enhances supervised learning, by using modified features for supervised learning as well as applying rules based on linguistics and sentic computing. Based on our results, we can conclude that unsupervised emoticon rules and modified n-grams for supervised learning help improve sentiment analysis. They do so by handling peculiar linguistic characteristics introduced by special parts-of-speech such as emoticons, conjunctions and conditionals. Moreover, we have shown that verifying or changing the low-confidence predictions of a supervised classifier using a secondary rule-based (high-confidence, unsupervised) classifier is also immensely beneficial.

We created an online system to demonstrate the method described in chapter 4. The differences between the online system and the aforementioned method are as follows:

- Firstly, we removed tweets containing profanities and other inappropriate words.
- Secondly, we removed tweets containing more than 1 URL to minimise spam.
- Lastly, we only displayed tweets our system could label with a decision score $>=$ 0.5 or $<= -5$ in order to remove tweets the Supervised as well as the rule-based unsupervised classifiers failed to label with reasonable confidence

5.2 Ideas For The Future

In future, we would like to increase the performance of our method. We can do this in a number of ways, such as:

- Expanding common-sense knowledge bases like SenticNet [45], and using concept based text analysis like [63] in order to boost the predictions of the unsupervised classifier, and therefore improve the predictions of the whole system.
- Using enriched sentiment and emotion lists like the ones used by [64].
- Further analysing the linguistics of tweets to take into account other conjunctions like “or", conditionals like “assuming", or modal verbs like “can", “could" , “should", “will" and “would".
- Devising more sophisticated rules to improve the classification of tweets that the supervised classifier assigns a low decision score to.
- Incorporating common-sense vector space resources like [65] and [66].
- Increasing the number of labels to more than just positive or negative by using methods such as fuzzy clustering used by [67].

Using this project as inspiration, we can also apply a method based on linguistic rules and concept-level sentic polarities to other tasks relevant to sentiment analysis, such as depression detection [20] and dispute detection [18]. Moreover, the proposed approach can also be fed into a multimodal sentiment analysis framework like [68] and [36].

References

[1] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter, and V. Stoyanov, "Semeval-2015 task 10: Sentiment anal- ysis in twitter," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.

[2] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, and T. Wilson, "Semeval-2013 task 2: Sentiment analysis in twitter," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, (Atlanta, Georgia, USA), pp. 312–320, Association for Computational Linguistics, June 2013.

[3] A. Go, R. Bhayani, and L. Huang, "Sentiment140 twitter corpus." http://help.sentiment140.com/for-students.

[4] S. Rosenthal, A. Ritter, P. Nakov, and V. Stoyanov, "Semeval-2014 task 9: Sentiment analysis in twitter," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, (Dublin, Ireland), pp. 73–80, Association for Computational Linguistics and Dublin City University, August 2014.

[5] P. Chikersal, S. Poria, E. Cambria, A. Gelbukh, and C. E. Siong, "Modelling public sentiment in twitter: Using linguistic patterns to enhance supervised learning," in *Proceedings of the 16th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2015, Part II, LNCS 9042*, pp. 49–65, 2015.

[6] P. Chikersal, S. Poria, and E. Cambria, "Sentu: Sentiment analysis of tweets by combining a rule-based classifier with supervised learning," in *Proceedings of the International Workshop on Semantic Evaluation, SemEval 2015*, Association for Computational Linguistics, 2015.

[7] E. Cambria, S. Poria, F. Bisio, R. Bajpai, and I. Chaturvedi, "The CLSA Model: A novel framework for concept-level sentiment analysis," *LNCS*, vol. 9042, pp. 3–22, 2015.

[8] R. W. Picard, *Affective Computing.* Cambridge, MA, USA: MIT Press, 1997.

[9] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, 2013.

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

[11] G. Mishne and N. S. Glance, "Predicting movie sales from blogger sentiment.," in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pp. 155–158, 2006.

[12] E. Cambria, A. Hussain, C. Havasi, C. Eckl, and J. Munro, "Towards crowd validation of the uk national health service," *WebSci10*, pp. 1–5, 2010.

[13] C. Johnson, P. Shukla, and S. Shukla, "On classifying the political sentiment of tweets," *cs. utexas. edu*, 2012.

[14] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle," in *Proceedings of the ACL 2012 System Demonstrations*, pp. 115–120, Association for Computational Linguistics, 2012.

[15] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment.," *ICWSM*, vol. 10, pp. 178–185, 2010.

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

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

[18] L. Wang and C. Cardie, "A piece of my mind: A sentiment analysis approach for online dispute detection," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 693–699, 2014.

[19] G. Coppersmith, M. Dredze, and C. Harman, "Quantifying mental health signals in twitter," in *Proceedings of ACL Workshop on Computational Linguistics and Clinical Psychology. Association for Computational Linguistics*, 2014.

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

[21] J. Li, X. Wang, and E. Hovy, "What a nasty day: Exploring mood-weather relationship from twitter," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 1309–1318, ACM, 2014.

[22] N. Howard and E. Cambria, "Intention awareness: improving upon situation awareness in human-centric environments," *Human-centric Computing and Information Sciences*, vol. 3, no. 1, pp. 1–17, 2013.

[23] S. Poria, E. Cambria, A. Hussain, and G.-B. Huang, "Towards an intelligent framework for multimodal affective data analysis," *Neural Networks*, vol. 63, pp. 104–116, 2015.

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

[25] B. Liu, "Opinion mining and sentiment analysis," in *Web Data Mining*, pp. 459–526, Springer, 2011.

[26] N. Jindal and B. Liu, "Mining comparative sentences and relations," in *AAAI*, vol. 22, pp. 1331–1336, 2006.

[27] S.-M. Kim and E. H. Hovy, "Crystal: Analyzing predictive opinions on the web.," in *EMNLP-CoNLL*, pp. 1056–1064, Citeseer, 2007.

[28] S. Volkova, G. Coppersmith, and B. Van Durme, "Inferring user political preferences from streaming communications," in *Proceedings of the Association for Computational Linguistics (ACL)*, 2014.

[29] R. Mihalcea, C. Banea, and J. Wiebe, "Learning multilingual subjective language via cross-lingual projections," in *Annual Meeting-Association for Computational Linguistics*, vol. 45, p. 976, 2007.

[30] N. Malhotra, A. Vij, N. Nandan, and D. Dahlmeier, "Sap-ri: A constrained and supervised approach for aspect-based sentiment analysis," *SemEval 2014*, p. 517, 2014.

[31] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," *SocialNLP 2014*, pp. 28–37, 2014.

[32] C. Zhang, D. Zeng, J. Li, F.-Y. Wang, and W. Zuo, "Sentiment analysis of chinese documents: From sentence to document level," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 12, pp. 2474–2487, 2009.

[33] E. Cambria, A. Livingstone, and A. Hussain, "The hourglass of emotions," in *Cognitive behavioural systems*, pp. 144–157, Springer, 2012.

[34] E. Cambria, Y. Song, H. Wang, and N. Howard, "Semantic multidimensional scaling for open-domain sentiment analysis," *Intelligent Systems, IEEE*, vol. 29, no. 2, pp. 44–51, 2014.

[35] V. Loia and S. Senatore, "A fuzzy-oriented sentic analysis to capture the human emotion in web-based content," *Knowledge-Based Systems*, vol. 58, pp. 75–85, 2014.

[36] S. Poria, E. Cambria, N. Howard, G.-B. Huang, and A. Hussain, "Fusing audio, visual and textual clues for sentiment analysis from multimodal content," *Neurocomputing, doi:10.1016/j.neucom.2015.01.095*, 2015.

[37] L.-P. Morency, R. Mihalcea, and P. Doshi, "Towards multimodal sentiment analysis: Harvesting opinions from the web," in *Proceedings of the 13th international conference on multimodal interfaces*, pp. 169–176, ACM, 2011.

[38] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining Text Data*, pp. 415–463, Springer, 2012.

[39] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets," *arXiv preprint arXiv:1308.6242*, 2013.

[40] F. Benamara, C. Cesarano, A. Picariello, D. R. Recupero, and V. S. Subrahmanian, “Sentiment analysis: Adjectives and adverbs are better than adjectives alone.,” in *ICWSM*, 2007.

[41] M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A. Montoyo, “A survey on the role of negation in sentiment analysis,” in *Proceedings of the workshop on negation and speculation in natural language processing*, pp. 60–68, Association for Computational Linguistics, 2010.

[42] R. González-Ibáñez, S. Muresan, and N. Wacholder, “Identifying sarcasm in twitter: a closer look,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 581–586, Association for Computational Linguistics, 2011.

[43] P. Chaovalit and L. Zhou, “Movie review mining: A comparison between supervised and unsupervised classification approaches,” in *System Sciences, 2005. HICSS’05. Proceedings of the 38th Annual Hawaii International Conference on*, pp. 112c–112c, IEEE, 2005.

[44] E. Cambria and B. White, “Jumping nlp curves: A review of natural language processing research,” *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, 2014.

[45] E. Cambria, D. Olsher, and D. Rajagopal, “SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis,” in *Twenty-eighth AAAI conference on artificial intelligence*, pp. 1515–1521, 2014.

[46] S. Poria, A. Gelbukh, B. Agarwal, E. Cambria, and N. Howard, “Common sense knowledge based personality recognition from text,” in *Advances in Soft Computing and Its Applications*, pp. 484–496, Springer, 2013.

[47] O. Owoputi, B. O’Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith, “Improved part-of-speech tagging for online conversational text with word clusters.,” in *HLT-NAACL*, pp. 380–390, 2013.

[48] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[49] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

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

[51] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word–emotion association lexicon," *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, 2013.

[52] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of LREC*, vol. 6, pp. 417–422, Citeseer, 2006.

[53] C. Fellbaum, *WordNet.* Wiley Online Library, 1998.

[54] M.-C. De Marneffe, B. MacCartney, C. D. Manning, *et al.*, "Generating typed dependency parses from phrase structure parses," in *Proceedings of LREC*, vol. 6, pp. 449–454, 2006.

[55] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!," *ICWSM*, vol. 11, pp. 538–541, 2011.

[56] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining.," in *LREC*, vol. 10, pp. 1320–1326, 2010.

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

[58] S. Poria, E. Cambria, G. Winterstein, and G.-B. Huang, "Sentic patterns: Dependency-based rules for concept-level sentiment analysis," *Knowledge-Based Systems*, vol. 69, pp. 45–63, 2014.

[59] Y. Liu, X. Yu, B. Liu, and Z. Chen, "Sentence-level sentiment analysis in the presence of modalities," in *Computational Linguistics and Intelligent Text Processing*, pp. 1–16, Springer, 2014.

[60] R. Narayanan, B. Liu, and A. Choudhary, "Sentiment analysis of conditional sentences," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pp. 180–189, Association for Computational Linguistics, 2009.

[61] R. Speer and C. Havasi, "Conceptnet 5: A large semantic network for relational knowledge," in *The People's Web Meets NLP*, pp. 161–176, Springer, 2013.

[62] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006.

[63] B. Agarwal, S. Poria, N. Mittal, A. Gelbukh, and A. Hussain, "Concept-level sentiment analysis with dependency-based semantic parsing: A novel approach," *Cognitive Computation*, pp. 1–13, 2015.

[64] S. Poria, A. Gelbukh, E. Cambria, P. Yang, A. Hussain, and T. Durrani, "Merging senticnet and wordnet-affect emotion lists for sentiment analysis," in *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, vol. 2, pp. 1251–1255, IEEE, 2012.

[65] S. Poria, A. Gelbukh, E. Cambria, A. Hussain, and G.-B. Huang, "Emosenticspace: A novel framework for affective common-sense reasoning," *Knowledge-Based Systems*, vol. 69, pp. 108–123, 2014.

[66] E. Cambria, J. Fu, F. Bisio, and S. Poria, "AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 508–514, 2015.

[67] S. Poria, A. Gelbukh, D. Das, and S. Bandyopadhyay, "Fuzzy clustering for semi-supervised learning–case study: construction of an emotion lexicon," in *Advances in Artificial Intelligence*, pp. 73–86, Springer, 2013.

[68] S. Poria, E. Cambria, A. Hussain, and G.-B. Huang, "Towards an intelligent framework for multimodal affective data analysis," *Neural Networks*, vol. 63, pp. 104–116, 2015.