SMART SENTIMENT AND EMOTION ANALYSIS

by

SANMIT TATOBA DESAI

(Under the Direction of I. Budak Arpinar)

ABSTRACT

With the rising social conflicts in different parts of the world the need to understand the feelings and opinions of the general populous is also growing. Since surveying the whole population is both resource intensive and time consuming we can resort to more a modern approach to this problem. The bloom of social media, especially Mirco-Blogs has extended the horizon of information gathering. With this research, we aim to solve the problem of finding Emotion and Sentiment from the Mirco-Blogging platform Twitter. With 284 million monthly active users and 500 million Tweets generated per day, Twitter contributes to a significant chunk of the vocal population who are not afraid to voice their opinions. What we seek to provide with this work is a fast and accurate way to extract emotions and sentiments from the data Twitter offers. This research is a part of the vision of SMART (Social Media Analysis in Real Time) Barometer which will help us analyze and evaluate text data understanding social conflicts better.

INDEX WORDS: Emotions Analysis, Sentiment Analysis, NLP, Information Gathering, Data Mining, Text Mining, DBpedia

SMART SENTIMENT AND EMOTION ANALYSIS

by

SANMIT TATOBA DESAI

BE, University of Mumbai, India, 2011

A Thesis Submitted to the Graduate Faculty of the University of Georgia in Partial

Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2014

SMART SENTIMENT AND EMOTION ANALYSIS

by

SANMIT TATOBA DESAI

Major Professor:	I. Budak Arpinar
Committee:	Kzysztof J. Kochut Hamid R. Arabnia

Electronic Version Approved:

Julie Coffield
Interim Dean of the Graduate School
The University of Georgia
December 2014

DEDICATION

I dedicate my work to the victims of Israel – Gaza and Ukraine - Russia conflicts. Hope this humble work adds to the global effort of Peace on Earth. I would also like to dedicate this work to my family and friends and all my mentors, academic or otherwise, who gave me the belief to push the boundaries of my achievements.

ACKNOWLEDGEMENTS

I would like to thank Dr. Arpinar for his prudent and pragmatic leadership throughout in my journey at University of Georgia. I would also like to thank the team of Computational Linguistics from the Department of Artificial Intelligence at UGA for their inventive input. I would also like to appreciate the help of the students of Dr. Arpinar Nilayan, Mumita and Anuja and the class of AIS, fall 2014 for their valuable contribution.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS vi

LIST OF TABLES ix

LIST OF FIGURES x

CHAPTER

1 INTRODUCTION..1

1.1 Introduction 1

1.2 Contribution to the Field 3

2 BACKGROUND 4

2.1 Sentiment Analysis 4

2.2 Sentiment Analysis Impact on Social Media 6

2.3 Social Conflicts 7

3 RELATED WORK 9

3.1 Sentiment Analysis 9

3.2 Emotion Analysis 15

3.3 Negation 17

3.4 Summary of Related Work...18

4 SMART SENTIMENT EMOTION ANALYSIS 21

4.1 Problem Statement..21

4.2 Data Gathering and Pre-Processing for Word List.............................24

4.3 SMART Word List 25

4.4 Sentiment and Emotion Consolidation 27

5 ARCHITECTURE 29

5.1 High Level Architecture 29

5.2 Step 1: Data Gathering 30

5.3 Step 2: Pre-Processing Part 1 Cleaning Tweets 31

5.4 Step 2: Pre-Processing Part 2 Loading Word Lists 33

5.5 Step 3: Sentiment and Emotion Extraction Part 1 Emotion Labeling . 35

5.6 Step 3: Sentiment and Emotion Extraction Part 2 Valence Shifters and Intensifiers 39

6 RESULTS AND EVALUATIONS 42

6.1 Evaluation Procedure 42

6.2 Results 46

6.3 Inter-Evaluator Agreement 48

6.4 Valuable Observations 50

7 CONCLUSION AND FUTURE WORK 51

7.1 Conclusion 51

7.2 Future Work 52

REFERENCES 53

LIST OF TABLES

Page

Table 1: Summary of Related Word....19

Table 2: Opposite Emotions....23

Table 3: Combination of Emotions....23

Table 4: Entry of the word protest in NRC word list....36

Table 5: Entry of the word impossible in SMART word list....36

Table 6: Initial condition of the word in the Tweet....37

Table 7: word 'protest' after adding sentiment score from SentiWordNet....38

Table 8: Tweet after labeling emotion....38

Table 9: Tweet after Valance Shifter....40

Table 10: Word emotion and sentiment after intensifier....40

Table 11: Total Sentiment and Emotion Score....41

Table 12: Example Annotated Tweet....45

Table 13: Model Answer for Table 12....45

Table 14: Example output of the system for one emotion....45

Table 15: Model answer for Table 14....46

Table 16: Model answer for Table 14....46

Table 17: System output for Tweets with no emotions....46

Table 18: Model answer for Table 17....46

Table 19: Model answer for Table 17....46

LIST OF FIGURES

Page

Figure 1: POS tree example 5

Figure 2: Spanned out POS Tree Example 5

Figure 3: Combinations to extract Bi-Grams (Turney, 2002)....................10

Figure 4: Plutchik's Wheel of Emotions and Grape Fruit Diagram (Plutchik, 2001).......15

Figure 5: Krech's representation of Plutchik's chart 16

Figure 6: SMART Barometer....................22

Figure 7: Plutchik's Wheel of Emotions,....................22

Figure 8: Example JSON Tweet with Tweet Text Highlighted....................25

Figure 9: Higher Level Architecture 29

Figure 10: Step 1 Data Gathering....................30

Figure 11: Step 2 Pre-Processing Part 1 Cleaning Tweets 31

Figure 12: Pre-Processing Part 2 Loading WordLists....................33

Figure 13: Step 3 Sentiment and Emotion Extraction Part 1 Emotion Labeling 36

Figure 14: Step 3 Sentiment and Emotion Extraction Part 2....................39

Figure 15: Results and Evaluation Sheet 43

Figure 16: Results for Emotion Analysis....................47

Figure 17: Results for sentiment analysis....................48

Figure 18: Inter-evaluator agreement for Emotions....................49

Figure 19: Inter-evaluator agreement for sentiment....................49

CHAPTER 1

INTRODUCTION

1.1 Introduction

Opinions have always been a part of our everyday lives. We have been expressing our opinions in different ways for many years; may it be voting in ancient civilizations like Rome or the modern day online reviews. May it be as small scale as gathering signatures for school projects or as big as presidential elections. Our opinions have always made an impact on the way world moves towards the future. Also our opinions have always been an integral part of defining who we are and what we do. In the past, opinions of general public which went on record were less detailed. In today's day and age, we have devised ways to express ourselves more elaborately through microblogs and social networking. Unlike in the past, when most of the opinions were vocal and accessible to very few people and open to interpretation, now we have access to a massive amounts of data with introduction of different forms of social media.

"Sentiment analysis (opinion mining) is the automatic extraction of opinions, feelings, or likes and dislikes from text" (Shellman, Covington, & Zangrilli, 2010). This chapter is an overview of sentiment analysis with its current technical and theoretical challenges, with particular reference to the Plutchik's wheel of emotions (Plutchik, 2001) based NRC Lexicon sentiment analysis (Mohammad & Turney, 2013). We give special attention to the nature of sentiment, its relation to microblogs and social conflicts, and the requisite semantic representations and language understanding techniques.

We will be focusing on the data from one of these social media platforms called Twitter. “Twitter is a fast growing microblog where users tweet about any topic within the 140-character limit and follow others to receive their tweets” (Kwak, Lee, Park, & Moon, 2010). Twitter has 271 million monthly active users with 500 million Tweets sent per day. This large amount of data makes Twitter prime for information gathering. Twitter is well known for its users expressing themselves on major on-goings in the world. What we will be focusing on is their opinions on social conflicts.

Social conflicts are key moments in human history which are responsible for shaping our future. They are the galvanization of different opinions materializing into debates, activities, protests and sometimes wars. These conflicts sometimes result in both anthropic and monetary loss. These conflicts on the other hand tend to bring a wave of Tweets which contain many sentiments and emotions. These sentiments are difficult to gather since the data is unstructured and difficult to assess. This poses a particular problem in the domain of data mining and information gathering.

“An important part of our information-gathering behavior has always been to find out what other people think” (Pang & Lee, 2008). This problem of retrieving what other people think can be tackled by Sentiment Analysis. With sentiment analysis we can measure if what people say is favoring the topic or opposing it. We can find the intensity of the sentiment expressed in these emotions. We can find various emotions ingrained within these opinions and also we can find if these expressed emotions form a pattern. Sentiment analysis can be classified broadly in four categories. First is Machine Learning based sentiment analysis, here a machine is trained with pre-annotated data and that trained machine is then used to classify the rest of the text into categories of neutral, negative or positive. Second, Lexicon based sentiment analysis where they concentrate on certain specific parts of the speech like adjective and adverbs which contain most of the sentiment to classify the overall text. Third is Semantic sentiment analysis, here various graph based approaches are used to identify the sentiment. Lastly, there is statistical approach where a decision is made on whether a piece of text belongs in a given category based on the probability measure.

In CHAPTER 2 we will look at the background on sentiment analysis and what it means and how it works, along with some of its applications. We will then talk about some past works in sentiment analysis in CHAPTER 3.

1.2 Contribution to the Field

What we seek to contribute with this prototype system is a novel methodology to analyze emotions and sentiment in Social Media. We seek to include some low level context identification in the field of emotion mining. Along with NLP aspects like implementing valance shifters and intensifiers in the field of emotion analysis. We also seek to include emotion indicators such as emoticons and other emotion signals to identify emotions in Tweets. We plan to apply sentiment weight to emotions. With sentiment weights we seek to offer improved accuracy. We will also create a bag of words which is unique for the social conflict text on Twitter. Our ultimate goal in this endeavor is to achieve improved accuracy compared to NRC Emotion Analyzer. The existing accuracy of NRC Emotion Analyzer is 54% we seek to improve on the same. All the above mentioned ideas are novel to the field of Emotion Analysis, we seek to make a humble but valued contribution with our work.

CHAPTER 2

BACKGROUND

2.1 Sentiment Analysis

Sentiment Analysis spans a number of different fields, most notably those of (Computational) Linguistics, Natural Language Processing (NLP), Machine Learning (ML) and Semantic Web. Each of the fields brings with it a number of challenges that need to be addressed when working within Sentiment Analysis.

This chapter presents the theoretical background Sentiment Analysis builds upon and what needs to be considered when developing systems in this area. Along with this, a number of concrete applications of the concepts described are mentioned and explained to illustrate the main issues.

"Sentiment Analysis is a linguistic analysis technique where a body of text is examined to characterize the tonality of the document"(Tually, Beer, & Faulkner, 2007). Though the method pre-dates modern technological tools, the use of sentiment analysis has accelerated in recent years with the development of large-scale computational infrastructure that can analyze large unstructured textual data sets.

In a Machine Learning (ML) approach the system a trained on a pre-annotated dataset. Looking at the trained dataset the system classifies the data that follows. Meaning we tell the system how to classify a dataset and then the system classifies the data from here on taking lessons from the past.

In Natural Language Processing (NLP) domain the approach is more in-depth. We understand the structure of the language using the grammar and accordingly guess the sentiment. For example for the sentence “the cat in the hat knows some good tricks”; the Parts-Of-Speech tag tree will be Figure 1 If we span it out I will look like Figure 2Figure 1 Here we can clearly see the tags associated with each word. Here ‘cat’, ‘hat’ and ‘tricks’ are classified as nouns; ‘good’ is an adjective. In sentiment analysis adjectives and adverbs hold the most value as they express the most emotion.

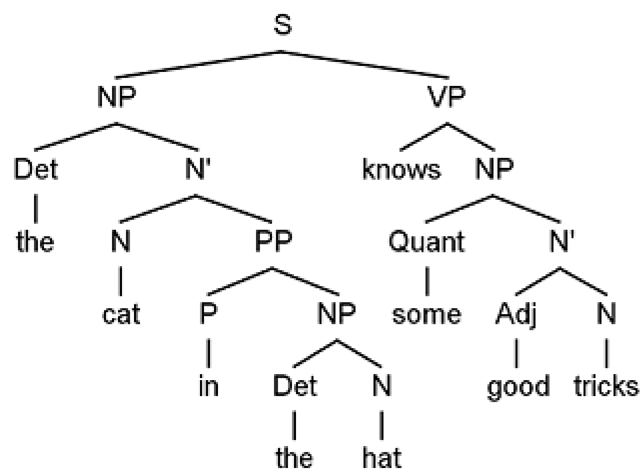

Figure 1: POS tree example

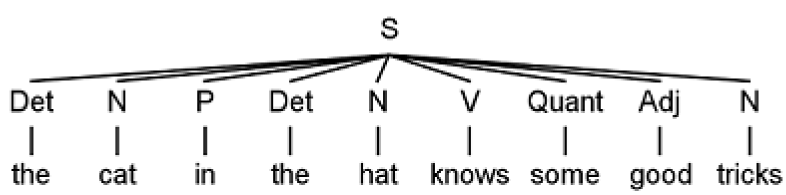

Figure 2: Spanned out POS Tree Example

In a bag-of-words approach we look at only the sentiment words. This may be misleading, for example if we have a sentence such as ‘I saw a documentary on the great depression’. If you use the regular bag-of-words approach we will think that the sentence is negative as the word ‘depression is typically a negative word. If we look at the sentence ‘great depression’ is used as a noun. NLP will look at it from the perspective of Parts-Of-Speech (POS) tags (Toutanova, Klein, Manning, & Singer, 2003). Then ‘great depression’ will be classified as a noun and will not contribute to the sentiment. Hence NLP plays a vital roles in identifying the sentiment and categorizing it correctly.

In Semantic Web approach we divide the given text document into sentences then we map the given sentences onto an ontology, much like in the POS tags, the ontology gives a loose relation of words with each other at a semantic level. The advantage here is, since we map each sentence onto an ontology we can find who the target is for a given sentiment easily. Thus we can identify the actor in a given sentence. For example if a there is a couple of sentences like “Owen is a dog. He is a good dog.” Then with regular bag-of-words approach we will know that the sentiment is positive but to find who the actor is (in this case ‘Owen’) we will require deeper analysis. With ontology this becomes a simpler problem as we can solve it using simple graph traversal. Thus we can find the context of a given sentiment using the Semantic approach.

2.2 Sentiment Analysis Impact on Social Media

Given the growth of user-generated content, sentiment analysis is useful in social media monitoring to automatically summarize the overall feeling or mood of users.

Sentiment Analysis a.k.a. Opinion mining can be useful in several ways. It can help marketers evaluate the success of an ad campaign or new product launch, determine which versions of a product or service are popular and identify which demographics like or dislike particular product features. For example, a review on a website might be broadly positive about a digital camera, but be specifically negative about how heavy it is. Being able to identify this kind of information in a systematic way gives the vendor a much clearer picture of public opinion than surveys or focus groups do, because the data is created by the customer.

There are several challenges in opinion mining. The first is that a word that is considered to be positive in one situation may be considered negative in another situation. Take the word "long" for instance. If a user said a laptop's battery life was long, that would be a positive opinion. If the user said that the laptop's start-up time was long, however, that would be is a negative opinion. These differences mean that an opinion system trained to gather opinions on one type of domain may not perform very well on another.

This brings us to Sentiment analysis in Political setting. In the political setting, people are more vocal about the negative part of any topic compared to the positive (Bakshy, Hofman, Mason, & Watts, 2011). This poses a problem as the data we will have will always be largely polarized. This situation demands a better insight into the polarity. Meaning presenting just Positive Negative or Neutral is not sufficient. We need to find a more diverse opinion to match to the given data. Meaning we need to categorize the data on more than the basic three categories.

2.3 Social Conflicts

Social conflicts (Kriesberg, 1973) are the focal points of regional unrests of differing scales. These events may be caused by difference of opinion between two or more classes of people. These classes may be formed due to religious, political, diplomatic or other points of views. “Best way of resolving a conflict is by avoiding it” (Lewin, 1945). This can be achieved by sentiment analysis. If we analyze social media for an ongoing topic with various sentiments and their scales we may come across some patterns in the flux in sentiment.

Social conflicts on the other hand pose a different kind of problem for the data from platforms such as Twitter. In the research by Amy Mitchell et al. (Mitchell & Hitlin, 2013) we found that Twitter data is rather liberal than the actual masses. They compared a person to person survey result with twitter opinions and found that at times the data can be unreliable. Thus most of the sentiments on Twitter would be extreme. Thus a normal Negative, Positive and Neutral classification of data would be insufficient. We need to dig deeper and find more detailed emotions. We need to characterize the emotions on more than three dimensions.

CHAPTER 3

RELATED WORK

3.1 Sentiment Analysis

There is a well-known book by Bing Liu on Sentiment Analysis and Opinion Mining which describes how a sentiment analysis task works and how to go about gathering opinions (Liu, 2012). The book represents some specific problems in gathering opinions such as topical relevance (which the book states is still-open problem), gathering the overall sentiment of a sentence or document and finally presenting the gathered sentiment as a whole. These problems have been there for a long time in the field of sentiment analysis. We will look at the problem of gathering sentiment in greater detail.

Bo Pang et al. (Pang, Lee, & Vaithyanathan, 2002) claimed that standard machine learning techniques outperform human-produced baseline with their experiments with Naïve Bayes, maximum entropy and support vector techniques. Bo pang claimed that “there are certain words people tend to use to express strong sentiments, so that it might suffice to simply produce a list of such words by introspection and rely on them alone to classify the texts.” (Pang et al., 2002). Upon evaluation it was found that support vector machine outperformed both Naïve Bayes and maximum entropy by giving and accuracy of 82.9% as compared 81% (Naïve Bayes) and 80.4% (maximum entropy).

This claim was bolstered by Das et al (Das & Chen, 2007) who describes comparison of many of these approaches in presence of ambiguity and describes methods to minimize the ambiguity. They used techniques such as Naive Classifier, Vector Distance Classifier, Discriminant-Based Classifier, Adjective-Adverb Phrase Classifier, etc. Naive Classifier is “based on a word count of positive and negative connotation words. It is the simplest and most intuitive of the classifiers” (Das & Chen, 2007). In Vector Distance Classifier “If there are D words in the lexicon, and each word is assigned a dimension in vector space, then the lexicon represents a D-dimensional unit hypercube. Every message may be thought of as a word vector ($m \in R^D$) in this space” (Das & Chen, 2007). Discriminant-Based Classifier is Naive Classifier with weights for words, since naïve classifier treats all the words the same an approach where the weights were assigned to words according to their sentiment intensity would be more effective. Adjective-Adverb Phrase is “based on the assumption that adjectives and adverbs emphasize sentiment and require greater weight in the classification process” (Das & Chen, 2007). They ran the data through these various systems and compared the results. For low, medium and high ambiguity data with sample set size 374 the Discriminant-Based Classifier performed the best with and accuracy of 57.24%, 58.13% and 54.89% respectively. But when the Sample set size was increased to 913 Adjective-Adverb Phrase Classifier was the most effective one with and average accuracy of 62% in low, medium and high ambiguity sets, in fact the accuracy kept increasing as the ambiguity of the data increased. Thus it is safe to say that the adjective and adverbs

	First Word	Second Word	Third Word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

Figure 3: Combinations to extract Bi-Grams (Turney, 2002)

contain the most valuable data, also a system which is weighted yields better results in general.

Peter Turney (Turney, 2002) explains in his research the importance of bigrams in sentiment analysis. Turney extracted bigrams with a combination as described below Figure 3. The combination was well received, by the experiments they conducted the accuracy was on an average 74% when the bigrams were introduced to the equation. Another important find was by Polanyi et al (Polanyi & Zaenen, 2006). They talk about the use of contextual valence shifters. From all the previous research we discussed main emphasis was on how we can use the adjectives and adverbs we miss out on the importance of verbs in sentiment analysis. With emphasis away from verbs a sentence such as “He is not a good boy.” Will be classified as ‘positive’ because of the presence of the word ‘good’. Thus what Polanyi et al. (Polanyi & Zaenen, 2006) describe as valence shifters (for example no, not, none, nobody etc.) Which change the meaning of the sentence, are vital for extracting sentiments and opinions. Polanyi et al. (Polanyi & Zaenen, 2006) say that when encountered with a valence shifter the sentiment of the upcoming sentiment word (like in the above example ‘good’) will be reversed. This is accurate in the everyday grammar and its incorporation in sentiment analysis provides valuable insight into sentiment/opinion extraction.

Taboda et al. (Taboada, Brooke, Tofiloski, Voll, & Stede, 2011) on the other hand talk about the importance of intensifiers (for example very, really, most etc.) which also affect the sentiment score of a given sentence. Shellman et al. (Shellman et al., 2010) describe the impact of the same as “good is +1, then maybe very good is about +1.6”. This is also an important find in this field as intensifiers are very important in interpreting what the nature of the sentiment is in a sentence.

Twitter on the other hand poses a different kind of problem for sentiment analysis. The microblogging platform allows a Tweet of the size 140 characters. This means the users use a lot of abbreviations to express emotions they use emoticons or smileys. Apoorv Agarwal et al. (Agarwal, Xie, Vovsha, Rambow, & Passonneau, 2011) talk about a method that uses unigram model where they use smileys to factor out positive and negative sentiments. According to their research unigram model is the most popular of the models for sentiment analysis of Twitter data. They also describe a tree based approach to go about sentiment analysis. As the unigram approach does not take into consideration the POS (Parts of Speech) for analysis; they introduce a tree for stop words and POS words to synthesize tweets. This approach yielded significantly better results as per their research. Their research also tells that the unigram model provides a good baseline and is effective in many cases.

This brings us to SENTIWORDNET 3.0 (Baccianella, Esuli, & Sebastiani, 2010) which was developed by Stefano Baccianella et al. an enhanced lexical resource explicitly devised for supporting sentiment classification and opinion mining applications. It is an improved version of SENTIWORDNET 1.0 (Esuli & Sebastiani, 2006), a lexical resource publicly available for research purposes. SENTIWORDNET takes advantage of the synsets (Synonym sets) of original WORDNET (Miller, 1995) along with some crowd sourcing to provide a comprehensive wordlist of more than hundred thousand words. Each word is classified as negative or positive on a scale of -1 to +1. The sentiment score are the weighted average of the ranks of the synsets from WORDNET.

Which means if the given word has synonyms which have a higher rank than itself, then the sentiment score of the current word will be closer to zero. If the ranks of the synonyms of the current word are lower than itself then the sentiment score is higher and closer to 1 or -1 depending on the positive or negative leaning of that word. Apart from that SENTIWORDNET also uses POS tags to provide a better description of the sentiment of a word, for example ‘I am feeling blue’ and ‘I am wearing blue’ here the word ‘blue’ is used in two separate ways. Once as a noun (sentence 2) and other as an adjective (sentence 1). On both occasions the sentiment the word ‘blue’ contributes to the sentence is different. When used as a noun ‘blue’ does not contribute any sentiment, but when used as an adjective the word represents ‘sadness’ which is a negative sentiment. SENTIWORDNET claims to identify such nuances in the sentence with help of POS tags. SENTIWORDNET 3.0 has 19.48% relative improvement for the ranking by positivity and a 21.96% improvement for the ranking by negativity as compared to SENTIWORDNET1.0.

Tetsuya Nasukawa et al (Nasukawa & Yi, 2003) describe a lexicon based approach to identify sentiments; they introduce an approach where the system gathers opinion of the subject rather than the whole document. They also built a prototype which worked at 75-95% precision depending on the data. They applied this system on a corpus containing different products and reviews about them. They compared the results with human raters and results were convincing, there seemed to be some scope for improvement here, never the less the idea was interesting.

Godbole at al. (Godbole, Srinivasaiah, & Skiena, 2007) describe a statistical approach where sentiment index relay critically on tracking the reference frequencies of adjectives with positive and negative connotations. They present a method for expanding small candidate seed lists of positive and negative words into full sentiment lexicons using path-based analysis of synonym and antonym sets in WordNet (Miller, 1995). They use sentiment-alternation hop counts to determine the polarity strength of candidate terms and eliminate the ambiguous terms.

In the research by Albert Weichselbraun et al. (Weichselbraun, Gindl, & Scharl, 2013), they used ontologies to build context aware system to analyze corpora and extract sentiment of the same. This was built upon technologies such as (www.weblyzard.com). The research tries to exploit the metadata about the identified context to add to the mechanism of sentiment analysis thus disambiguating sentiment terms. "It *paves the way to incorporation of semantic databases into sentiment analysis"* as described by the author. This research was widely based on the idea of Media Watch by Alexander Hubmann-Haidvogel et al. (Hubmann-Haidvogel, Scharl, & Weichselbraun, 2009) This is a web intelligent system which automatically gathers opinions form large text corpora extracted from news and social media.

Hassan Saif et al. (Saif, He, Fernandez, & Alani, 2014) discuss an important point of context in sentiment analysis. He describes an example "I have studied the great depression, it is interesting". With traditional bag of words approach we will label the text as negative as it contains the word 'depression' but that will be inaccurate as in this case 'depression' does not participate in the sentiment of the text. Here 'great depression' is a noun and it does not contribute to the overall sentiment of the text. With that in mind Hassan Saif et al. (Saif et al., 2014) emphasize only considering the word which offer sentiment and ignore the others to maintain the accuracy level of the system. They offer an elegant solution to this by providing POS (Parts of Speech) tags to the text. This way we can easily filter the useful text from the less useful. With POS tags we can separate out the text which are either Adjective, Adverb or Verb; which are sentiment rich part of any text (Turney, 2002) so that the accuracy is maintained in case we come across a potential sentiment term which is used in a non-sentiment fashion.

Xai Hu et al. (Hu, Tang, Gao, & Liu, 2013) emphasize the importance of emoticons and other non-textual aspects, ‘Emotional Signals’ of a Tweet in Opinion Mining. They used Emotional Signals to classify text. What they claim was the sentiment of the statement/ Tweet will be consistent with the Emotional Signals. They conducted several experiments to verify the same. They used Emotional Signal alongside various unsupervised classification methods to measure the gain in accuracy. They had a constant gain with their algorithm and the hypothesis was proved.

3.2 Emotion Analysis

There has been a debate over emotion detection and sentiment categorization. Ekman (Ekman, 1992) proposed that the most basic emotions were joy, sadness, anger, fear and

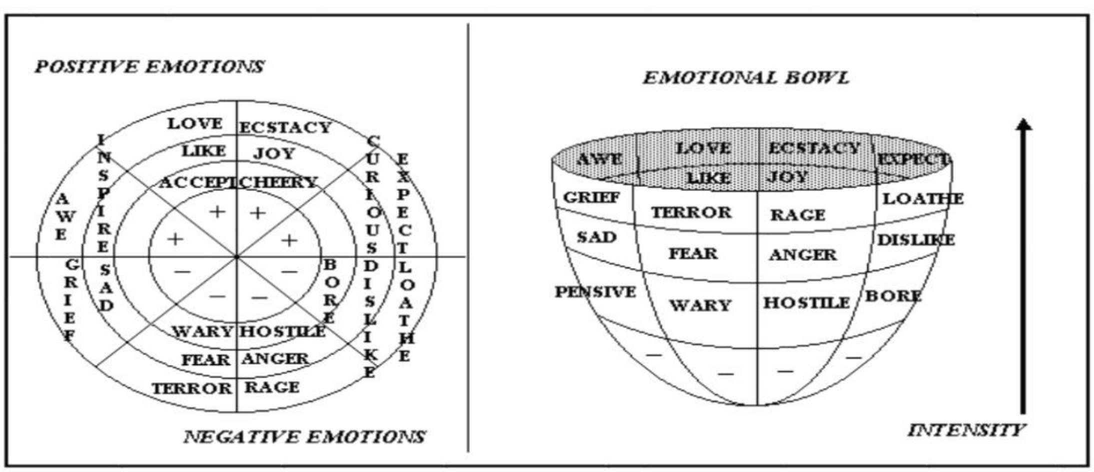

Figure 4: Plutchik’s Wheel of Emotions and Grape Fruit Diagram (Plutchik, 2001)

disgust. Plutchik (Plutchik, 2001) argued that there were eight; Ekman's six, surprise and anticipation. Plutchik's categorization was more refined and was found to be more detailed with better scope for merging the emotions and generating more combinations, expressing more holistic picture of the emotions expressed. Here is the Plutchik's wheel of emotions Figure 4: Plutchik's Wheel of Emotions and Grape Fruit Diagram which describes all the emotions and the combinations possible. Automatic emotions detection system such as NRC Emotion Lexicon (Mohammad & Turney, 2013) and WordNet Affect (Strapparava & Valitutti, 2004) use Plutchik's wheel of emotions to express emotion categories. A better representation is displayed in the following image Figure 5. Here we can see how the activation level changes the intensity of the emotions. For example if we reduce intensity of 'grief' then it becomes 'sadness', if we decrease the intensity further emotion becomes 'pensiveness'

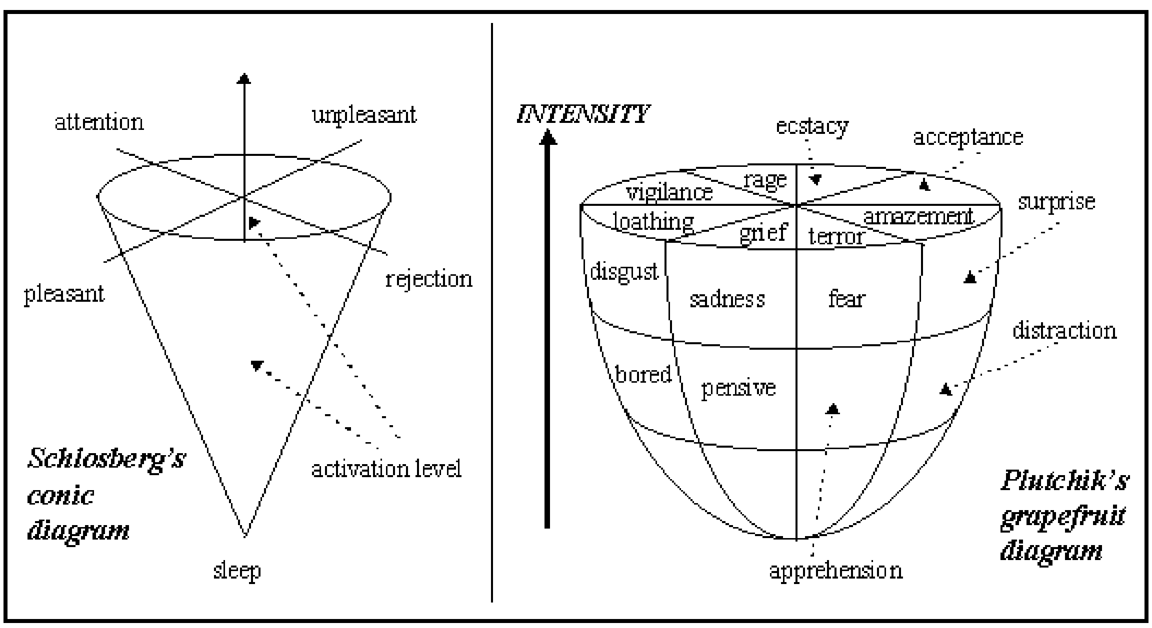

Figure 5: Krech's representation of Plutchik's chart (Krech, Crutchfield, & Livson, 1974)

Mohammad et al (Mohammad, Zhu, & Martin, 2013) collected over a million tweets related to United States Presidential Elections. They collected Tweets with keywords

#Barak, #campaign2012, #elections, etc. Then cleaned the tweets that were retweets and cited tweets to bring that number down to 170,000. Later for ease of experiment they chose 2000 random Tweets each by a different Twitter user and forwarded them to Amazon's Mechanical Turk. They had two questions for the crowd workers in AMTurk. One consisting for how many emotions are present in a given Tweet. The second was to find out if the Tweet was relevant to the topic of US Politics. Each Tweet was annotated by at least three annotators. Then Tweets containing single emotion were forwarded to five more crowd workers for further analysis. Where the first set of questions found the general emotion the second set of questions looked for more refined emotion. In the experiment 87.98% Tweets were identified as having an emotional attitude. Inter-annotator agreement was that at least two annotators agreed with each other on the emotion of the Tweet. Then using Porter's stemmer(Porter, 1980) all stop words were removed from the dataset. From the remaining words all the relevant terms were extracted and the data was used as a lexicon with appropriate annotations for emotions. The lexicon contained both unigrams and bigrams. The system obtained an accuracy of 56.84% which was significantly higher than the majority baseline. It should be noted that the highest scores in the SemEval 2013 task of detecting sentiment analysis of tweets was around 69% (Mohammad, Kiritchenko, & Zhu, 2013) even though it involved only three classes (positive, negative and neutral). Thus it is not surprising that for an 8-way classification task, the performance is somewhat lower.

3.3 Negation

Negation is another important aspect of sentiment and emotion analysis. There are many different theories of how to handle negation. For sentiment analysis Polanyi et al.

expressed that with negation the sign of the sentiment weight shifts. That is, if sentiment weight of a word was say ‘3’, then with negation it becomes ‘-3’. Thus, stating that something like ‘not good’ is equivalent to ‘bad’

Khan et al. {Khan, 2011 #47} on the other hand, discusses the possibility of reduction in intensity of the sentiment weight of a word. So, if the sentiment weight of the word was say ‘3’ before negation it will become something like ‘-2’ after negation. This means, something like ‘not good’ is less negative than ‘bad’.

The effect on the emotions however remains the same. As shown in Table 2, with negation the emotion just become the opposite emotion. This is a property of the Plutchik’s chart.

For this experiment, we will make an assumption based on the research of Polanyi et al. This decision is made because, both the cases separately cover different scenarios, but none of them cover all the scenarios of a negation. Thus it is a no win situation. But with Polanyi et al., even though we assign higher weight to a word, we can compensate the same when we consolidate the weights at the end of the process. Thus, this approach promises better information gain.

3.4 Summary of Related Work

Above we have discussed many sentiment analysis and some emotion classification techniques. Each technique used a different way to solve the problem of sentiment analysis with varying accuracy and effectiveness for the data at hand. Table below Table 1 summarizes the papers and methods discussed above.

Table 1: Summary of Related Word

Author	Data	Technique	Method	Classes for classification	Accu-racy
Pang, Lee et al. 2002	Plain Text	Support Vector Machine	Machine Learning	Negative, Positive and Neutral	83%
Das and Chen 2007	Ambiguous Text	Adjective-Adverb Phrase Classifier	Machine Learning	Negative, Positive and Neutral	62%
Turney 2002	Partly Ambiguous Text	Bi grams and Unigrams	Bag of Words	Negative, Positive and Neutral	74%
Polanyi and Zaenen 2006	N/A	Contextual Valence Shifters	Lexical Analysis	Negative, Positive and Neutral	N/A
Taboada, Brooke et al. 2011	N/A	Intensifiers	Lexical Analysis	Negative, Positive and Neutral	N/A
Agarwal, Xie et al. 2011	Twitter Data	Unigram with POS tags	Bag of Words + Lexical Analysis	Negative, Positive and Neutral	N/A
Baccianella, Esuli et al. 2010	Various Sources	WordNet Synsets	Bag of Words + Lexical Analysis	Negative, Positive and Neutral	71%
Nasukawa and Yi 2003	Various Sources	Contextual Sentiment Analysis	Lexical Analysis	Negative, Positive and Neutral	75%
Godbole, Srinivasaiah et al. 2007	News Articles	Probability Based Sentiment Analysis	Statistical	Negative, Positive and Neutral	69%
Weichselbraun, Gindl et al. 2013	Various Sources	Graph Traversal	Semantic Sentiment Analysis	Negative, Positive and Neutral	N/A
Mohammad, Kiritchenko et al. 2013	Twitter Data	Bi grams and Unigrams	Bag of Words	Negative, Positive and Neutral	69%
Mohammad, Zhu et al. 2013	Twitter Data	Plutchik's Chart Based Emotion Analysis	Bag of Words	Joy, Sadness, Anger, Fear, Disgust, Anticipation and Surprise	55%

We observe many things in the research we studied. More prominently bi-grams, valence shifters, intensifiers and POS tags play a big role in sentiment analysis. As the number of features to be extracted increase the accuracy of the system may decrease. Each technique is effective for its own domain of data never the less a good combination of these technique may prove effective for the problem of extracting emotions from Tweets for Social Conflicts.

CHAPTER 4

SMART SENTIMENT AND EMOTION ANALYSIS

4.1 Problem Statement

The motivation behind our system is the need to offer a flexible platform to analyze emotions in Social Media. Also, to offer assistance in various fields of data analysis. For example, in the field of finding ‘Identifying Patterns of Social Conflicts’, ‘Depression Assessment on Twitter for potential suicide victims’ or in the field of ‘Bully Detection on Twitter’. In all these areas, the traditional three way classification of text into categories such as ‘negative’, ‘positive’ and ‘neutral’ was proving insufficient. Thus we took the initiative to classify the text on the dimensions of Plutchik’s wheel of emotion (Plutchik, 2001) *Figure 7*. We offer to provide the classification platform such that it can adopt to the eight classes of Plutchik’s wheel of emotions.

The major motivation behind this endeavor was to fulfil the vision of the SMART Barometer. SMART stands for Social Media Analysis in Real Time. A system which can display real-time analysis of data about Social Media, Traditional Media, Social conflict activity (or Protest activity) and Government activity Figure 6. This is a revolutionary idea in the field of social conflict analysis and prediction as it offers a novel platform to compare various aspects of a Social Conflict. This work will contribute to the Social Media aspect of the SMART Barometer.

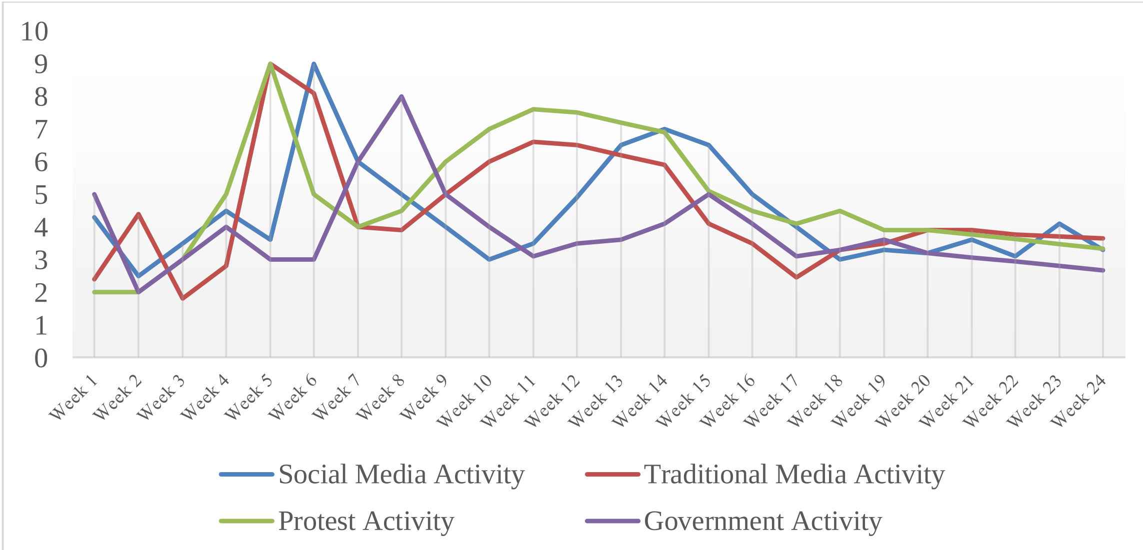

Figure 6: SMART Barometer

In our approach, we classify the Tweet emotion on a scale of 0-5. Zero being the absence of emotion and 5 being the presence of the most intense emotion. Upon emotion labeling if a Tweet emotion score falls on or within the range of 1-2 then the emotions is in the outer most ring of the Plutchik's wheel which represents serenity, acceptance, apprehension, distraction, pensiveness, boredom, annoyance and interest. These are the weakest of the emotions offered by this emotion representation system. If the emotion score is on or

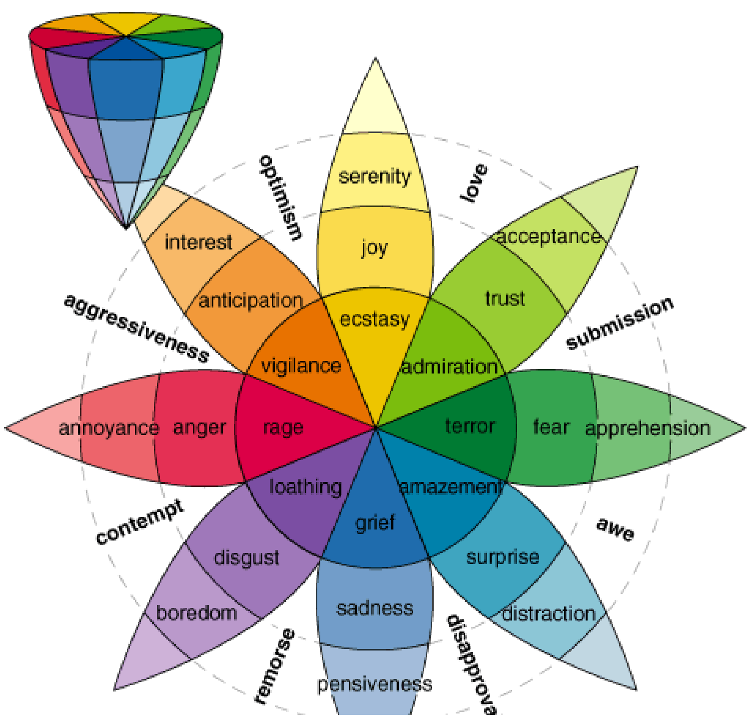

Figure 7: Plutchik's Wheel of Emotions,

Image courtesy (www.fractal.org)

within the range of 3-4, then the emotion fall on the middle ring of the Plutchik's wheel which represents joy, trust, fear, surprise, sadness, disgust, anger and anticipation. For all the emotion scores of 5 and above, the emotions fall on the innermost ring which represents the most intense of emotions such as ecstasy, admiration, terror, amazement, grief, loathing, anger and vigilance.

According to Table 2, we can see how the emotions have relations between them.

Table 2: Opposite Emotions

Basic emotion	**Basic opposite**
Joy	Sadness
Trust	Disgust
Fear	Anger
Surprise	Anticipation

We will take the example of the middle ring of Plutchik's chart from here on for ease of explanation. According to Table 2 we can see that the emotions opposite each other in *Figure 7*, like joy and sadness are complimentary emotions. Thus, if we are to find the opposite of a given emotion in Plutchik's chart we need to look at the opposite end of the emotion in perspective. We can broaden the spectrum of emotions by combining the emotions for example in Table 3.

Table 3: Combination of Emotions

Combined Emotion	**Constituent Emotions**	**Opposite Emotion**
Optimism	Anticipation + Joy	*Disapproval*
Love	Joy + Trust	*Remorse*
Submission	Trust + Fear	*Contempt*
Awe	Fear + Surprise	*Aggression*
Disapproval	Surprise + Sadness	*Optimism*
Remorse	Sadness + Disgust	*Love*
Contempt	Disgust + Anger	*Submission*
Aggressiveness	Anger + Anticipation	*Awe*

We can see that the emotions anticipation and joy together represents optimism whose opposite is disapproval and so on. This gives us a possibility of 32 classes to classify the emotions. But the 32 classes may be few too many for analysis and thus we stick with the middle ring of Plutchik's chart for the sake of ease of experimentation. Thus, we will annotate our wordlist on the basis of joy, trust, fear, surprise, sadness, disgust, anger and anticipation. The reason behind this is that for annotating a wordlist the annotator should not be confused and should be comfortable with the system. Also 32 types of emotions with experiments have proven to be too overwhelming for an annotator (Mohammad, Kiritchenko, et al., 2013).

We also observed that the NRC word dictionary was not sufficient for the Tweets we encountered. Thus we took the initiative to create our own word dictionary which would help us classify the Tweets into their respective emotions. The following sections section 4.2 and 4.3 will talk about how we created this word dictionary.

4.2 Data Gathering and Pre-Processing for Word List

In this section, we will talk about data gathering and pre- processing for the purpose of creating a word list. We gathered over three and a half million Tweets about conflicts in Urkain-Russia and Israel-Gaza; the keywords we used were 'ukraine', 'UkraineCrisis', 'euromaidan', 'IsraelGaza', 'Ferguson', 'Hamas'and 'Palastine'. The Tweets come in JSON format Figure 8, thus it was essential that we added the Tweets to some sort of data management platform. Thus, all the Tweets were then added to MongoDB (Chodorow, 2013) using 'mongoimport' command. MongoDB is a NoSQL (Cattell, 2011) database. It is designed to be scalable and easily adaptable to basic structured text notations like JSON or XML (Nurseitov, Paulson, Reynolds, & Izurieta, 2009). We named the table of Tweets UkraineIsrael_v1 (in NoSQL terms a table is called a 'collection').

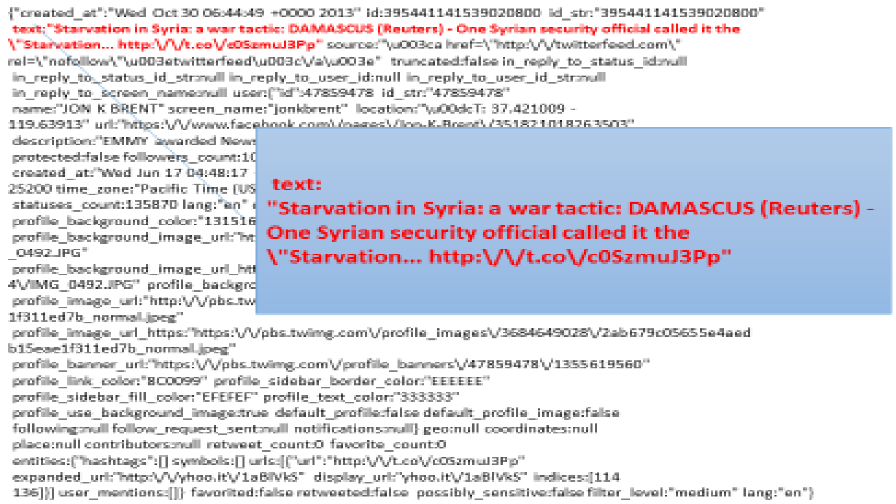

Figure 8: Example JSON Tweet with Tweet Text Highlighted

We indexed the MongoDB collection on the section 'text' as highlighted in Figure 8. Using a Java (Gosling, 2000) program we accessed the Tweets one by one with the help of MongDB drivers. We read only English Tweets. We then cleaned the Tweets. We removed the '#' sign from the HashTags. Removed the usernames. We also removed repeated letters, like 'hellooo' was changed to 'hello'. Some of the Tweets were retweets which were basically repeats. While developing this system we did not deal with real-time data, even though the end product would work easily for real-time data. Thus repeating tweets were counterproductive for the effort. So by matching the Tweet-id with the retweet-ids of other Tweets we removed repeats. After cleaning we were left with a bit more than one million Tweets.

4.3 SMART Word List

We matched words from both NRC word dictionary for emotions (Mohammad & Turney, 2013) and SentiWordNet (Esuli & Sebastiani, 2006) with all the words present in the Tweets we had. We realized that the word lists may not be sufficient as

SentiWordMet was not custom made for Twitter and NRC emotion lexicon contained 35,455 words which was a good number but had some scope for improvement as it did not contain emoticons, profanity words and emotion signals. This led to the decision of creating our own word list.

To create a word list we first narrowed down only the useful words in terms of sentiment by using porter stemmer (Porter, 1980) on the Tweets to remove all the stop words. We then used the Stanford Parser (De Marneffe, MacCartney, & Manning, 2006) to put POS (Parts of Speech) tags on the text. Then we removed all the proper nouns from the set. We gathered all the remaining words and sored them in descending order of their number occurrences. We chose first 1,500 words which had a minimum occurrence of 56 this included emoticons and other emotion indicators (like combination of punctuation marks such as '!!' for surprise and '…' for pensiveness). We annotated these words for emotions manually by using two annotators. For each word, we took only that emotion on which both the annotator agreed. We needed a sentiment score on a scale of +5 to -5 for each word as well. It would have been a lot of effort to provide a sentiment score to each word by hand thus we used a pre-existing word dictionary offered by Alex at el. (Davies & Ghahramani, 2011). Using this, we provided sentiment score to 1,004 words. For the remaining words, we asked both the annotators to provide sentiment score. To provide the sentiment score/weight, we asked the annotators to think of the superlative of the given word and think of that word as the most intense in that category and then annotate the current word accordingly. We took the average of the sentiment score provided by both the annotators and rounded the value to the smallest nearest integer. After annotation was complete, we removed all the words with sentiment score zero and which had no emotions listed, there were 72 such words.

We then extracted bi-grams as described by Turney (Turney, 2002). We concentrated on the combination of 'verb or adverb followed by adjective not followed by noun', 'adverb followed by adjective not followed by a noun', 'adjective followed by adjective not followed by a noun' and 'adverb followed by verb not followed by a noun'. These combinations were considered because they offer the most sentiment rich bigrams. We then sorted the bigrams according to their number of occurrences and took the first 53 bigrams which had the least number of occurrence as 28. We annotated these with both sentiment score and emotions same way as we did for other words. We removed all the bi-grams which had sentiment score of zero and had no emotion labels, 13 bi-grams were dropped at this stage. We combined both annotated unigrams and bi-grams to make one comprehensive dictionary which was of 1,468 words specifically targeted for Social Conflict Tweets. This word list contained emoticons and other emotion indicators such as '!!','…' and '??' etc.

4.4 Sentiment and Emotion consolidation

We will discuss how the wordlist was used to extract emotions and sentiments from Tweets in CHAPTER 5. We will now discuss how we can consolidate emotions present in a Tweet. Let's look at the following example "#Gaza is a utopia for the people from world over, it is a great example of affection and well-being'. In this Tweet we have the following sentiment words 'utopia' (sentiment sore +3), 'great' (sentiment sore +4) and 'well-being' (sentiment sore +3). The total sentiment of the Tweet will be '3+4+3=10'. To be able to put the sentiment into perspective we will require the score to be on the scale of 5 to -5. In order to do that, we found sentiment of all tweets and found the 90th percentile of the negative and positive extreme of all the tweets. I.e. if there were ten positive tweets then we can sort them in ascending order of their sentiment score and pick the second highest number from the list. This will give us the 90th percentile. We then extrapolated this number on the scale of +5 to -5. I.e. if 90th percentile was say 10 then anything between 0-1 becomes 0, anything between 2-3 becomes 1, anything between 4-5 becomes 2, 6-7 becomes 3, 8-9 becomes 4 and 10 becomes 5. We did the same for the negative sentiment and got an output for the same. The 90th percentile value keeps changing as new Tweets come in thus constantly changing the extrapolation.

For emotions if according to the word list a word has emotion as sadness and anger and its sentiment score is -3 the emotion score will become 'joy=0, trust=0, fear=0, surprise=0. Sadness=3, disgust=0, anger=3 and anticipation=0'. This represents all the emotions and their intensities. We then add together the emotion score to get the overall emotion of the tweet. For example if there were two sentiment words in the tweet say word 1 'joy=0, trust=0, fear=0, surprise=0. Sadness=3, disgust=0, anger=3 and anticipation=0' and word 2 'joy=0, trust=0, fear=0, surprise=0. Sadness=2, disgust=2, anger=2 and anticipation=0'. Then overall emotion will be 'joy=0, trust=0, fear=0, surprise=0. Sadness=5, disgust=2, anger=5 and anticipation=0'. We will extrapolate the emotion score for each emotion the same way we did for the sentiment, using 90th percentile for each emotion over all the Tweets. The score will be extrapolated to the scale of 0-5. Where 0 is no emotions present and 5 is most intense form of emotion.

CHAPTER 5

ARCHITECTUTE

5.1 High Level Architecture

In this chapter we will look into the inner working of the prototype we have proposed. The system we created is a three step process. Let us look at the high level architecture of the same in Figure 9.

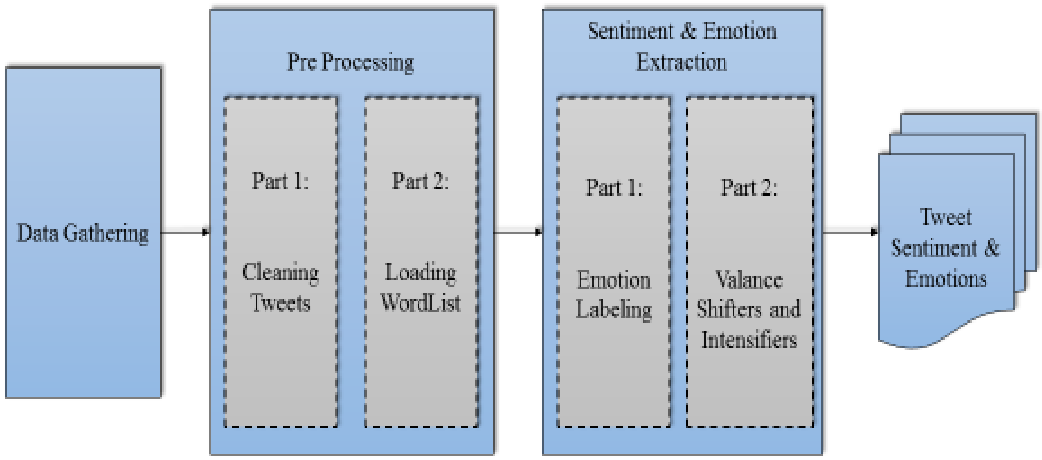

Figure 9: Higher Level Architecture

As we can see in the diagram we have three primary components of the architecture. Data Gathering, Pre-Processing and Sentiment & Emotion Extraction. Data Gathering as the name suggests deals with gathering the data required for analysis from Twitter and adding that data to MongoDB. Pre Processing deals with part 1, cleaning the data gathered in step 1 and part 2 loading the word lists. Then comes the step where we perform Sentiment and Emotion Extraction. This step also has two parts part 1, Emotion

Labeling and part 2 applying valance shifters and intensifiers. At step 3 we get the final output which can be added back to MongoDB. Once added to MongoDB since the format will revert back to JSON we can use that data in any way we want.

5.2 Step 1: Data Gathering

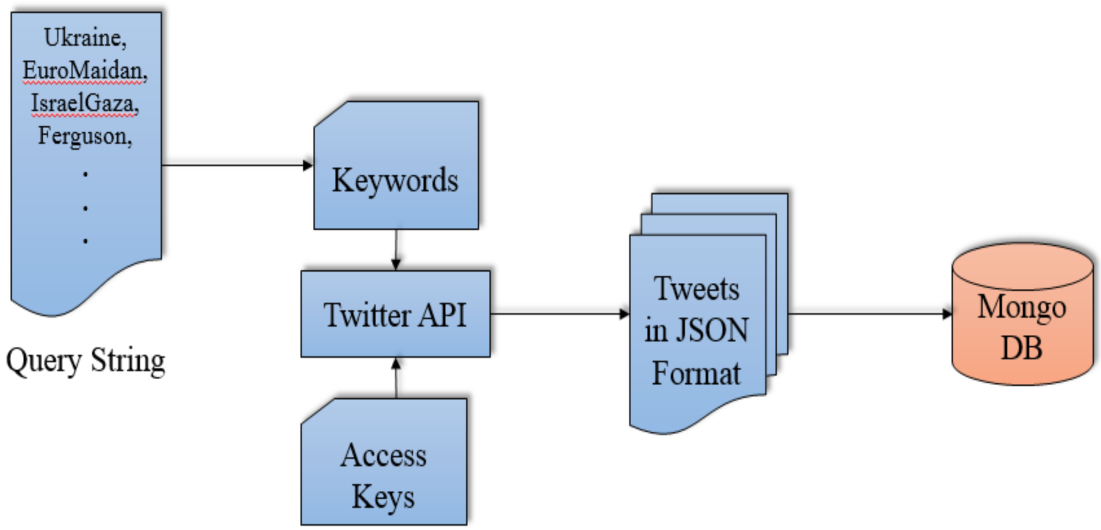

Figure 10: Step 1 Data Gathering

For data gathering we used a common Python (Sanner, 1999) script which used a Twitter driver which enabled us to tap into live Tweeter feed. The driver offers us access to Twitter API which houses several methods to access Tweets. The access can user specific or according to keywords.

In order to establish access we require to create Twitter Developer's account. Each Developer account may have one or more applications. This account offers us Asses keys which are Customer key and Customer Access Token. These are unique to the application we created in Twitter Developer's account. We use these Access Keys in the Python script to gather the Tweets. We then provide query string which in this case is Ukraine, EuroMaidan, IsraelGaza, Ferguson, etc. These were all the HashTags which were trending at the time of data gathering related to the Israel-Gaza and Ukraing-Russia crisis. We get Tweets in JSON format as shown in Figure 8 using the Python script we created, we added these Tweets to MongoDB using the command 'mongoimport' after creating a collection named 'UkrainIsrael_v2'. We then applied index on the section named as 'text' like in section 4.2 which contained all the Tweet text. Once added to MongoDb, we then move to the part of pre-processing. The data we gathered in this section was different from the data we gathered for creating the word list.

5.3 Step 2: Per-Processing Part 1 Cleaning Tweets

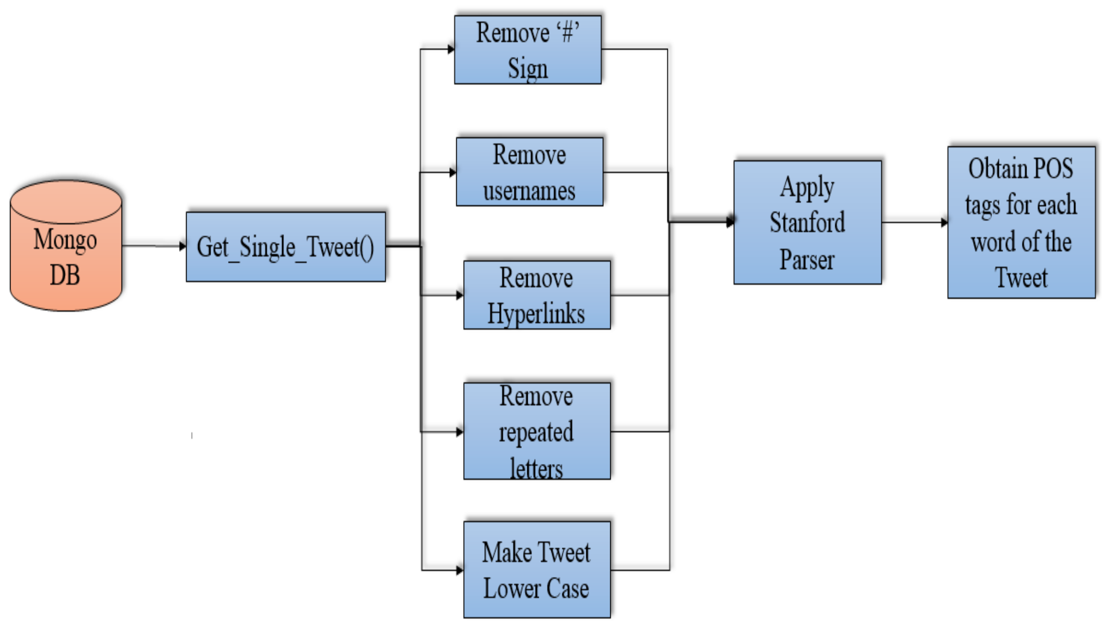

Figure 11: Step 2 Pre-Processing Part 1 Cleaning Tweets

In Step 1, we Gathered the data from Twitter and added that data to a collection in MongoDB{Hows, 2013 #46}. In this step, as shown in Figure 11 we use that data for extraction and cleaning. We access the data using MongoDB driver which is available for

Java. We then access one Tweet at a time and proceed with cleaning the Tweet. First we remove the ‘#’ sign from the Tweets. Typically this sign is used to signify a HashTag on Twitter. Sometimes HashTags are sentiment words (Davidov, Tsur, & Rappoport, 2010) thus they are important for our objective. But with the pound sign it will be difficult for us to match those word with our wordlist thus it is vital to remove it. We then proceed to removing usernames from the Tweets. We do this by using a Regex (Habibi, 2004) which are regular expressions. We look for a regular expression which starts with the symbol ‘@’ and ends with a space. We then replace all such matched regular expressions with blank text. We remove usernames from the Tweets because they do not contribute to the sentiment of the Tweet and removing them helps making the process faster. We then remove the hyperlinks present in the Tweets. We look for the regular expression starting with ‘http://’ ending with a space. We match this regular expression and replace it with blank space. Again this is done because hyperlinks do not contribute to sentiment or emotion of the Tweet and removing them helps making the process faster. We then remove the repeated letter from the Tweets. For example we change “ddooooonnnneeee” to “done”. We use ‘replaceAll ("(.)\\1{2,}", "$1")’ on the string to do this. What the regex does is if a letter appears more than two times in a word then those set of letters are replaced by a single occurrence of the same letter. This is vital as on Social Media many times users type this way. This obscures the words for analysis. Thus it is important that we fix this occurrence. We than turn the sentence into lowercase. The wordlists we are using are all in lowercase thus changing the sentence to lowercase helps. We then apply the Stanford Parser {De Marneffe, 2006 #40} on the text. This provides us with POS tags for all the Tweets. We convert sentence into a two dimensional array list. Where the sentence is an array list with each word inside as an array list by themselves. The inner array lists have the size of 11. With word itself being the first element of the array list. Second element is the sentiment score. Next eight elements are the emotions starting from joy till anger. Last element is the POS tag we obtained from the Stanford Parser {De Marneffe, 2006 #40}. For now, we apply all the entries representing emotion or sentiment as zero.

Let us look at the algorithm for cleaning again.

1. Get the Tweet from MongoDB.
2. Remove '#' sign.
3. Remove usernames (any word that starts with '@').
4. Remove hyperlinks.
5. Remove repeated letters.
6. Make Tweet lower case.
7. Apply the Stanford Parser.
8. Obtain POS tags for each word.

5.4 Step 2: Pre-Processing Part 2 Loading Word Lists

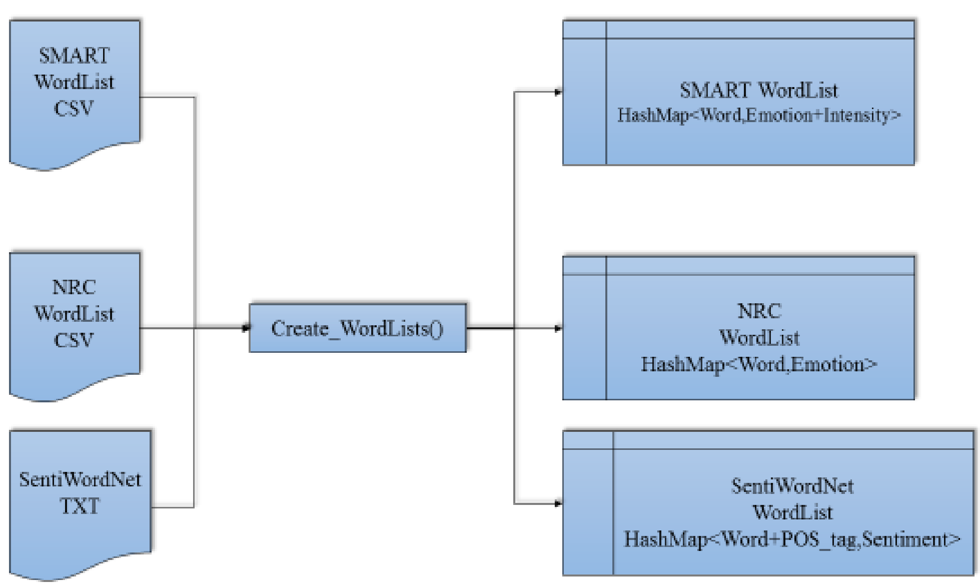

Figure 12: Pre-Processing Part 2 Loading WordLists

We have three word lists which we require to load. First is SMART word list which we have created by ourselves. It contains 1,468 words then there is the NRC word list which contains 35,000 words and SentiWordNet 117,658 We will create three separate HashMaps which will have the word as key and emotion and the sentiment as value. Thus, the HashMap will contain word as key and in value section it will have 9 numbers. First containing the sentiment score and then 8 number which will contain emotion score for joy, trust, surprise, etc. For NRC word list the values will be 1, 0 or -1 for sentiment and 1 or 0 for the emotions. For SMART word list the sentiment value will be between -5 to +5 with both the number included and for emotion the number can vary from 0 to +5 with both the numbers inclusive. We also load the SentiWordNet dictionary in a HashMap with word plus the POS tag as key and the Sentiment score as value. Let's take an example of an entry in both the word lists. Firstly in SMART the entry is say the word 'impossible, sentiment score – -3, emotions present – sadness, surprise' then the HashMap entry will look like Key = impossible, value = -3, 0, 0, 0, 3, 3, 0, 0, 0. Here -3 is sentiment score second entry is for the emotion joy, the next one is trust, then fear, surprise, sadness, disgust, anger and lastly anticipation. Now let's take an entry in NRC word dictionary. Say the entry is the word 'protest, sentiment score – -1, emotions present – anticipation, disgust' then the HashMap entry will look like Key = protest, value = -1, 0, 0, 0, 0, 0, 1, 0, 1. Here -1 is sentiment score second entry is for the emotion joy, the next one is trust, then fear, surprise, sadness, disgust, anger and lastly anticipation. Let's say Sentiword net contains an entry of the word 'protest' then the entry will look like 'Key = protestNNS and Value= -0.63'. NNS represents noun in POS tags thus we have the entry 'protestNNS'.With a HashMap we can look up a word in either of these dictionaries and get a quick response of respective emotion and/or sentiment. There is an internal hash function at play in the java HashMap which enables us to access any word present in the data structure with O(1) complexity, making data access extremely fast.

Let us look at the algorithm present here again.

1. Load SMART word list into a HashMap.
   a. Entry example, key = ‘impossible’; value= ‘-3, 0, 0, 0, 3, 3, 0, 0, 0’.
   b. Values are 1= sentiment score, 2=joy, 3=trust, 4=fear, 5=surprise, 6=sadness, 7=disgust, 8=disgust and 9=anticipation.
   c. Here -3 is the sentiment weight. Since emotion present in this example are ‘sadness and surprise’ the respective values contain the number ‘3’.
2. Load NRC word list into HashMap.
   a. Entry example, key = ‘protest’; value= ‘-1, 0, 0, 0, 0, 0, 1, 0, 1’.
   b. Same values as SMART word list.
3. Load SentiWordNet into HashMap.
   a. Entry example, key = ‘protestNNS’; value= ‘-0.63’.
   b. Here the key is the word and its POS tag and value is sentiment weight on a -1 to 1 scale.

5.5 Step 3: Sentiment and Emotion Extraction Part 1 Emotion Labeling

This part of the process is extremely important as here the core of the system resides which determines the emotion of the sentence. This part may also prove to be complicated, as shown in Figure 13. Let us take an example of a sentence to understand how the system works. Let the Tweet after cleaning be “protests in ukraine are not an impossible situation for russian governments”. Here, there are two words which represent emotions and sentiments, the word ‘protest’ and the word ‘impossible’. Let the word ‘impossible’ be in the SMART word list and let the word ‘protest’ be in the NRC word dictionary and SentiWordNet word dictionary.

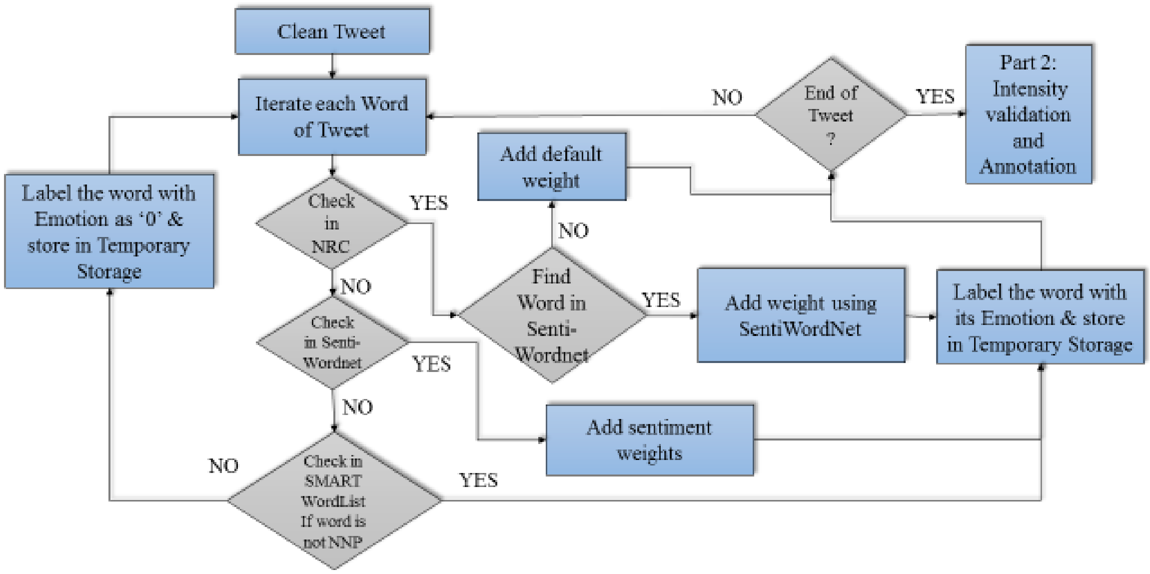

Figure 13: Step 3 Sentiment and Emotion Extraction Part 1 Emotion Labeling

Now let us look at the entries in SMART word list Table 4 and NRC word list Table 5. Since NRC has no intensities we can see in Table 4 it will contain one in the emotion which is present for that word and zero for the rest. Thus we have one present in the column labeled as ‘Disgust’ and ‘Anticipation’.

While for SMART word list, since we have emotion intensities, we can see in Table 5 that for the word ‘impossible’ the columns ‘Surprise’ and ‘Disgust’ both are labeled as 3. This means on a scale of 0-5 impossible has level 3 Surprise and level 3 Sadness. Let us assume in SentiWordNet we have the word ‘protest’ present which has a sentiment score of 0.4.

Table 4: Entry of the word protest in NRC word list

Word	Sentiment	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
protest	-1	0	0	0	0	0	1	0	1

Table 5: Entry of the word impossible in SMART word list

Word	Sentiment	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
impossible	-3	0	0	0	3	0	3	0	0

Now, when we feed the tweet to the system. Initially, all the words will have emotion and sentiment as zero and will contain their POS tag, as shown in Table 6.

Table 6: Initial condition of the word in the Tweet

Word	Sentiment	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation	POS
protests	0	0	0	0	0	0	0	0	0	NNS
in	0	0	0	0	0	0	0	0	0	IN
ukraine	0	0	0	0	0	0	0	0	0	NN
are	0	0	0	0	0	0	0	0	0	VBP
not	0	0	0	0	0	0	0	0	0	RB
an	0	0	0	0	0	0	0	0	0	DT
impossible	0	0	0	0	0	0	0	0	0	JJ
situation	0	0	0	0	0	0	0	0	0	NN
for	0	0	0	0	0	0	0	0	0	IN
russian	0	0	0	0	0	0	0	0	0	NN
government	0	0	0	0	0	0	0	0	0	NNS

There will be iteration of every word and each word will be checked first in NRC word list by forwarding the word as key in the respective HashMap. For the word ‘protest’ we will find an entry. Thus we will execute the next step in the process and check the same word in SentiWordNet by forwarding word plus its POS tag as key to the respective HashMap. In this case key will be ‘protestNNS’. In this case we get the sentiment score of -0.4. We will then take the sentiment score present in the SentiWordNet and extrapolate that on the scale of 5 to -5. In this case the extrapolated score will be -2. Now we will apply this score to the emotions as well. We will take the absolute value of the sentiment and apply it to the emotions which are present in the entry represented in the NRC word list. Thus for the word ‘protest’ the new values will be as shown in Table 7.

Table 7: word 'protest' after adding sentiment score from SentiWordNet

Word	Sentiment	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
protest	-2	0	0	0	0	0	2	0	2

We will then label that word with its respective emotion and sentiment and store it in temporary memory. The following words do not match in NRC thus they go to the next step where it is checked whether the word is a proper noun or not thus the word ‘ukraine’ will not be considered for emotion. We will eventually stumble upon the word ‘impossible’. This word is not present in NRC word list and neither is a proper noun. Thus we will check this word in SMART word list by passing the word itself as key to the respective HashMap. We will find an entry as described in Table 5. We will then label that word with the respective sentiment and emotion and store it in temporary memory. Now the sentence will look like Table 8. As we can see all the words which are not emotion words have zero in all their columns which indicate sentiment and emotion. And the words which have some sentiment value have their respective weights in the corresponding columns.

Table 8: Tweet after labeling emotion

Word	Sentiment	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation	POS
protests	-2	0	0	0	0	0	2	0	2	NNS
in	0	0	0	0	0	0	0	0	0	IN
ukraine	0	0	0	0	0	0	0	0	0	NN
are	0	0	0	0	0	0	0	0	0	VBP
not	0	0	0	0	0	0	0	0	0	RB
an	0	0	0	0	0	0	0	0	0	DT
impossible	-3	0	0	0	3	0	3	0	0	JJ
situation	0	0	0	0	0	0	0	0	0	NN
for	0	0	0	0	0	0	0	0	0	IN
russian	0	0	0	0	0	0	0	0	0	NN
government	0	0	0	0	0	0	0	0	0	NNS

5.6 Step 3: Sentiment and Emotion Extraction Part 2 Valance Shifters and Intensifiers

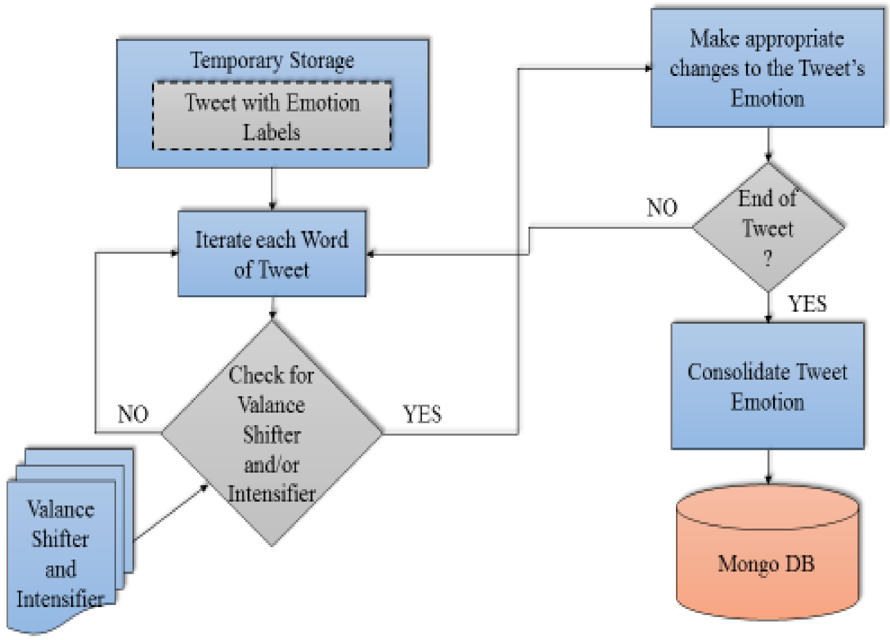

Figure 14: Step 3 Sentiment and Emotion Extraction Part 2 Valance Shifters and Intensifiers

Now we will move towards the last part of this system. Here the Tweet with emotion labels present in temporary storage, like in Table 8, will be forwarded to this stage. Here we will check the words against the valance shifter and intensifiers list. In this example we will come across the word ‘not’ which is a valance shifter. Now we will look for a word which comes after the word ‘not’ in this Tweet which is a sentiment word (which does not contain 0 in their sentiment column or any emotion column). In this case we will come across the word ‘impossible’. We will flip the sign of its sentiment and will swap the values in the emotions with its counterparts. In this case values in column ‘disgust’ and ‘surprise’ become zero and values in column ‘trust’ and ‘anticipation’ become 3.

This is because ‘trust’ is opposite of ‘disgust and ‘anticipation’ is opposite of ‘surprise’; the effect can be seen in Table 9.

Table 9: Tweet after Valance Shifter

Word	Sentiment	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation	POS
protests	-2	0	0	0	0	0	2	0	2	NNS
in	0	0	0	0	0	0	0	0	0	IN
ukraine	0	0	0	0	0	0	0	0	0	NN
are	0	0	0	0	0	0	0	0	0	VBP
not	0	0	0	0	0	0	0	0	0	RB
a	0	0	0	0	0	0	0	0	0	DT
impossible	3	0	3	0	0	0	0	0	3	JJ
situation	0	0	0	0	0	0	0	0	0	NN
for	0	0	0	0	0	0	0	0	0	IN
russian	0	0	0	0	0	0	0	0	0	NN
government	0	0	0	0	0	0	0	0	0	NNS

If it was an intensifier instead of valance shifter, then we would have increased the intensity of the next immediate emotion word by one. For example in case of a word like ‘very’ the intensity of the next emotion word would have been increased by one. I.e. if the word after ‘very’ was say ‘impossible’ then the primary state would have been like in Table 5. After applying changes for the intensifier ‘very’ the word will look like in Table 10.

Table 10: Word emotion and sentiment after intensifier

Word	Sentiment	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
impossible	-4	0	0	0	4	0	4	0	0

After end of tweet is reached we will add all the emotion scores and sentiment scores together like in Table 11. Then we will consolidate the emotions and sentiment as explained in CHAPTER 4 section 4.4.

Table 11: Total Sentiment and Emotion Score

Word	Sentiment	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
Consolidated	1	0	3	0	0	0	2	0	5

As we can see in Table 11 the system guesses correctly that the Tweet is ‘slightly positive’ and also guesses the top two emotions correctly with first being ‘anticipation’ and second being ‘trust’.

CHAPTER 6

RESULTS & EVALUATION

6.1 Evaluation Procedure

In this chapter we will talk about how we measured the accuracy of the system. Measuring accuracy was a challenge for the prototype of such type as we have to compare the accuracy with a human. With that notion in mind the question of reliability comes into picture. If we ask just one person to evaluate the system then the question arises regarding the perspective. One person may not see the results of the system from the same perspective of the other. Also since the system has the potential to evaluate on 32 different emotions the possibility of measuring the same is that much more difficult. Classifying the text into 32 classes may be overwhelming for humans thus measuring the accuracy on a scale of 32 is challenging.

For evaluation we looked into several ways and finally settled on our own approach. We decided to classify the emotions into eight categories instead of 32. Mainly because with experimentation we found that it was overwhelming for evaluators to classify with 32 classes. We also found that classifying into eight emotions was manageable, easy to evaluate and also less time consuming. Mohammad et al. {Mohammad, 2013 #28} faced the same problem. They decide to go with a smaller set of emotions for evaluation (middle ring of Plutchik's wheel of emotions) for the sake of evaluation. The other problem Mohammad faced was that no two evaluators necessarily agree with each other. This was because in his case they asked human evaluators to classify tweets on all eight emotions and with that there was more scope for disapproval and confusion among evaluators.

In our approach we present evaluators with minimum of 75 tweets to evaluate which are classified by the system. We arranged the output of the system such that it provided the top 3 emotions for each Tweet. This was possible because we had added weights to our emotions unlike NRC word list. Let's take the example of Table 11. Here the most prominent emotion was 'Anticipation' since its score was the highest (in this case five). Second most prominent emotion was 'Trust' (with a score of three) and third most prominent emotion was 'Disgust' (with a score of two). Similarly we gave an excel sheet with Tweets to evaluate for the evaluators which looks like Figure 15.

	A	B	C	D	E	F	G	H	I
1	Tweet	Sentiment	Emotion 1	Emotion 2	Emotion 3	1st Most Prominent Emotion Present in position	2nd Most Prominent Emotion Present in position	3rd Most Prominent Emotion Present in position	Sentiment Accuracy
2	Praying for Ukraine...	slightly positive	Sadness	Fear	Anticipation	0/1/2/3	0/1/2/3	0/1/2/3	Correct/Incorrect/ Partially Correct

Figure 15: Results and Evaluation Sheet

For this endeavor we gathered the Tweets about 'Israel-Gaza' and 'Ukraine-Syria' conflicts as explained in CHAPTER 5 section 5.2. These Tweets were different from the Tweets we gathered for creating the Word List. These Tweets were little over one million. We then extracted 1000 random tweets using the command db.UkraineIsrael_v1.find({ random_point : { $near : [Math.random(), 0] } }).limit(1000) in MongoDB. We extracted these Tweets then cleaned them. Upon cleaning some Tweets which had only a username or just a hyperlink were gone. This reduced the total number of Tweets to 983. We divided these Tweets into documents like the one shown in Figure 15. And distributed them for evaluation. The distribution was done such that each

Tweet was evaluated by at least two evaluators to maintain consistency and to measure the deviation of evaluation.

We classified the sentiment as follows

- Very Positive
- Positive
- Slightly Positive
- Neutral
- Slightly Negative
- Negative
- Very Negative

And we classified the emotions as

- Joy
- Sadness
- Surprise
- Trust
- Anger
- Disgust
- Fear
- Anticipation
- None (being no emotion present)

In Figure 15 the columns labeled as ‘Emotion 1’, ‘Emotion 2’ and ‘Emotion 3’, are emotions in a given Tweet. They are presented in order of their prominence, i.e. ‘Emotion 1’ is the most prominent emotion, ‘Emotion 2’ is the second most prominent emotion and ‘Emotion 3’ is the 3rd most prominent emotion in the Tweet. The columns labeled as ‘1st Most Prominent Emotion Present in position’ , ‘2nd Most Prominent Emotion Present in position’ and ‘3rd Most Prominent Emotion Present in position’ will provide the user to choose from the possible results. The possible answers to these questions are 0/1/2/3. For example, let’s say the software generated a result such as follows Table 12.

Table 12: Example Annotated Tweet

Tweet	Sentiment	Emotion 1	Emotion 2	Emotion 3	What is the Tweet About?
Praying for Ukraine...	slightly positive	Sadness	Fear	Anticipation	Ukraine

If the evaluator thinks here the 1st most prominent emotion is *Fear, Sadness* is the 2nd most prominent emotion and 3rd most prominent emotion is *disgust* but it is not present in the options then his answer will look like Table 13.

Table 13: Model Answer for Table 12

1st Most Prominent Emotion Present in position	2nd Most Prominent Emotion Present in position	3rd Most Prominent Emotion Present in position
2	1	0

If there is just one emotion present in the Tweet then the system should provide output as follows Table 14.

Table 14: Example output of the system for one emotion

Ukraine Moves Toward Martial Law as Western Region Splits http://tco/2aKfkOpx3h	Slightly Negative	Sadness	None	None

In this case the evaluator can either say

Table 15 or Table 16 both responses are correct.

Table 15: Model answer for Table 14

1	0	0

Table 16: Model answer for Table 14

1	1	1

For Tweets with no emotions in them the system should provide output as Table 17

Table 17: System output for Tweets with no emotions

S/o to those in the ukraine right now	Neutral	None	None	None

Your answer to this can be either Table 18 or Table 19 both responses are acceptable.

Table 18: Model answer for Table 17

1	1	1

Table 19: Model answer for Table 17

1	2	3

6.2 Results

We evaluated results for 'Most prominent emotion', '2^{nd} most prominent emotion' and 'Sentiment score'. We also calculated the inter-evaluator agreement i.e. we checked how many times for 'Most prominent emotion' , '2nd most prominent emotion' and 'Sentiment score' the evaluators agreed.

For Emotion analysis results are as follows Figure 16.

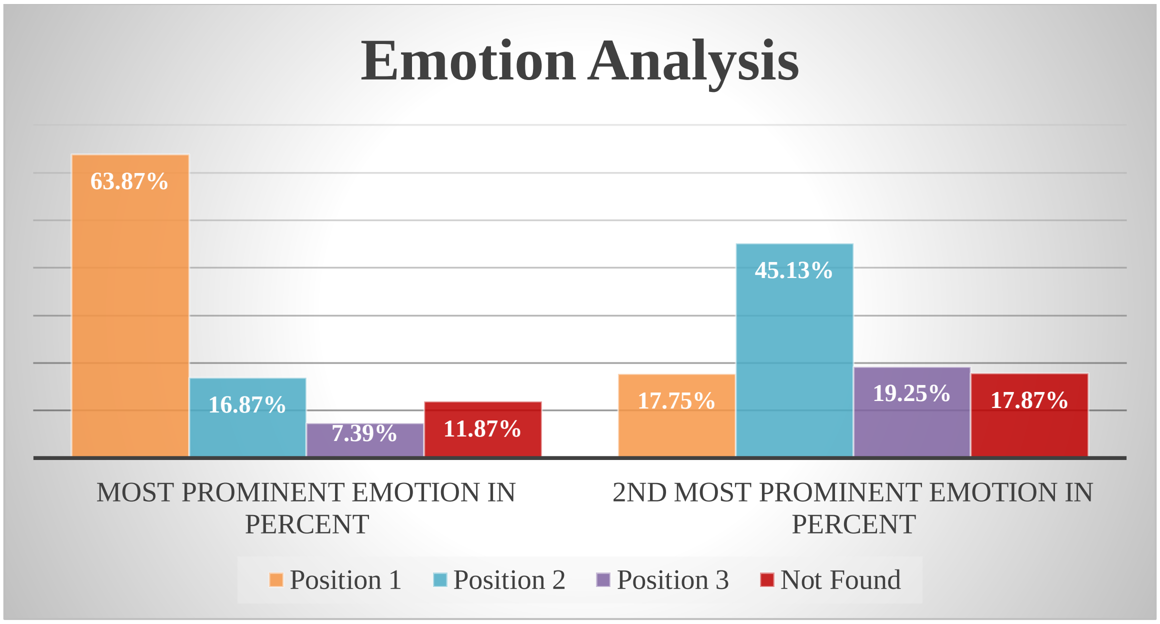

Figure 16: Results for Emotion Analysis

As we can see 63.87% of the time we guessed the 'Most prominent emotion' correctly. We guessed 'Most prominent emotion' as '2nd most prominent emotion' 16.87% of the time. We got the 'Most prominent emotion' wrong 11.87% of the time.

Also see 45.87% of the time we guessed the '2nd most prominent emotion' correctly. We guessed '2nd most prominent emotion' as 'most prominent emotion' 17.87% of the time. We got the '2nd most prominent emotion' wrong 17.87% of the time.

As we can observe most of the emotions present in the Tweet are covered by the first and second most prominent emotion presented by the system. Thus the accuracy is minimum 63.87%. It can be improved by incorporating a module to identify the 'Most prominent emotion' from the top three choices.

Results for sentiment analysis are as follows Figure 17

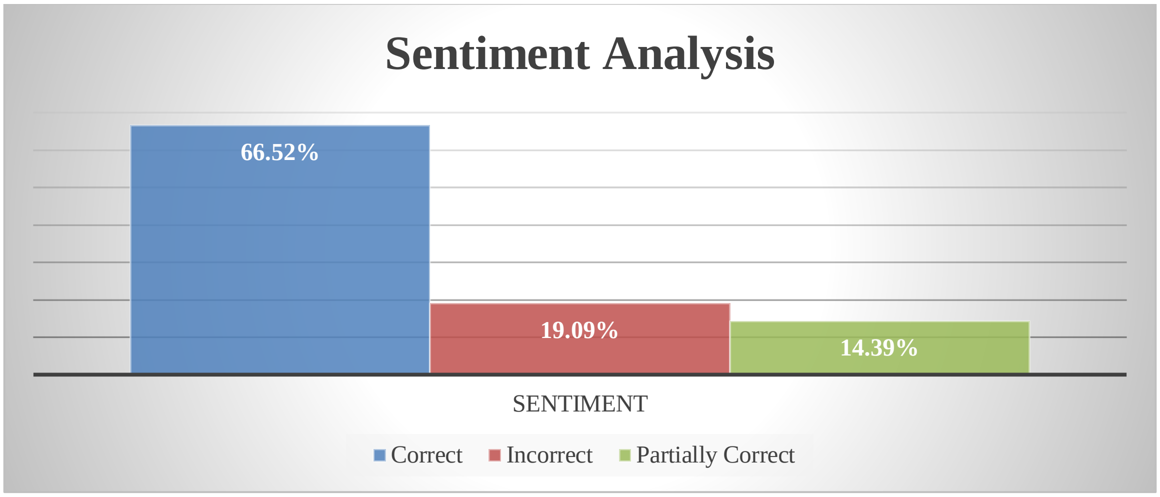

Figure 17: Results for sentiment analysis

As we can see we guessed the sentiment correctly 66.52% of the time. We partially guessed the sentiment correctly 14.39% of the time and we got the sentiment wrong 19.09% of the time. We can observe here that we got the sentiment wrong only 19.09% of the times which mean the accuracy is actually more than 66.52%.

6.3 Inter-Evaluator Agreement

Here we will talk about how much did evaluators agreed with each other. What we are trying to imply here is how much reliability is displayed by evaluators. We gave some Tweets to two separate evaluators to get these statistics. When two evaluators checked the same Tweet for accuracy we observed their level of agreement.

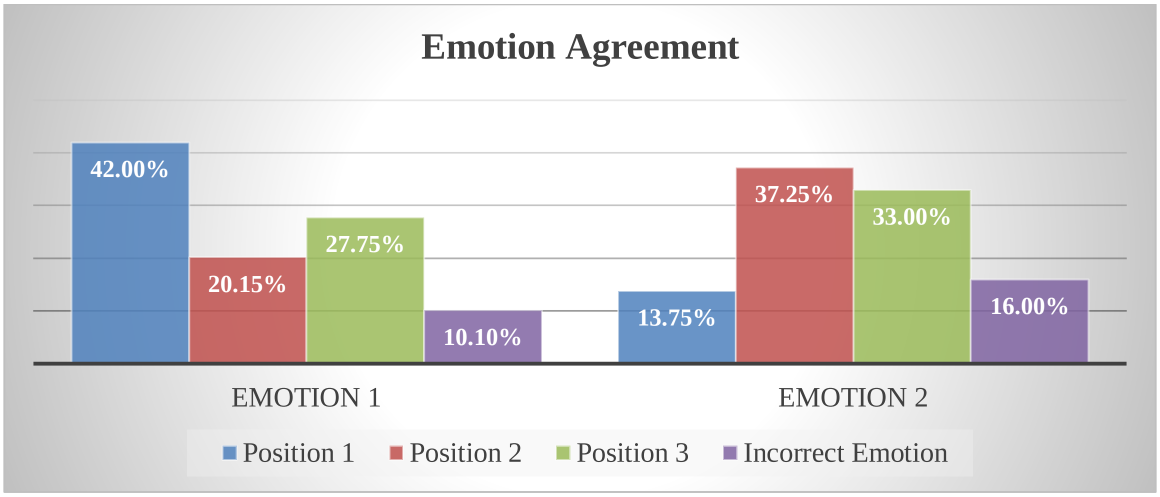

Figure 18: Inter-evaluator agreement for Emotions

As we can see in Figure 18. Evaluators agreed that the 'Most prominent emotion' (i.e. Emotion 1 in the above diagram) was guessed correctly 42% of the time, whereas for Emotion 2 the agreement was 37.25% for correctness. For incorrectness of the system the users did not agree with each other much. As we can see the evaluators agreed on the incorrectness only 10.10% of the time and 16.00% of the time for emotion 1 and emotion 2 respectively.

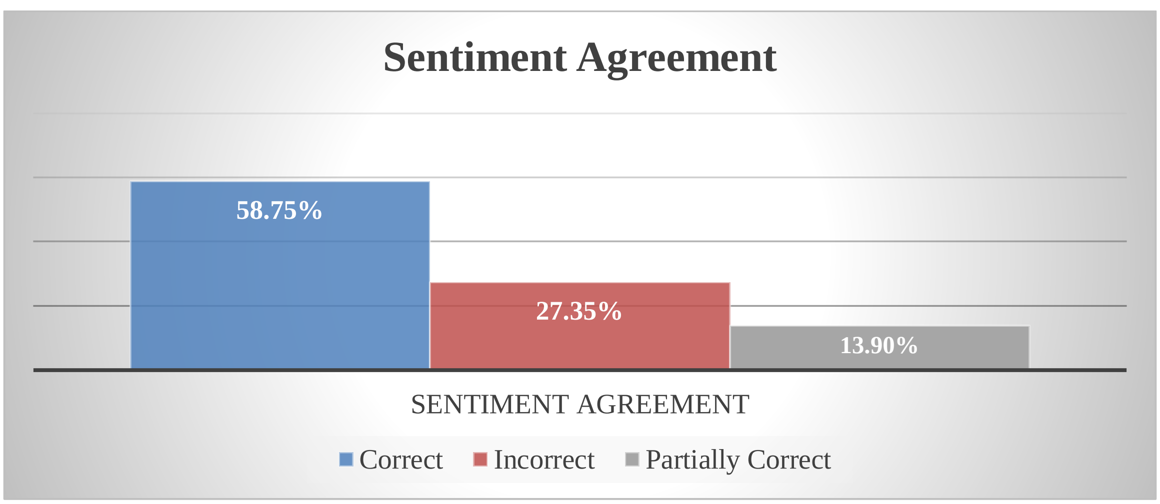

Figure 19: Inter-evaluator agreement for sentiment

As we can see in Figure 19 the same can be observed as compared the emotion analysis. Evaluators agreed with each other 58.75% of the time when the system guessed the sentiment correctly. When it did not on the other hand, evaluators only agreed with each other 13.90% of the time.

This give us perspective on nature of evaluation. When the system guessed an emotion/ sentiment correctly we can see that the evaluators agreed with each other most of the time. When the emotion/sentiment was guessed incorrectly on the other hand the evaluators had a split opinion, meaning some though the systems results were correct and some thought they were not. This shows that the system accuracy is actually higher than what we have shown in Figure 16 and Figure 17.

6.4 Valuable Observations

We observed that some emotions had a direct correlation with its corresponding sentiment. Whenever a Tweet contained emotions like 'Fear' or 'Anger' the Tweets almost always contained negative sentiment. Whenever the Tweets contained positive sentiment then Tweets also contained one or both of 'Joy' and 'Trust'. Keep in mind all the aforementioned emotions were present alongside other emotions.

On the other hand we observed most of the time when the Tweet was classified as 'Positive' sentiment the Tweet did not contained 'Anger' in any of the possible emotions it offered.

These observations offer us with a possibility of correlation between emotions and sentiment in a way that presence or absence of a sentiment or emotion gives rise to the possibility of having its counterpart present in the results.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

We have presented a novel prototype which can work in the context of Social Conflicts related conversations on Twitter to extract useful information like emotions and sentiments. We have tackled challenges like identifying emotion indicators, creating a new bag of words, checking context of a word etc. We successfully implemented a platform that took into perspective ‘Sentiment weights’ for emotion analysis, use of valance shifters and the use of low level context identification. We achieved a minimum accuracy of 63.87% which was 10% more than the base system NRC Emotion Analyzer (accuracy 54%). We also observed that there was clear correlation between some of the emotions and some of the sentiments. Since we are targeting real time data a system with a low execution time was a necessity and we achieved that with an execution time that varies between 3-5 milliseconds for each Tweet.

Also, we selected Twitter as a domain because as discussed in CHAPTER 1 it contains vast quantities of potential public opinion which is vital for the vision of SMART Barometer discussed in CHAPTER 4. The main objective here is to offer a platform which is fast, accurate and flexible with low system overhead. As shown in CHAPTER 5 and 6 we have achieved these goals. The results of this system will help researchers to detect patterns in social conflicts and help them in analysis and/or prevention of the same.

7.2 Future Work

Many improvements can be made to the system to enhance its performance and scope. For example adding a new custom bag of words will enable the system to be used in other domains such as ‘bully detection on Twitter’ and ‘identifying possible suicide victims on Twitter’. These are some fields in which this prototype can work. We can also use this system with, minor modifications, for identifying emotions and sentiment for ‘Traditional Media’. There is also a potential to have an add-on which may improve the word list on the fly such that we can improve the accuracy as much as we use the system. There is a possibility to introduce retweets and favorites into the weighing system. As the Tweets which are re-tweets or favorites tend to be more important and widely accepted by Twitter users as compared to others. Incorporating emotion density is also an important step. Emotion density is number of emotion words divided by the total number of words in a Tweet. This gives us the measure of how much emotion resides in a given Tweet and provides us insight into the emotion value of that Tweet. We have also observed a potential to include Ontologies into this system. As the data we offer is available in JSON format we can map it easily onto Ontologies. Ontologies offer the possibility of understanding a conflict and identifying different patterns.

REFERENCES

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). *Sentiment analysis of twitter data.* Paper presented at the Proceedings of the Workshop on Languages in Social Media.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). *SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.* Paper presented at the LREC.

Bakshy, E., Hofman, J. M., Mason, W. A., & Watts, D. J. (2011). *Everyone's an influencer: quantifying influence on twitter.* Paper presented at the Proceedings of the fourth ACM international conference on Web search and data mining.

Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record, 39*(4), 12-27.

Chodorow, K. (2013). *MongoDB: the definitive guide*: " O'Reilly Media, Inc.".

Das, S. R., & Chen, M. Y. (2007). Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science, 53*(9), 1375-1388.

Davidov, D., Tsur, O., & Rappoport, A. (2010). *Enhanced sentiment learning using twitter hashtags and smileys.* Paper presented at the Proceedings of the 23rd International Conference on Computational Linguistics: Posters.

Davies, A., & Ghahramani, Z. (2011). Language-independent Bayesian sentiment mining of Twitter.

De Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). *Generating typed dependency parses from phrase structure parses.* Paper presented at the Proceedings of LREC.

Ekman, P. (1992). An argument for basic emotions. *Cognition & Emotion, 6*(3-4), 169-200.

Esuli, A., & Sebastiani, F. (2006). *Sentiwordnet: A publicly available lexical resource for opinion mining.* Paper presented at the Proceedings of LREC.

Godbole, N., Srinivasaiah, M., & Skiena, S. (2007). Large-Scale Sentiment Analysis for News and Blogs. *ICWSM, 7*.

Gosling, J. (2000). *The Java language specification*: Addison-Wesley Professional.

Habibi, M. (2004). *Java Regular Expressions: Taming the java. util. regex Engine*: Springer.

Hu, X., Tang, J., Gao, H., & Liu, H. (2013). *Unsupervised sentiment analysis with emotional signals.* Paper presented at the Proceedings of the 22nd international conference on World Wide Web.

Hubmann-Haidvogel, A., Scharl, A., & Weichselbraun, A. (2009). Multiple coordinated views for searching and navigating web content repositories. *Information Sciences, 179*(12), 1813-1821.

Krech, D., Crutchfield, R. S., & Livson, N. (1974). *Elements of psychology*: Alfred A. Knopf.

Kriesberg, L. (1973). *The sociology of social conflicts*: Prentice-Hall Englewood Cliffs, NJ.

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). *What is Twitter, a social network or a news media?* Paper presented at the Proceedings of the 19th international conference on World wide web.

Lewin, K. (1945). Resolving social conflicts.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies, 5*(1), 1-167.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM, 38*(11), 39-41.

Mitchell, A., & Hitlin, P. (2013). Twitter reaction to events often at odds with overall public opinion. *Pew Research Center, 4*.

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*.

Mohammad, S. M., & Turney, P. D. (2013). *NRC Emotion Lexicon*.

Mohammad, S. M., Zhu, X., & Martin, J. (2013). Semantic role labeling of emotions in tweets.

Nasukawa, T., & Yi, J. (2003). *Sentiment analysis: Capturing favorability using natural language processing.* Paper presented at the Proceedings of the 2nd international conference on Knowledge capture.

Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C. (2009). Comparison of JSON and XML Data Interchange Formats: A Case Study. *Caine, 2009*, 157-162.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval, 2*(1-2), 1-135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Thumbs up?: sentiment classification using machine learning techniques.* Paper presented at the Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10.

Plutchik, R. (2001). The nature of emotions. *American Scientist, 89*(4), 344-350.

Polanyi, L., & Zaenen, A. (2006). Contextual valence shifters *Computing attitude and affect in text: Theory and applications* (pp. 1-10): Springer.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: electronic library and information systems, 14*(3), 130-137.

Saif, H., He, Y., Fernandez, M., & Alani, H. (2014). Semantic Patterns for Sentiment Analysis of Twitter *The Semantic Web–ISWC 2014* (pp. 324-340): Springer.

Sanner, M. F. (1999). Python: a programming language for software integration and development. *J Mol Graph Model, 17*(1), 57-61.

Shellman, S. M., Covington, M. A., & Zangrilli, M. (2010). *State of the practice and art in sentiment analysis*. 4th Intl. Conference on Applied Human Factors and Ergonomics.

Strapparava, C., & Valitutti, A. (2004). *WordNet Affect: an Affective Extension of WordNet.* Paper presented at the LREC.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics, 37*(2), 267-307.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). *Feature-rich part-of-speech tagging with a cyclic dependency network.* Paper presented at the Proceedings of the 2003 Conference of the North American Chapter of the

Association for Computational Linguistics on Human Language Technology-Volume 1.

Tually, S., Beer, A., & Faulkner, D. (2007). Too Big To Ignore. *Future Issues for Australian*.

Turney, P. D. (2002). *Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews.* Paper presented at the Proceedings of the 40th annual meeting on association for computational linguistics.

Weichselbraun, A., Gindl, S., & Scharl, A. (2013). Extracting and grounding context-aware sentiment lexicons. *IEEE Intelligent Systems, 28*(2), 39-46.