A Sarcasm Detection Framework in Twitter and Blog posts Based on Varied Range of Feature Sets

by

Hamed Minaee

B.Sc. in Industrial Engineering, Islamic Azad University of South Tehran, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Faculty of Computer Science

Supervisor(s):	Ali A. Ghorbani,Ph.D.,Computer Science
Examining Board:	Huajie Zhang, PhD, Faculty of Computer Science, UNB (Chair) Ebrahim Bagheri, PhD, Faculty of Engineering,Ryerson University; Faculty of Computer Science, UNB Donglei Du, PhD, Faculty of Business Administration

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

September, 2016

Abstract

This thesis addresses the problem of sarcasm detection by using a framework which is designed to effectively detect sarcastic blog and microblog posts. This framework consists of two components. Each component consists of different sub components including crawler, preprocessing and classification. The long text sarcasm detection classification consists of a two-step process, in each step, we use some feature sets along with different classifiers. These feature sets are utilized to analyze each blog post as a whole in addition to every isolated sentence. In the first step, Scoring Component is used to classify the documents into groups of sarcastic and non-sarcastic. Also in order to find sarcastic sentences in each sarcastic document, Decision Tree is applied. Considering the difficulties in sarcasm detection, the Document Level Sarcasm Detection achieved an outstanding result: 75.7% Precision rate. In the Short Text, Decision Tree is applied in order to classify the tweet texts into groups of sarcastic and non-sarcastic. Precision of 86.6% is obtained for this component which is very good considering the difficulty of sarcasm detection as well as inherent complexity of Twitter texts.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The rapid growth of social media, such as Twitter and blogs, has dramatically increased demand for analyzing the content of posts. Governments, corporations, and non-profits would all greatly benefit from the ability to better comprehend the conversations that are taking place in the digital sphere. Information retrieval is able to shed light on numerous domains such as text categorization, sentiment analysis, topic extraction and issue discovery and gender classification. Furthermore, different techniques can be used in information retrieval such as pattern matching, lexical analysis, syntactic structure and linguistic rules.

All these applications allow for greater insight of those who interact in the digital world. Amongst the most common applications of information retrieval is to discover the thoughts, feelings, and perspectives of social media users , extract the viewpoint(s) of a post span ,and identify the sentiment

orientation of society towards an individual, entity or a topic through blogs and short posts. Generally, most who use social media freely and extensively express their ideas, thoughts, and feelings on multiple platforms (i.e. twitter, facebook, blogs, and reddit). Hence, information retrieval is a key to keep a finger on the pulse of communities and societies at large. Social Media is a robust and free source of information with mountains of data ripe for analysis. Consequently, many successful companies have realized the importance of tracking and analyzing social media to gain a competitive edge.

Sarcasm detection, which is closely related to all of the topics previously discussed, is an important topic in this field. Sarcasm is defined as the rhetorical process of intentionally applying words or phrases for conveying a meaning different from what literally has been said. The explosion of online information, either blogs or microblogs demands new approaches to interact with text data and sarcasm detection is an example of that. The effect of sarcasm on different area in this field is an undeniable fact. In today's world, with the rapid growth of technologies people have more chance to express their feeling and opinion toward a person, product or an issue freely and the effect of these opinions is an undeniable fact. For example in [41], it is shown how a viral sarcastic tweet can ruin a company public image as a result of poor customer service.

The accuracy of sarcasm detection depends on every aspect of language; from the lexical to the semantic. Ergo, detecting sarcasm needs multiple parameters in place to be effective; Grammar and context are examples of

features that are often used. In most previous research, sarcasm detection is heavily reliant on binary classification. However, in this thesis, we propose a novel framework that not only uses binary classification to group posts into two categories (sarcastic and non-sarcastic), but we also score each texts according to their level of sarcasm and group them based on their scores.

1.1 Motivation

The occurrence of sarcasm in social media such as blogs, microblogs has dramatically increased and detecting this phenomenon is an inherently difficult task even for human. Also sarcasm is considered to be a very common element in English and it is used in many cases to ease the communication. One might use sarcasm to criticise someone else as another might use it to ignore answering unpleasant questions. So due to the common use of sarcasm in English, its automatic detection is very important. Sarcasm is considered as the complex form of language and have effect on other natural language processing tasks. Tasks such as sentiment analysis, question answering and text summarization are an example of that. Also the inherent complexity of social media such as space limitation and using hashtags can add to the complexity of sarcasm detection. Considering all of the mentioned complexities, having a framework which is able to detect sarcasm based on different feature sets is necessary. In this thesis, we propose a framework which is able to detect sarcasm in short texts and long texts using different feature sets

and classifiers.

1.2 Summary of Contributions

The main goal of this thesis is to develop a framework for sarcasm detection. The contributions of the thesis can be summarized as follows:

- A framework for detecting sarcasm in blog posts as the representative of long texts and detecting sarcastic sentences in each document.
- A framework for detecting sarcasm in tweet texts as the representative of short texts considering limitations and inherent complexity of tweet texts.
- Application of varied range of feature sets to detect sarcasm in both short texts and long texts and determine the rank of each feature.
- Comparison of the performance of different classifiers in sarcasm detection and choose the best one to be utilized in the components of the framework considering both recall and precision.

1.3 Thesis Organization

The rest of the thesis is organized as follows:
In Chapter 2, some of the related works in the area of sarcasm detection are reviewed. Some challenges in this research area are briefly reviewed. Also

some classification algorithms and common features used in this area are explored. Chapter 3 provides the details of proposed framework for sarcasm detection in long texts. This chapter explores sarcasm detection in document level as well as sentence level by using different set of features and two different classification algorithms. Furthermore, a detailed discussion of feature selection and cluster's analysis is provided.

In Chapter 4, sarcasm detection in short text is explored. Twitter as a rich source of data is used. Also a detailed discussion of feature selection is provided.

Chapter 5 presents the experiments conducted to benchmark the performance of components of the framework. A detailed discussion of classification algorithm results is provided. At the end this chapter a survey is reported which is conducted in order to show the difficulty sarcasm detection task.

Finally, the conclusion and some suggestion for further work are given in Chapter 6.

Chapter 2

Literature Review

2.1 Introduction

Sarcasm is defined as a means of conveying the opposite meaning in order to be funny or to make a point. Researches on language disorders such as sarcasm do not only limit to language. In [32], a research has been made on sarcasm as an example of pragmatic inference and considered psychological processes underlying comprehension of sarcasm and types of inference. Sarcasm is considered an indirect form of speech in order to affect its target dramatically. Also irony is usually used to intensify the occurrences that deviate from what is expected, while sarcasm is considered as a version of irony to criticise implicitly to target an audience as a victim in a polite way. To this end, it is very common that a sarcastic comment is wrapped with negatively worded utterance to convey a feeling of praise [32].

Also as is mentioned in [22], the consideration of the reason why a speaker is being sarcastic is important. A reason for being sarcastic is conceptualized as an indirect negation. It is believed that sarcastic form of speech is simpler than direct forms such as criticism. To illustrate the latter, [32] pointed at the following example: "He is a genius"; to directly negate this sentence we need to say it this way: "He is not a genius" which means that he is less than genius but this does not transmit the meaning that he is fool. However, if we say the latter sentence sarcastically the exact meaning is that the person is fool. So this shows that sarcasm makes the communication easier. Also sarcasm can be considered as a tool to increase the perceived politeness of an utterance [8]. In [12], it is claimed that normal speakers consider sarcasm as less aggressive than criticism. Using sarcasm instead of criticism transmits less rudeness. So sarcasm can be considered as a polite version of criticism [27]. Although it is not possible to certainly depict that why a speaker is being sarcastic, there are different reasons can be mentioned as follows[32]:

- Being playful by using word play to transmit the meaning to have a dramatic effect on the target.
- To apply it as a non-threatening procedure to transmit the speakers or writers point of view.

There is an important theory in sarcasm inference called relevance theory. In [49], it is argued that the interpretation of the sarcastic comment considered the most relevant. For sake of example A lovely day for a picnic indeed! is

considered as relevant since it reflects a knowledge about a fact which has taken place at an earlier time. For example, something very annoying has happened which the speaker complains about it this way . Speakers react to the earlier incidents but by doing that they also convey their angers or complaint about it[32].

It is clear that a sarcastic comments may create different inferences. It is claimed that one of the most important inference transmit by sarcasm is counter factuality. According to the research [16], based on clinical and normal researches, the unexpectedness of sarcastic comments lead the audiences to infer sarcasm. Also in [32], sarcasm is analysed in terms of processing time. For more explanation, it is stated that usually sarcastic comments need more time for audiences to understand and response to it.

There are different phases needed for an audience to be able to detect sarcasm. Firstly he should observe some information about the environment to visualize the sarcastic comment in action. Then, considering the mental state including the prior knowledge and the status of speakers is important. Also other clues such as counter factuality of the the comment, the purpose of the communicaion and the intention of the speakers can be helpful [32]. Considering all of the above statements, still the inference of the sarcasm is very controversial [32] and this intensify the fact that sarcasm detection is very relied on different factors such as culture, beliefs and assertions,characteristics and might vary from person to person. So if sarcasm detection is a difficult task for humans, and its automatic detection is even harder.

2.2 Who Cares about Sarcasm

The occurrence of sarcasm in social media such as blogs, microblogs and etc. has dramatically increased and detecting this phenomenon is an inherently difficult task even for human [31]. Sarcasm is an inherent element of British and American culture and even there is a specific page in BBC news to teach sarcasm to foreigners. Based on the Oxford dictionary, sarcasm is defined as "a sharp, bitter comment." However in [7], sarcasm is defined as a means of conveying the meaning exactly opposite of what is sated in order to be funny or to show the point indirectly. To put it another way, one can assume sarcasm as the insincere form of politeness which will be used to be off putting [7]. Also traditionally, verbal irony was considered to be indirect where as sarcasm were assumed to be more direct. However, another view on sarcasm and verbal irony considers irony as a special case sarcasm.

Different factors such as context, culture, and specific topics and people involved in sarcasm can distinguish if a person is being sarcastic. One of the area in which sarcasm shows its importance is in sentiment analysis. In [31], they showed the importance of sarcasm detection in sentiment analysis; in this research it is claimed that using sarcasm detection in sentiment analysis has made a huge improvement on the result. In [31], they focused on hashtags. One interpretation of the effect of sarcasm on sentiment is to assume that it simply acts as a negation. For example, to show this, when one is said I love being bit by insects,he is using sarcasm and the polarity of this sen-

tence is flip flopped. However this can be a very naive approach considering the inherent complexity of sarcasm detection let a lone mixed with sentiment analysis. To show the later fact, [31], brought up the following example: "I am not happy that I woke up at 5:15 this morning..#greatstart #sarcasm;" in this tweet since we have "not" before word "happy", the polarity of the sentence is already negative and taking sarcasm into account the wrong result is obained. Also it is noteworthy to mention that sarcasm does not necessarily always change the polarity of the positive sentences. As is mentioned, Maynard and et. al focused on hastags to deal with the effect of sarcasm on sentiment analysis. hashtags can contain very helpful clues in capturing sentiment information, however since hashtags may includes more than a word, this can decrease the performance. So word tokenization and some preprocessing may be needed. In [31], an algorithm was developed to form a token and match them against a dictionary to extract meaningful words out of that though this might not be able to deal with complicated cases such as #greatStar. After this phase a set of rules are defined to identify the polarity based on considering sarcasm, for instance, if the polarity of the sentence is positive and the sentence is tagged with #sarcasm or any similar sarcastic related hashtag then the polarity of the sentence will be revested or if tweet is followed by a positive hashtag plus a sarcastic hashtag then we reverse the polarity of the whole text to negate it. However, just relying on hashtags to detect sarcasm might no be efficient since there are many cases in which sarcasm is generated but not any sarcastic hashtag used.

According to [4], in sentiment analysis, emotional words as a bag of words approach is widely used to measure the sentiment of the sentence which is unable to deal with the advanced forms of speech. One advance form of speech is sarcasm which is widely used in different social domains. In [4] two different set of features has been used to detect sarcasm along with the features used to analyze the polarity of the text. The features they used is similar to the one used in [6] which is as follows:

- Sentiment-related Features
- punctuation-related features
- syntactic features
- patterns

The accuracy for the proposed framework in [4], showed at least 20 percent improvement to the baseline after considering sarcasm. This proves the importance of sarcasm. Sarcasm detection is applied in different areas. Just as an example, In [10], sarcasm detection is used to recognize brain injury in an early stage. This research shows that people who suffers from brain injury have more difficulty to understand sarcasm compared to other form of speech.

2.3 Feature Engineering

Feature selection is one of the most important element of each machine learning task. The more relevant the feature set is, the more accurate the result would be. However, before finding features and processing them, some preprocessing on the text is mandatory to enhance the text. Some of the common text processing techniques are listed as follows:

- Tokenization: Token is the smallest unit we define in the machine learning tasks and everything is defined based on that. For instance a unit can be a number of words in a sentence or the number of posts posed by the user so far. In java and python as representatives of two famous programming languages, there are many good libraries to tokenize the content to smaller units such as sentences or words.

- Stemming and Lemmatization: Stemming and lemmatization are two common tasks in data mining. Stemming simplify the word by cutting all suffix to form the base form. This can be very helpful to reduce the dimensionality by finding the stem of all words and remove the similar words. Also lemmatization is similar to stemming but it takes the part of speech into account.

- Part-Of-Speech (POS-tagging): In grammar, a part of speech is a linguistic category of words. Part of speech consists of 8 categories including verb, adverb, preposition, conjunction, noun, adjective, pronoun,

and interjection.

2.3.1 Feature Selection

In this section we explore the common features used extensively in automatic sarcasm detection. Features are the key component of any machine learning task and researches on sarcasm detection have used a broad range of features from lexical to contextual.

One type of features commonly used in this area is textual features. In [15], four types of features are used and all of them are textual. The first and common feature they used is punctuation. In [15], they used a set of features containing the ratio of commas, full stops, colons,semi-colons, ellipsis, quotations, hyphens, question marks and exclamation marks normalized by the total number of punctuation in the text. Also they considered the upper case letter words in this category. Forslid and et all. claimed that a tweet is considered ironic if the combination of upper case letters and ellipsis is used. Another feature used in [15], is related to capture words which are good to express sarcasm and irony.

In [51], the effect of a very common combination "yeah right" in sarcasm generation is explored. In this research, they conducted some experiments to detect sarcasm using different cues such as prosodic, spectral, and contextual and they were able to confirm that tone alone is not enough to find the sarcastic texts. This shows the importance of other features such as textual features in sarcasm detection though the effect of tone in this area is an

undeniable fact.

Furthermore, in [15], they claimed that there are some compounds such as "yeah right" which occur a lot in sarcastic texts. In [15], another feature called vocabulary used to detect the tone of the text. To capture the most occurred tonic phrases they used porter stemmer to stem all the words and then, they compiled a list all unigrams and bigrams in their training set. Also structure of the text is shown to be an effective feature [15]. In the latter research, in order to discover the structure of a document, the ratio of adjectives, adverbs, nouns, prepositions, determiners, modal verbs interjections and verbs are calculated. Also it is claimed that Interjections are one of the most effective feature in sarcasm detection [15]. In respect to the interjections feature, in [9], it is claimed that some interjections are helpful in detecting irony. However, the context in which the interjections are appeared is important.

In [26], Kreuz and Caucci investigated the importance of lexical features in sarcasm detection. In order to generate sarcastic contents, they used Google book and they collected 100 sentences which contained the phrase said sarcastically. They examined these sentences in three dimensions:

- POS (adverbs and adjectives)
- Interjections
- Punctuation

101 participant were asked to rank these sentences based on the likelihood

of being sarcastic without showing the context to the participants. They used a metric of 0 for not sarcastic at all to 7 showing very sarcastic. After getting the result and performing regression analysis, an interesting result were obtained: only interjections were shown to be an effective feature in detecting sarcasm. This will justify the importance of interjections such as tonic words in a text. In this thesis as you will see, the effect of tone is very bold through the whole process of sarcasm detection in long text and short text.

Also for sake of comparison, in [42], they did a research on the relation of sarcasm and polarity. The common belief is that sarcasm usually happens in the sentences with positive polarity. However, in [15], they showed that 60.5% of the sarcastic sentences have positive polarity versus the rest which was labeled as sarcastic. This proves that irony and sarcasm do not always happen in the sentences with positive polarities.

In [1], they used different set of features since the data set used obtained from twitter resources, they were able to test different set of features from lexical to contextual. In terms of the textual features, they tried the following features and also they reported the accuracy of their classifier by using only these features in isolation:

- Word Unigram and Word bigram in which the most recurrent unigram and bigram were captured in the sarcastic texts. According to [1], unigrams such as dare, shocked,#lol and bigrams such as you mean, how dare, im shocked, im sure were mentioned as the most recurrent

phrases in the sarcastic texts. Also as is claimed in [1], the accuracy of these feature in isolation were 72.4% and 69% accordingly.

- Part of speech features; This feature is very common in sarcasm detection task and the accuracy reported for this feature was 66.0.

- Pronunciation features: this feature is aimed to capture the writing style of a twitter user. [41] conducted a very extensive research on the writing style of the twitter user for detecting sarcasm. This feature was claimed to have 57.5 % improvement on the accuracy alone. btw, brb are examples of this feature.

- Capitalization features which includes all words with capitalized letters or starting with capitalized letter. The accuracy reported for this feature was 57.5 %.

- Sentiment which was used in sentence level and also word level. In order to find the sentiment of the whole tweet they used Stanford library. Also in order to capture the writing style of the twitter user, the maximum sentiment, minimum sentiment and the distance between max and min were calculated. Also the accuracy reported was 55% and 53.3% respectively.

- Intensifiers: According to [28], hyperbole is the key element in generating sarcastic contents and it claimed that the presence of hyperbole in the text will increase the likelihood of ironic interpretation. In [41]

the accuracy of 50.1% reported for this feature.

In [14], mainly two sets of features is used. The first set is related to part of speech which is mainly relied on verb, noun and adjective. Also in order to tag the tweets they used tagger introduced in [36] which is a supervised part of speech tagger defined for twitter.The second set called pragmatic particles. Owoputi and et. all in [14], used a variety of features to elicit the non-literal meaning in a text; features such as punctuation, emoticons and initialisms such as ROLF and BM which are used to capture the facial expression is an example of that. Also another feature called Onomatopoeic expressions is used; in order to capture this feature in the text a dictionary compiled in [38] used.

In [23], a set of supervised learning methods used for detecting sarcasm; also different set of features including nastiness is used which is considered as a way of showing anger or complain in the text. In order to extract the feature set, the following mechanisms were used:

- Mechanical Turk Cues: which is used to extract the most recurrent n-gram (1-3) in 510 nasty documents and 617 sarcastic documents. This resulted in a set of phrases such as "oh really", "idiot", "your ignorance is" and so on [14].

- Linguistic Cues: Since relying on statistics needs a very large training set to have an acceptable accuracy, another set of feature called linguistic information was used in [23]. Also, the frequent patterns is

discovered and fit into the feature set. To show the effect of pattern as a feature, Pattern such as ADV ADV which is extracted from “Really? well” will be able to discover another sarcastic phrase such as “Really? then” even if it does not exist in the train set.

- Semantic information: which is used to capture the semantic effect of each word on detecting sarcasm. In order to capture this feature Justo and et. all used LIWC [37] dictionary consisting of 64 categories for words. So by comparing each word in the document with the categories of this dictionary, the most frequent categories in sarcastic contents can be obtained.

- Length information: which is claimed to be helpful due to the varying length of documents. So a vector containing the number of words, number of sentence, number of characters were used.

- Concept and Polarity Information: in order to calculate this feature, SenticNet-3.0 is used which is applied to extract the semantic information associated with the content. Also the sentiment score of the sentence is calculated.

in [33], two sets of feature are used. Textual features consist of some commonly used feature such as interjections, punctuation, positive and negative words and number of words. Also the following set of feature is used as well:

- Discourse Connectors

- Explicit Incongruity
- Largest Pos/Neg Subsequence
- Readability

In [5], the usage of sarcasm has been investigated from 3 perspectives: An individual might use sarcasm to act funny. In this way, the author tries to change the tone of his voice to be different than usual. However, in writing he might use different clues such as use of words with capital letter , exclamation and question marks and some sarcasm-related emoticons to transmit sarcasm. Also another usage of sarcasm mentioned in [5], is to show anger. When sarcasm is used as whimper, it transmits the bad situation by using exaggerative elements or using very positive words mixed with some negative expressions to describe a frustrating situation. The last interesting usage of sarcasm is when a person wants to avoid answering. This way he may use some uncommon words or sentiment related features or answering back with question. [5] relied on the above assumptions to extract the feature set. One feature they used was related to sentiment of the text which is a common feature in many sarcasm detection researches. Based on [45], sarcasm can be detected when a positive statement occurs in a negative situation. To capture this feature two lists of positive and negative words were collected. Also in punctuation related features, all punctuation, upper case words and n-gram equal character were included. In order to capture the scenarios where the speaker tries to avoid answering, some lexical and syntactical features such

as use of uncommon words, use of common sarcastic expression, interjections are applied. In order to enhance the feature set in the latter research, another feature called pattern was used. For more explanation, two types of pattern is used: grammatical patterns and content patterns.

In [40], an interesting comparison has been made between sarcasm detection in English and Czech. The set of features used is very similar to what is used in other works such as [43] and [44]. Features such as punctation, pointedness, n-gram and skip bigram are good example of features used. [41], explored sarcasm from behavioural point of view. In this work, they focused on sarcasm detection and it is claimed that although sarcasm detection is inherently challenging, the nature of twitter itself adds to this complications. Rajadesingan and et all mentioned two reasons to support the idea that sarcasm detection in tweet texts are more challenging:

- Twitter's nature of the text is inherently challenging. Challenges such as the ever evolving use of expressions and slangs can be a good example of that.
- Space limitation which is 140 per tweet limit the textual feature clue and this can add to complexity of the sarcasm detection problem.

In [46], the effect of cognitive complexity of an individual on the tweets posted by that user is explored. This is the key factor in determining the language complexity of the the tweets posted. So psychological and behavioral clues plays a prominent role in detecting sarcasm. In [46], they focused on different

features to address this. To do so, different set of feature used which all of them were based on contrast. For example, sarcasm as a contrast of present with the past is applied by tracking the change in the mood of tweet sentiment. The later shows the effect of imbalance in generating sarcasm. Also in terms of readability, sarcastic posts have been discovered to be hard to read. To this end in [46], Kincaid and et all. extended a known approach called Flesch-Kincaid and used it to measure the readability of the text. [25] proposed flesch as a formula to measure the readability of the text based on the syllables. However, the extended version of flesch used in [46], is based on the number of words. For more explanation the words are categorized based on the number of their characters; also they made a comparison to see if there is any noticeable difference between the number of words in current tweet and tweets posted in the past. Furthermore, user's expertise is another factor considered in [46]. For more explanation, the expectation is that a user uses sarcasm has a good command of English. So the user's language skill can be a helpful factor in detecting sarcasm. In [21], a formula called cloze test used to measure the skill of the author based on the usage vocabulary, grammar, spelling, and reading level. In order to determine the vocabulary skills of the user, the ratio of distinct words used by the user over total words used is calculated [46]. Also in order to measure grammatical skills, the investigation accomplished on how different part of speech is employed. For more explanation, after tagging the words in the sentence, they check for common grammatical mistakes and consider that as a negative point. Also

users familiarity and tweet familiarity are other features assessed as well [46]. In terms of user familiarity the number of tweets posted by user per day and the average number of daily tweets were calculated in order to measure the duration and ratio of user's twitter usage. Also in terms of tweet familiarity they captured the number of retweets and hashtags and mentioned used in the past tweets posted by the user. Rajadesingan and et all. also reported a very interesting results obtained from their framework:
first of all the feature importance analysis shows that mostly the first ten important features are textual. The following is the top sixth features resulted from [41].

- emoticons ratio
- adjectives ratio
- ratio of past positive words
- Number of polysyllables
- Lexical density
- ratio of past negative words

Also another interesting results shows that the performance of 79.38% is obtained which outperforms all the baselines[41]. By adding the users history an increase of 4.14% is obtained however the computational expense needs to be taken into account.

In [52], a program was designed to analyze vocal cues in a speech for finding any possible sarcasm. To do so, different features such as intensity, tempo and pitch have been tried. The results showed that intensity, longer pause and higher frequency which are all related to vocal tone are the most important features. However as is clear all of these features are related to vocal tone which is lost in the text. This is another proof for showing the difficulty of sarcasm detection in the text.

2.4 Classification

In every machine learning algorithm, one of the key step after defining feature set, is to train a model. For training a model there are different options which will be categorized into one of the following groups:

- Supervised Learning: In this method we separate the data into two groups of training and test set. The two most common types of supervised learning is regression and classification. Text classification is to assign label to text documents [34]. Also in order to measure the performance of classification, usually, a portion of the labelled data will be kept from the training set and after training the model with the training set, it is tested on the test set. Also in order to measure the performance of the algorithm, 3 important metrics is used including precision, recall and F-Measure. Precision is defined as the portion of the retrieved documents which are relevant. Also recall denotes the

number of the relevant documents which are retrieved. It is noteworthy that there is a trade off between precision and recall. For example if only the documents with the high degree of membership is assigned to the target class the precision is very high. However, this way many relevant documents is ignored which results in the low recall. Also the F-Measure is a mix of precision and recall [19]. There are many different classification algorithms and some of them are listed as follows:

- Index Term Selection: This method is helpful when documents have a lot of words. For more explanation, this method extract the most informative words instead of considering all words. This way the complexity of the classification decrease dramatically [19].
- Naive Bayes Classifier: This classifier is based on conditional probability. Also for document classification the specific order of the words or phrases in a document is not important. In order to reduce the complexity and increase the performance, the effect of the presence of a word in a document on other words is ignored. Although this model is not realistic due to the independence assumption, it has shown to be effective in many classification asks. Also using this classifier mixed with other classifiers can result in improvement of performance. For example in [35], the combination of expectation maximization(EM) and naive bayes classifier decrees the error rate up to 30%.

– Nearest Neighbor Classifier: In this classifier, instead of explicitly building models of different classes, documents from training set which are similar to the target document are selected. The target document is labeled based on the class label of the similar documents obtained from the training set. In terms of similarity measures, there are different options. A simple case would be to count the number of similar words in two documents. Also normalization needs to be considered in order to differentiate between documents with different length [19].

– Decision Trees: This classifier is based on set of rules which are applied to make decision sequentially. The procedure is to use divide and conquer to choose the feature which can predict the target class with the highest information gain. Decision tree is a very common algorithm in Data Mining since it is very fast and scalable in terms of both the number of features and size of training set. Also there are different variation of decision trees, for example, [48] use a simpler decision tree containing only one rule and an impressive results were obtained.

– Support Vector Machines(SVM): This classifier defines a hyperplane to separate positive and negative rows of the training set. There is a margin which is a distance between the nearest positive and negative document and the goal is to maximize them. A key factor in SVM is that it is independent of feature space dimen-

sionality. SVM is considered to be very effective in working with large feature space [20].

  - Hidden Markov Models: This model is good for classification tasks related to information retrieval. In [19], it is mentioned that standard classification approaches do not take the predicted labels of the surrounding words into account which is considered in Hidden Markov Models. This classifier is proved to work well in entity extraction [3].

- Semi-supervised Learning: in this machine learning algorithm, the goal is to make the use of mix of labeled and unlabeled data and take advantage of the result of the model to enhance the training set. The importance of semi-supervised learning would be more clear when we have a set of labeled and unlabeled data and it is expensive to label the unlabeled data [53].

- Unsupervised Learning: This type of machine learning algorithm is common when it is expensive or not possible to label data and we want to find a pattern in the data that we have. This might not be as accurate as supervised learning however it is better than supervised algorithm cost wise. Clustering is the most important example of this group.

  - Clustering: the purpose of clustering is to find groups of documents with similar content or features. So finding a pattern in

clustering is a key factor. Each cluster consists of some number of documents. Also the documents in one cluster should have more similarities with each other compared to the documents of other clusters. The performance of the cluster is considered as high when the similarity of the documents in a cluster and the dissimilarity of the documents in different clusters are both high [19]. Clustering has different varieties with different metrics as distance measure mechanisms[39]. One of the most common clustering algorithm is k-mean since it is simple to implement and can be applied to the large set of data sets [13].

One variant of k-mean clustering is Bi-Section-k-means. This variation of k-mean is a fast clustering algorithm specifically for text documents and it is claimed to outperform other agglomerative clustering techniques [50]. This algorithm repeatedly splits the largest clusters until obaining he acceptable number of clusters [19]. In [50], a comparison made between different clustering techniques. In this work, they compared the hierarchical clustering algorithms and bisecting kmeans. The result shows that bisecting algorithm performs better than regular k-mean.

2.5 Challenges

It is an undeniable fact that irony and sarcasm detection can be very important in different areas such as sentiment analysis and information extraction. However automatic detection of such a phenomenon is a difficult task. Negation is very common in language and several approaches have been developed to capture negation in the text [18]. Also in[17], the issue of negation has been explored from different perspective such as psycholinguistic perspective. Sarcasm and more generally figurative language can be considered as a form of negation.

In [42], the focus is on verbal irony. However as is mentioned before, in texts, verbal irony is considered as a type of sarcasm. For example in [11], sarcasm and verbal irony are considered the same. Based on [18], negation with all of its varieties such as false messages, contradiction,and sarcasm considered as the characteristics of human being. So detecting negation and specific form of that in language is important. In [42], in order to show the difficulty of irony detection they performed couple of experiments in detecting irony in document level and sentence level using 3 conceptual layers including signatures, emotional scenarios and unexpectedness. some layers also have their sub layers; for example Signature consists of pointedness, counter-factuality, and temporal compression. Also emotional scenarios includes features such as activation, imagery, and pleasantness. Unexpectedness contains temporal imbalance and contextual imbalance. An experiment was conducted to de-

tect irony in sentence level and a very poor result were obtained: only 6 out of 50 sentences regarded as sarcastic when the expectation was the contrary. So this shows that analyzing document in sentence level in order to detect sarcasm is not sufficient. This is another proof to show that sarcasm is very context dependant. For sake of an example the sentence "I never believed love at first sight until I saw this movie" could be considered as both ironic or positive depending on the context[42]. Another evaluation made was related to test the classifier on the whole documents instead of sentence level. The result was very promising which shows a considerable improvement.

Also in terms of polarity analysis, the results reported shows that 60.5 % of ironic documents are positive where as the remaining are negative. This proves that ironic contents are not always positive [42]. From all above said, it is clear that sarcasm detection is a very challenging concept and it goes beyond detecting some features and simple classification task.

Chapter 3

Long Text Sarcasm Detection

3.1 Document Level

In this section we analyze the documents to classify them into group of sarcastic and non-sarcastic. Figure 3.1, depicts the proposed framework for sarcasm detection in Document Level. As is depicted in figure 3.1, after crawling the data, the new document enters to preprocessing section. In preprocessing in document level all the punctuation is removed. Although punctuations such as ... or ! or ? are commonly used in sarcasm, the usage rate of these punctuation in non-sarcastic posts are also noteworthy which can be misleading for the sarcasm detection task. Also all non-alphabetic characters such as numbers, asterisks are removed. In scoring component, first the documents are scored and the model is build by clustering the documents based on their scores. After the newly arrived document is scored

and assigned to the right cluster.

3.1.1 Crawler

It is an undeniable fact that data is the fundamental part of every natural language processing task. Considering long text, it is very difficult to crawl data compared to Twitter since there is not a lot of API for crawling blogs and news. In this work, Word Press API used to collect the documents. This API is based on rest style HTTP GET. Rest API is simply a methodology to let programs which might be written in different programming languages to talk to each other. In order to crawl the data I use a table containing some predefined search terms. It is tried to choose the search terms which are

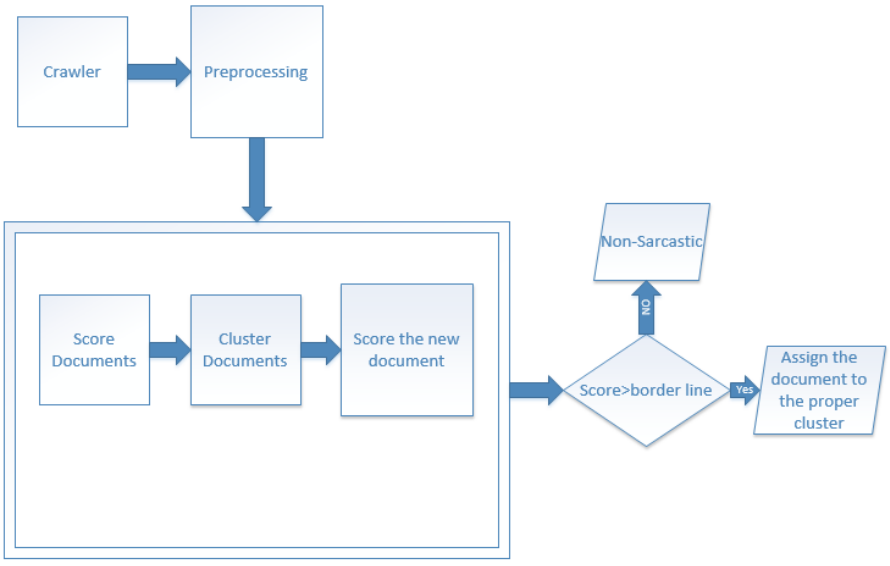

Figure 3.1: Document Level Sarcasm Detection component

more likely to crawl sarcastic posts. However, since the number of sarcastic posts are very low especially in blog posts and also word press API is unable to capture many of the sarcastic web sites, I manually added sarcastic posts from sarcastic blogs in order to enrich the number of sarcastic posts in the dataset. This way I made sure that the documents are truly sarcastic since they are extracted from sarcastic blogs. Word press API will be called in a loop each time sending s search term to the API and get the search result back.

The API response format is JSON which is a light-weight, language independent, data interchange format. Json is a widely used data format and makes it easy for the programs to communicate and send information back and forward.

3.1.2 Data set

Compared to Twitter, the availability of data set in long text is very low. This can be due to the fact that the popularity of microblogs such as Twitter increases daily and this can affect the usage of blogs and rate of blog audiences. As is mentioned in crawler section, in this thesis, due to the lack of good data set specifically labeled for sarcasm, I colleced a data set using WordPess API which crawls blog posts as a representative of long texts. Also since the traffic of sarcastic posts is very low in long texts, I manually added sarcastic documents extracted from sarcastic blogs to enrich the data set. In total 5069 posts were collected in which 595 posts are sarcastic and the rest

are labeled as non-sarcastic. Also for labeling the documents, some of them are clearly sarcastic since as is mentioned they are manually obtained from blogs which are specifically designed for sarcasm. Also for those which are crawled by the crawler, they are manually labeled. Furthermore, as will be discussed in this chapter, after finding the sarcastic documents we analyze them in sentence level to find the specific part in which sarcasm occurs. This has two advantages; firstly it decreases the complexity of sentence level sarcasm detection dramatically by limiting the number of sentences; secondly, if we know that a sentence is obtained from sarcastic documents, this gives us more flexibility to use different feature set to detect sarcasm. For example, if a sentence contains question mark and extracted from sarcastic documents it is more likely to be sarcastic compared to the question exist in non-sarcastic document. In order to test the performance of the feature set in sentence level, all the sentences of the documents resulted from document level sarcasm detection component were manually labeled. In order to correctly label the sentences they were considered in the context by reading the sentences before and after. In total 22045 sentences were labeled in which 6527 of them labeled as sarcastic.

Also in order to verify the data set labeling, since the number of sentences and documents is a lot, sampling approach is applied. For document level I chose a document from each centroid and for sentence level, two sentences were chosen from each cluster. Also 3 annotators were asked to label the documents and sentences into groups of sarcastic and non-sarcastic. In doc-

ument level, for 4 documents out of 5, majority of the annotators labeled documents as sarcastic which means 80% of the all samples. Also 8 out of 12 sentences were labeled as sarcastic using majority voting mechanism. 66% agreement between our manual labels and annotators labels in sentence level can be an acceptable degree considering the difficulty of sarcasm defection when it comes to people's attitudes and culture.

3.1.3 Document Level Feature Set

The following lists all the features used in document level to detect sarcasm in this thesis:

- Tone: as is mentioned before tone is a crucial factor in generating sarcasm. However, since a vocal tone is lost when it comes to text, it is very difficult to detect sarcasm. So, in order to generate sarcasm, authors, usually try to add some features or special words to transmit this meaning. Table 3.1 depict couple of examples of tonic phrases. Some of these tonic phrases or words might not have any meaning, but they are clear evidence of users attempt to bold the tone of the text. To capture this important feature I used bag of word approach. I collected 500 tonic words common in sarcastic contents. Also I collected a list including 500 common explicit language words used in sarcastic contents. It is noteworthy that although the usage of explicit language alone might not be enough evident for a text to be sarcastic, the essence

Table 3.1: Sample list of tonic phrases

number	Phrase
2	really!
3	Well, yeah
4	eh?
5	oh yeah
6	Wow
7	Oh Jeez
8	huh?
9	come on
11	hmmmm
12	hehe
13	Gosh!
14	so what!

of explicit language in the text increase the likelihood of sarcasm. After compiling the list of tonic words, I checked the document against the occurrence of words in this list. The more tonic words a document contains, the more is the likelihood of document to be sarcastic. This feature is normalized by total number of words in document.

- Part of Speech: In grammar, a part of speech is a linguistic category of words. For this section I used the following part of speech in the feature set:
  - Adverb which is the normalized weight of adverbs in the text except intensifiers which are categorized as a separate feature.
  - Adjective: This feature accounts for the total number of adjectives in the text divided by the total number of words.

In order to capture part of speech, I used the tagged form of the doc-

Table 3.2: A sample list of intensifier in text

number	Sentence
1	Basket makes so much money at his shows, its not even funny.
2	I like you too much to spoil what we have with a relationship WHAT?!!!
3	Thank you very much.
4	There is so much more to say, but I will leave it here.

uments tagged by NLP tagger.

- Intensifier: Intensifier is another feature which is tried in this thesis. Words which are used to add force to the meaning of verbs, adjectives or other adverbs are called intensifiers. The assumption is that sarcastic contents are more intense than non-sarcastic ones. For more explanation, exaggeration is an element of sarcasm and intensifiers are very good elements to bold this effect in the text. Table 3.2 shows some examples of intensifiers used in the text.

  As you can see from the table 3.2, the writer of these texts used some intensifiers such as too, very, so and etc. to intensify the text. Although some of these sentences may not be considered as sarcastic, it is an undeniable fact that the higher number of intensifier in the text will increase the likelihood of sarcasm. To measure the intensity of a document, I used a list including 30 adverb intensifiers. NLP tagger is used to tag all the words in the document. Then after, all the words are checked in the document against this list and the word coming after any of the matched intensifiers is checked to be adjective, verb or

Table 3.3: Sample list of sarcastic title

number	title
1	oh yes Beautiful Girls Are Ugly
2	5 Things Nobody Tells You About Sex
3	OH WELL, BACK TO BUSINESS
4	All I Want For Christmas LOOOLa dolphin
5	Oh, I look funny?
6	Experiment but dont be stupid
7	Are you F**KING KIDDING ME?
5	Doesn't She Look Like Fat Britney Spears? haha
6	HUMANS ARE STUPIDER THAN STUPID
7	How Funny Can The Endy Get?

adverb. If the word after the matched word is adverb,verb or adjective, the matched word is considered as adverb intensifier. However, it is noteworthy that the essence of this feature in an isolated sentence is not enough for the text being sarcastic since there are many cases where they contain this feature but they are not sarcastic. The aim of using this feature in document level is to measure the intensity of the text as a whole. This feature is normalized by the total number of words in the document

- Title: In some cases among the long documents, blog writers use some catchy and tonic words or even explicit language to catch the attention of readers. Table 3.3 shows some example of titles containing tonic words.

  To capture the essence of sarcasm in title I used two attributes: 1) Tone 2) Upper case. So this feature is the number of tonic words or upper case words occurs in the title text. This feature is normalized

by the total number of words in the title of the document.

- OOV: This feature is the normalized weight of out of word vocabulary in the document. For this feature, the assumption is that in sarcastic contents the number misspelled words or meaningless words is higher. This will be more noticeable if we compare sarcastic contents and other contents such as news and academic documents. So depending on the source of contents this feature may have more or less influence on the result. In our data set, our contents are mostly blogs. So since bloggers do not follow any special rules in writing and it is based on the personal writing style of the bloggers, the ratio of invalid words in non-sarcastic content is higher than expected. In order to capture the number of out of word vocabulary I used two different libraries. Firstly there are couple of cases in which words are valid but they do not have any synset. For more explanation I use Wordnet to detect if a word is valid or not. To do so, I feed a word to WordNet and if an array of size 1 or higher returned it shows that the word is valid in English. However, some words such as on, for and etc. which are prepositions and no synset returns for them should be considered as the valid words. To filter them out, I firstly used the tagged text which is produced by NLP tagger. If the tagger can find the word as an English word it will tag it and if not it will tag it as a noun by default. So all the words tagged as noun are filtered and are fed to synset. Also it is noteworthy that there are couple of cases such as everything, anything, nothing or

names which are the valid words but there are no Sysnset returned for them. To take care of these cases I compiled a bag of common words which are valid but they do no have synset and I exclude them before checking the words with the WordNet.

- Upper Case: Similar to many of the researches done in sarcasm detection, Upper Case words are very common in texts to intensify the specific part of the text and transmit the feeling of sarcasm in the text. This feature is normalized weight of all upper case words in the text.

- N-Gram Equal Character: N-Gram Equal Character is commonly used in sarcastic texts especially where the author wants to mock a person or make a joke about an annoying incident or events. This feature is the number of all words containing 3 or more equal characters. This feature is the normalized weight of all words containing 3 or more Equal Character.

3.1.4 Feature Importance Analysis

It is clear that not all of the above features mentioned are effective and we need to extract a subset of features which have an improvement on the result. In order to find the best features we need to rank them. In [2], they used Information Gain to rank the features. Similarly In order to rank the features in this thesis, I used Weka Ranker search method which use information gain to Rank attributes by their individual evaluations. The following equation

shows the Information Gain formula:

$$InfoGain(Class, Attribute) = H(Class) - H(Class|Attribute) \quad (3.1)$$

Information Gain is used to evaluates the worth of a feaure by measuring the information gain with regard to the class. The result from the evaluator shows that the following features perform better on the final results.

- Tone
- Intensifier
- Title
- Uppercase
- Adverb
- N-Gram Equal Char

You can see the rank of each feature obtained from Information Gain in figure 3.2. In figure 3.2, it is clear that Tone and intensifiers have the highest rank among all the others. Also Title and Upper case take the third and fourth place. Furthermore, Adverb and N-Gram Equal character showed to have an effect on the sarcasm detection task though there is a big gap between the rank of Tone as the best feature and Adverb and N-Gram Equal character as the least important features.

3.1.5 Scoring Component

In order to classify the documents into groups of sarcastic and non-sarcastic the customized scoring component used in [42] was applied. In [42] the importance level of features is not considered so this way less effective features may dominate more effective features. In order to consider the weight of each feature, the Weka Ranker with information Gain evaluator were applied. This way features with the higher discriminative power in sarcasm detection will be ranked higher. The following equation depict the scoring component:

$$SC(d) = (\sum_{i} fdf_i.w(i) + \ fdf_t.w(t)).v \tag{3.2}$$

Where i is the i-th feature and t stands for title used in the feature set.

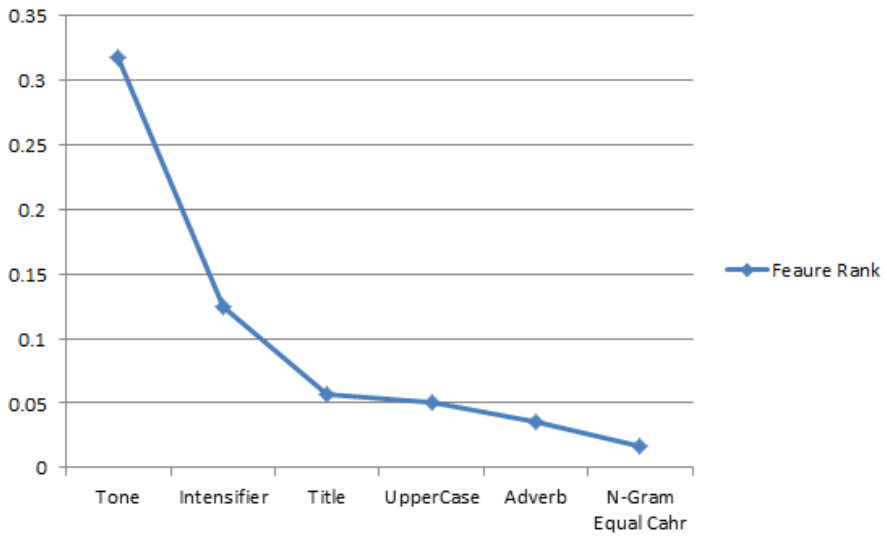

Figure 3.2: Features rank based on Information Gain

Also, fdf represent the frequency of the feature i in a document d. W is the function which results the rank of each feature obtained in feature importance analysis section. Also V is the variety of features used in the document. For example if two types of feature are used in a text V is 2. The assumption for considering v is that a document with 4 types of feature is more likely to be sarcastic than a document with just one type of feature. To distinguish between the documents in terms of their feature variety, parameter v needs to be considered in the formula.

3.1.6 Finding the Borderline

After scoring all the documents, it is time to find a borderline in order to classify the documents into groups of sarcastic and non-sarcastic. One option is to choose the first k documents which is used in [42]. However a better approach is to cluster the scores and choose the first k cluster as sarcastic. This way not only we can classify the documents into groups of sarcastic and non-sarcastic but also the relative sarcastic level of documents is considered by assigning them into different clusters based on their scores. For finding the threshold in order to set a border line for classifying the documents into sarcastic and non-sarcastic, I used K-Mean clustering algorithm with Euclidean distance.For this purpose I chose k as 6. Then since we consider this issue as binary classification and due to the fact that usually the sarcastic contents have the higher sarcastic degrees, I chose the first 5 clusters with the higher centroid to be sarcastic. Also the documents with the least score in

the cluster 5 is a good candidate to be set as a borderline. figure 3.3 depicts the centroid for cluster 1 to cluster 6 (the centroid in the figure are scaled by 1.5). Also this can gives us the ability to treat the problem of sarcasm

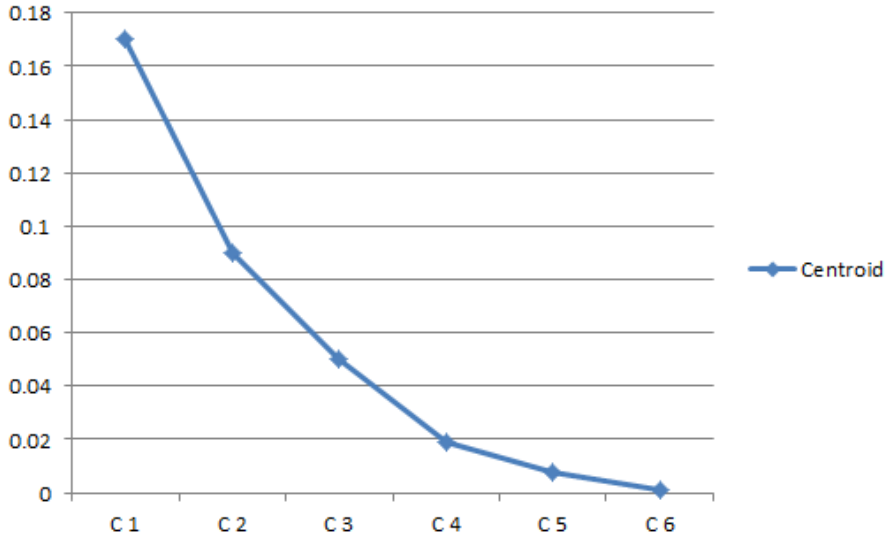

Figure 3.3: Cluster's Centroid

detection relative rather than binary classification. Of course, the first cluster contains the most sarcastic documents and the last one has the non-sarcastic documents. With having all the centroid we can set a different confidence level for border line based on our needs to distinguish sarcastic documents from non-sarcastic. Figure 3.4 and 3.5, shows the precision and recall of the sarcasm detection component based on different centroid as borderlines.

As we move the border line from centroid 1 to 6, it is obvious that precision drops or stays the same and recall increases. Also having different options for borderline can be very helpful in different scenarios. For example, in

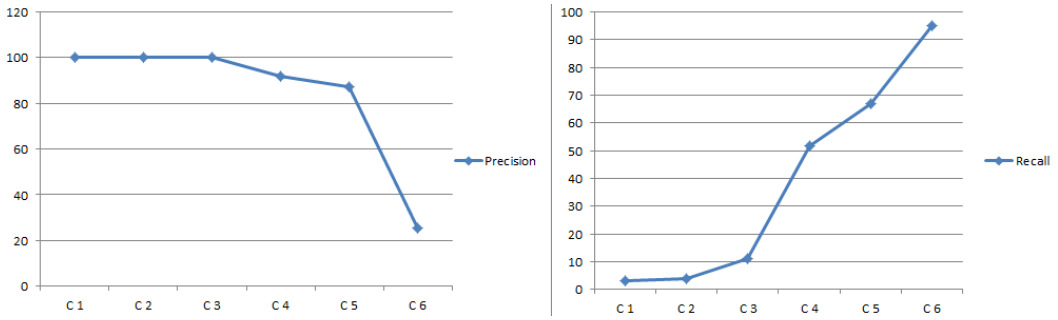

Figure 3.4: Precisions and recall comparison for different centroid

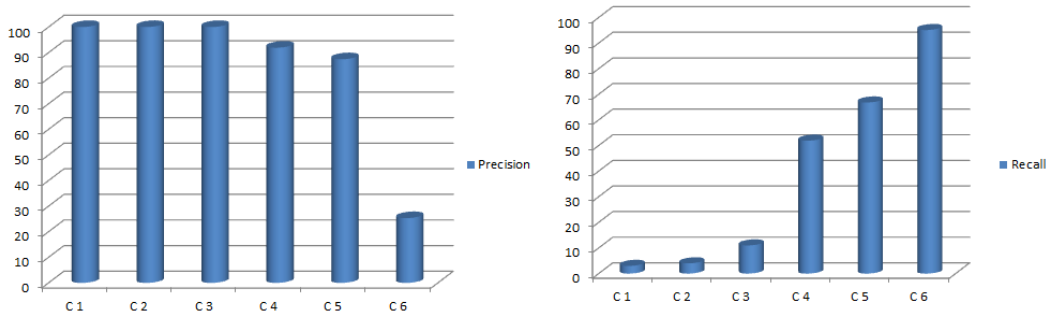

Figure 3.5: Precisions and recall comparison for different centroid

UI based systems when the precision is very important, we need to set a borderline with a very high confidence level. So in this case choosing the borderline as the centroid of cluster 1 or 2 is good. However, in this work we choose a document to be sarcastic if it falls into one of the clusters 1 to 5.

3.2 Sentence Level

In this section, we focus on the sentences related to the blogs we classified as sarcastic in the document level sarcasm detection component. As we discussed at the beginning of this chapter, considering sentences in isolation cannot be a good practice to detect sarcasm since we miss the context. However, if we know that a document is sarcastic then it is easier to detect sarcasm in sentence level. For sake an example consider the sentences in 3.4.

Table 3.4: Sample list of strong sarcastic sentences

number	Sentence
1	Youre like a brother to me You must be very stupid.
2	I doubt anyone would have considered that this nigga would have a bright future. haha
3	Is it disrespectful now to go hell for leather and lose yourself?
4	Well, building a house is not a freegan journey my friend.
5	And the Bible doesnt tell us if they were all female, but Im not saying anything. LOOOL
6	Well, yeah, it could be that his locks are special, the car door is faulty or youre a new girlfriend.
7	Autographs are too damn suspicious in this part of the world.25.
8	hahahaha..dont mind me, Im just a bitter moderator trying to play with your legs.LOL..Thanks for this.

If an individual take a look at these sentences they may consider them as sarcastic very quickly. However, not all the sentences contain clear evidence

of sarcasm like the sentences in table 3.4. For sake of comparison take a look at the sentences in table 3.5.

Table 3.5: Sample list of weak sarcastic sentences

number	Sentence
1	Sad part is that if youre not American, theres no cure for you.
2	So Im not gonna keep you here too long.Ebola is really serious guys.
3	Do I look like I am major in handling sounds?
4	If youre scared, just squeeze closer to me, hold me tighter.
5	What is really wrong with you people?
6	Are you even mad?
7	You smell really bad.

At the first glance, an individual might not consider these sentences sarcastic although they include some features such as questions. However, if we know that these sentences come from sarcastic contents then the likelihood of them being sarcastic increases. The framework for detecting sarcasm in sentence level is very similar to that of document level though we are using different feature set and two classifiers. After finding the sarcastic documents, we need to identify the sentences which are more likely to be sarcastic. Since we already know that the document is sarcastic relying on document level component, many of the features which are misleading in detecting sarcasm when we do not know if the document is sarcastic, can be effective. Figure 3.6, depicts the sentence level sarcasm detection component. As depicted in figure 3.6, the first subcomponent is preprocessing. In this subcomponent, all the punctuation except ! and ? are removed. In classification we treat

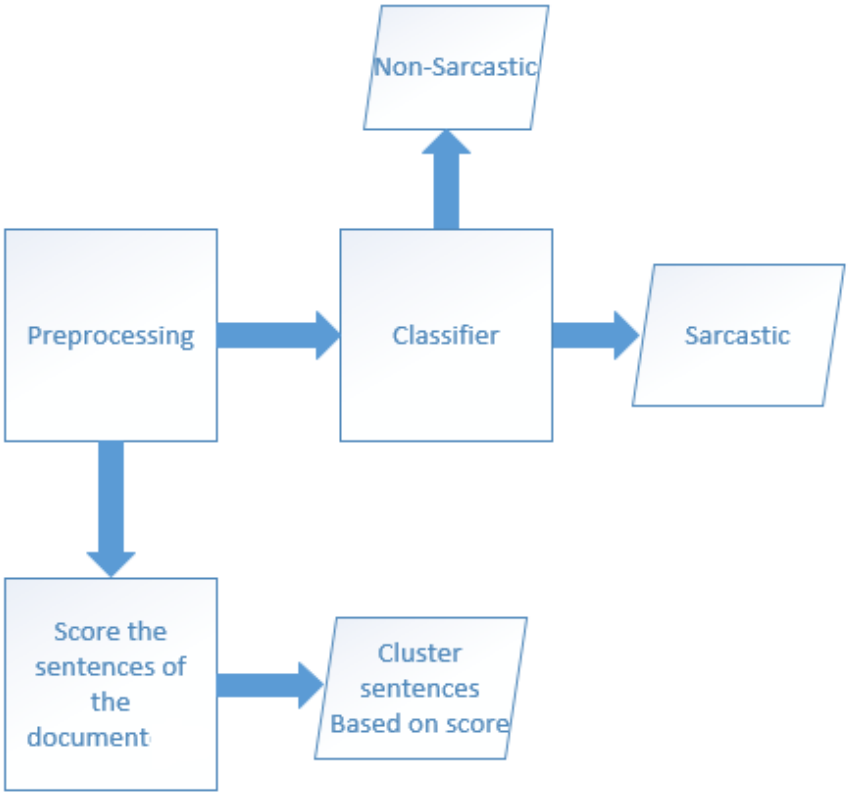

Figure 3.6: Sentence level sarcasm detection component

the problem in two ways: binary classification and clustering. For sake of binary classification, decision tree is used which classify sentences into groups of sarcastic and non-sarcastic. Also, as we will see in Chapter 5, scoring component showed a good result considering both precision and recall. So Scoring Component is also used in order to group sentences; to do so, after ranking sentences using scoring component, we used k-mean clustering to cluster the sentences of a document into different clusters.

3.2.1 Sentence Level Feature Set

The following lists all the features I tried in sentence level to detect sarcasm in this thesis:

- Tone: As is mentioned, tone is the key feature in the feature set since sarcasm is very relied on Tone. Also as is mentioned before, an individual might use sarcasm for different purpose such as acting funny (Sarcasm as wit) or showing anger. To show the anger in the text in social media the writer of the post extensively use explicit language. In this thesis, in order to capture the tone of the text, the list of tonic words and explicit language used in document level is applied. This feature is the normalized weight of the number of tonic phrases presented in a sentence.

- N-Gram Equal Character: This feature is very common in sarcasm detection task. By using this feature in a specific part of the text, the

writer shows the importance of that part to the readers. In order to capture this feature, first I generated all the 3 grams of all words in a sentence and all the words containing the 3 grams with equal characters is counted. Table 3.7 shows some example of sarcastic sentences using this feature. This feature is the normalized weigh of words containing N-Gram Equal character (N as 3 to 7) in a sentence.

Table 3.6: Example of sarcastic sentences using N-Gram Equal character

number	sentence
1	By the way, I did something thereOr should I share on how we become SOOOO BUSY this period?
2	what-it-doooo?
3	Whaaaa?
4	I love Simon SOOOOOO !!
5	Ive never been a CCCC.

- Neighborhood: Context has been proved to be very important in sarcasm detection. In this work, since we do not have the limitation exists in tweet texts (140 character), we have access to the sentence after and before the current sentence. This can give us a more flexibility to asses the essence of sarcasm by considering the sentences after and before. The assumption here is if a sentence after or before the current sentence contains any tonic words or explicit language words, the likelihood of sarcasm in the current sentence is higher. In order to capture this feature, we check the tone ratio of the sentence before or after, if the ratio is higher that 0, the neighborhood score will be increased by 1. Also in order to find the sentence before and after, after chuck-

ing documents into sentences, the order of each sentence is saved in the database. The score is divided by the total number of word in a sentence for normalization purpose.

- Part of Speech: In grammar, a part of speech is a linguistic category of words. However, for this section I used the following part of speech in the feature set:
  - Adverb: This feature is the normalized weight of adverbs in a sentence.
  - Adjective: This feature is the normalized weight of adjective in a sentence.

  In order to capture the part of speech, NLP tagger is used to get the tag form of the texts.

- Uppercase: This is the normalized weigh of upper case words in a sentence. A single sentence with more uppercase words is more likely to be sarcastic. For more explanation, sarcastic texts are usually intense and bloggers use different approaches such as using upper case words to intensify the content.

- Punctuation: This feature is the normalized weight of question marks and exclamation marks in a sentence. This can include true grammatically structured sentences or sentences using just question marks and exclamation marks.

- Comparison: Comparison can be considered as a way of indirectly attacking someone or mock a person. So it can be considered as a possible element of sarcasm generation in the text. In order to capture comparison in the text, I used the tagged text which was obtained by using NLP tagger. So by checking the part of speech of each word in the sentence, I used the following patterns to capture the common use of comparison in the text:
  - than+pronoun (such as than you).
  - more+ adjective
  - most+ adjective
  - adjective ending with er
  - adjective ending with est

  This feature is the normalized weight of comparison used in a sentence.

- Pronoun: Compared to situational irony which is usually about a situation, verbal irony and sarcasm target a person as a victim and pronoun is a common way for targeting another person in the text. So this feature is the number of pronouns in the text normalized by the total number of words in a sentence.

- Emoticons: This feature is common in informal texts and users usually use it to transmit their feeling to the readers. In order to capture this feature I collected a list conaining 102 emoticons widely used in

social media. This feature is the normalized number of emoticons in a sentence.

- Emotional Imbalance: This feature is the normalized sum of all positive words multiply by their polarity strength used in a sentence containing a negative words or tonic words. This feature is used in order to capture the unexpectedness in the text. In order to capture the positive words I used the dictionary of polarity words which is used in [24]. For more explanation in [24], a list of positive and negative words with their polarity strength is used. As is mentioned in [42], positive polarity of the text is not enough to decide if the text is sarcastic. However if the text which is positively worded contains very negative words such as "horrific" or explicit language or sarcastic tonic phrases, the text is emotionally imbalanced. As is mentioned earlier, emotional imbalance is categorized in the group of unexpectedness which is a key element in sarcasm generation [30]. So if a sentence contains very negative words or explicit language words, the number of positive words in the text increase the degree of sarcasm. It is noteworthy that for sake of normalization, the sum of positive words multiply by their strength is divided by the multiplication of total words and dictionary strength range.

3.2.2 Feature Importance Analysis

It is clear that not all of the above features mentioned are effective and we need to extract a subset of features which shows to have an impact on sarcasm detection. Similar to [2], for evaluating the worth of each attribute, Information Gain is used which evaluates the worth of an attribute by measuring the information gain with respect to the class. So higher information gain means better discriminative power for classification. The result from the evaluator shows that the following features performs better on the final results:

- Punctuation
- Tone
- Emotional Imbalance
- N-Gam Equal Character
- Neighborhood
- Pronoun

You can see the rank of each feature obtained from Weka Ranker in figure 3.7.

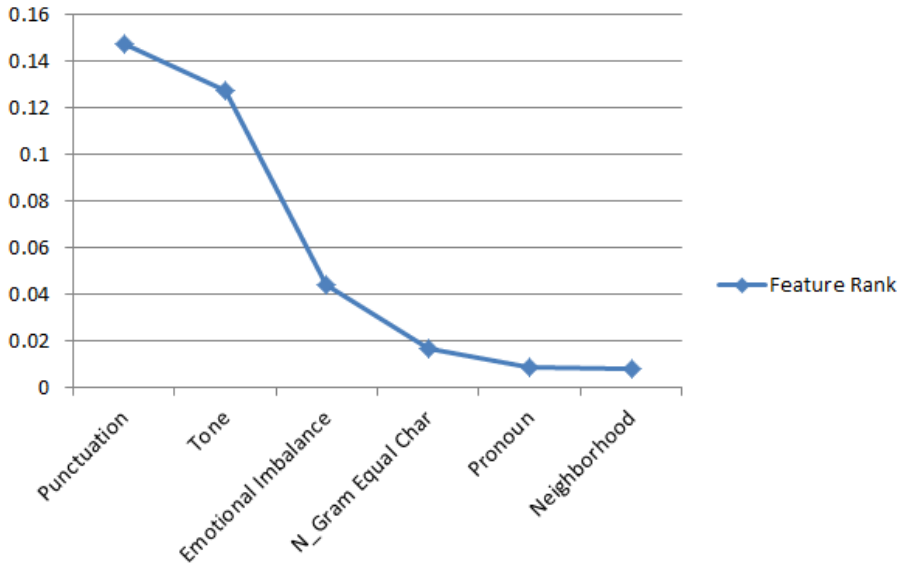

Figure 3.7: Features rank based on Information Gain

As is depicted in the figure 3.7, 6 types of features including Punctuation, Tone, Emotional Imbalance, N-Gam Equal Character, Neighborhood shows to have the highest rank. However, the rest of the features including Adverbs, Adjectives and comparative words shows to have either very low rank or not being effective at all. Also it is worth mentioning that Figure 3.7, shows the importance of punctuation including question marks and exclamation marks among all the other features used in the sentence level sarcasm detection.

3.2.3 Classification

3.2.3.1 Binary Classification

As will be discussed in Chapter 5, the tree based classifiers in sentence level showed to perform well. In this thesis, in order to classify sentences into groups of sarcastic and non-sarcastic in sentence level, decision tree as a classifier is used. To do so I used Weka java API library designed for java development. For more explanation, decision tree is an interesting classifier from different perspective such as increasing the performance and eliminating unnecessary computation, fixing the problem of high dimensionality [47, 29]. In terms of performance, contrary to many other conventional classifiers which each data is tested against all classes, tree classifiers is checked against only a ceratin subset of classes [47]. Also the problem of high dimensionality can be avoided by using Decision Tree which uses smaller number of features at each non-terminal node.

3.2.3.2 Relative Classification

One advantage of Scoring Component is that we can treat the problem of sarcasm detection relative rather than binary classification. To this end we end up with some sentences in different clusters based on their scores. To do this, first of all the sentences are scored. After scoring all the sentences, it is time to cluster them. To do so, I used K-Mean clustering algorithm with Euclidean distance. For this purpose I chose k as 7. Of course, the

first cluster contains the most sarcastic sentences and the last one has the non-sarcastic sentences. Figure 3.8 shows the centroid for the 7 clusters (the centroid in the figure are scaled by 1.5).

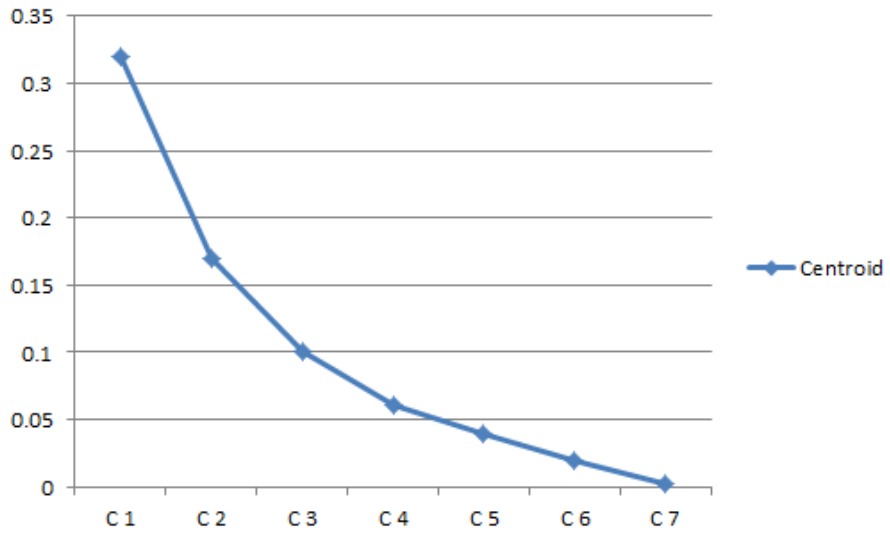

Figure 3.8: Cluster's Centroid

As is shown in Figure 3.8, the centroid 1 is the greatest and centroid 7 is the least among all the other centroid. With having different clusters we can group the sentences with varying degree of sarcasm rather than just classifying them into groups of sarcastic and non-sarcastic. Figure 3.9 and 3.10 compare the precision and recall of the scoring component based on different centroid as borderlines.

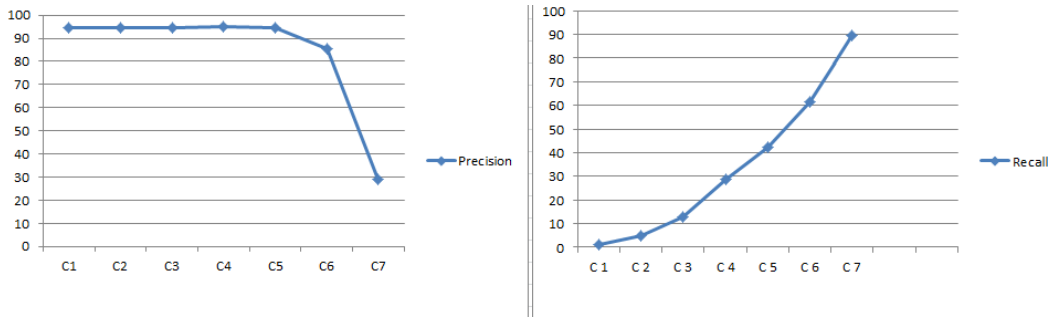

Figure 3.9: Precision and recall comparison based on different centroid

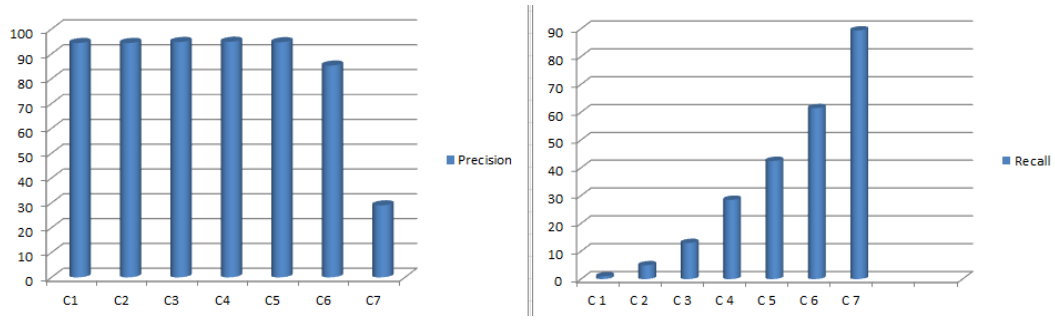

Figure 3.10: Precision and recall comparison based on different centroid

As is clear, as we move the border line from centroid 1 to 7, it is obvious that precision drops or stays the same and recall increases. The importance of having different clusters of sarcastic sentences is more clear when we are dealing with scenarios in which the precision is very important; so choosing the sentences from cluster 1 is the safest option. Although in this work the focus is on classifying the sentences into group of sarcastic and non-sarcastic using decision tree, the framework is able to assign sentences in a document into different clusters based on their scores using Scoring Component and

K-mean clustering. This can be one advantage of using scoring component over the other classifiers.

Chapter 4

Short Text Sarcasm Detection

Sarcasm detection in short text and more specifically twitter is different from what is done in long text. The very obvious difference is the way posts are labeled. In long text we do not have any choice to label the documents rather than doing it manually. However, this is much easier when it comes to twitter since usually Twitter users use #sarcasm or similar hash tags to show that the post is sarcastic although there are many cases in which the tweet is sarcastic and it is not tagged with #sarcasm. However, the latter can be a problem as well; there are many cases in which users use #sarcasm to show that they mean to be sarcastic but there is not enough evidence of sarcasm in the text. For more explanation, if the #sarcasm is removed from the text, it can be very difficult even for human being to detect it. Also in contrary to short text, the ratio of sarcastic blog posts are very low compared to Twitter. Figure 4.1 shows the sarcasm detection component

used in this thesis for micro blog texts(Twitter). In Figure 4.1, crawler is the

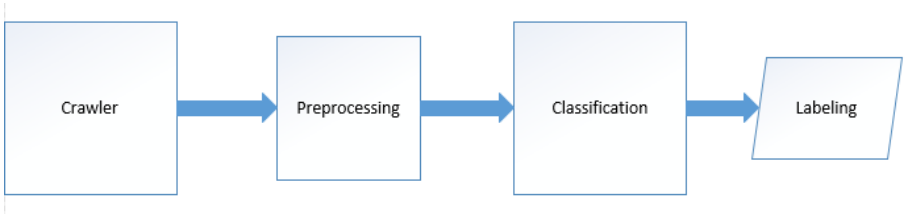

Figure 4.1: Short text sarcasm detection component

first subcomponent. After crawling data, in preprocessing, all punctuations except ! and emoticons are removed. Also all the non-alphabetic characters such as numbers, extra space, asterisk are removed. For classification which is the heart of this component, decision tree is used. First, the model is built using the set of tweets stored in the database. After the model is built, every new tweet is analyzed and feed into the classifier. The result is a tweet text with a label sarcastic or non-sarcastic.

4.1 Crawler

Data is the first and most fundamental part of all machine learning and NLP tasks. Twitter offers a streaming API which is a method of accessing to the public timeline of the users. In this work, I used one of the best streaming API which is commonly used to crawl twitter called Twitter4j. This API is the free API which is written in 100% pure java. Also this framework works with any platform written in java version 5 or later. Since there is no

dependency needed for adding this API to the code, many developers choose this API.

Using twitter4J API gives a lot of options to the programmers including tweet related and author related information. Information such as number of retweet counts, tweet texts, tweet longitude and latitude,Author Id and tweet id. Also some author related information such as favourite count, number of followers, number of followings, author screen name and etc. can be extracted.

A limitation of using this streaming API is that we can not retrieve more than 100 tweets in each request. Also multiple request to the API consequently is resulted in authentication failure and being blocked for a while. So a work around which is used in this thesis is to have a loop requesting tweets with specific hashtags with a waiting time of 15 seconds after each round of the loop. Each request to the Twitter server returns a json file containing 100 tweet information. After getting the Json file, this file is processed by the twiter4J API and all the information is saved to the database.

4.2 Data set

Twitter as a very common microblog, is a rich source of data. Compared to blogs, Twitter is very rich in terms of sarcastic posts. Also another advantage of twitter is the feature called hashtag. Users can tag their posts with a specific hashtag to show their purpose. In this work, in order to collect

sarcastic tweets, I used Twitter4J to search for tweets which contains specific hashtags such as #sarcasm. This way I make sure that all the tweets which contains #sarcasm are sarcastic since it is up to the author to decide what is sarcastic or not. Also in order to make sure that all the posts extracted with other hashags rather than #sarcasm are non-sarcastic, I manually checked them and those tweets which were not clear if they were sarcastic, were ignored. In total 33315 tweets were collected in which 6212 of them are sarcastic.

4.3 Feature Set

Among all social media, twitter is one of the most advanced API in terms the data it provides to the user. Along with the tweet itself which is a plain text, many other information related to the tweets or the author is provided. This makes up for the limitation of 140 character which limits the power of the textual features. However still textual features proved to be the most important features among all the other features. The following shows the list of features tried in this thesis. However, some of them showed to be effective in sarcasm detection.

- Tone Ratio: This is the total number of tonic words normalized by the total number of words in the tweet text. As is mentioned before, tone is the key feature in the feature set since sarcasm is very relied on Tone. Also, an individual might use sarcasm for different purpose

such as acting funny (Sarcasm as wit) or showing anger. To show the anger in the text in social media the writer of the post may use explicit language. This way they can transmit the nastiness through the text. Also there are many cases in which users use some common tonic words or phrases to act funny or avoid answering or criticise directly. In order to capture this feature in short text, the same list of tonic words used in long text is applied.

- Comparison Ratio: This is the normalized weight of comparison in the text.
- Emoticons Ratio: This feature is the total number of emoticons in the text divided by the total number of tweet words.
- Part of speech:
  - Adverb Ratio: This feature accounts for the total number of adverbs divided by the total number of words.
  - Adjective Ratio: This is the normalized number of adjective in the text.
  - Verb Ratio: This feature is calculated by dividing the total number of verbs by the total number of words in the text.
  - Noun Ratio: This feature is measured by dividing the total number of nouns by the total number of words.

- Intensifier Ratio: To measure this feature, the total number of adverb intensifiers is divided by the total number of words in the text. A tweet text with adverb intensifiers is more intense than the other texts and the intensity is used in the text to generate the feeling of exaggeration. However considering the tweet length limitation, this feature is not as effective as it is in long texts.

- Exclamation Mark Ratio: This feature as a representative of surprising factors in the text, is another commonly used feature. Usually Twitter users use exclamation marks a lot to show exaggeration in a text. Also this feature can be very handy to generate sarcasm considering the twitter limitation (140 characters). This feature is the normalized weight of exclamation marks in the text.

- N-Gram Equal Character Ratio: This feature is the total number of words containing 3 or more equal character normalized by the total number of words in the text.

- Link Ratio: Users use links in order to give a reference to their audience in order to prove their points. Although there are some sarcastic cases which uses the links, considering the low rate of sarcastic tweets compared to non-sarcastic one, link ratio in non-sarcastic tweets is higher. This feature is the number of links normalized by the total number of words in a tweet text.

- Uppercase Ratio: Upper case is usually used in the text specially in

microblogs (due to space limitation) to point an important issue. This way the author tries to catch the audiences eyes. This feature is the total number of words with all uppercase letters divided by the total number of words.

- Pronoun Ratio: Similar to Long Text Sarcasm Detection in Chapter 3, usually the target of sarcasm is a person. Pronoun is very common to point a person in order to generate sarcasm. This feature is the number of pronouns normalized by the total number of words in the tweet.

- Question Ratio: Question is another way to indirectly complain about something or using wit in a sentence. Although this can be very tricky since there are many cases use questions and there is no evidence of sarcasm, when this feature is used with other sarcastic features it enhances the likelihood of being sarcastic. This feature is the normalized weight of question mark in a tweet text.

- Space Efficiency: Due to the nature of twitter, space is very impotent. Space efficiency shows the skill of the author. Usually authors with a good command of English who uses twitter more often, writes shorter but at the same time they can transmit their meaning clearly. The assumption is that usually users who writes efficient is more likely to use sarcasm. This feature is the number all characters divided by the total number of characters.

4.4 Feature Importance Analysis

In order to find the effective features in the feature set, similar to long text sarcasm detection section, Information Gain used in [2], is applied which evaluates the worth of an attribute by measuring the information gain with respect to the class. So higher information gain means better discriminative power for classification. The result of evaluation shows that the following features performs better on the final results:

- Link
- Adverb
- Tone
- Pronoun
- Exclamation Mark
- Uppercase
- N-Gram Equal Character
- Comparison
- Emoticons

Also 4.2, shows the importance of each feature based on their rank obtained from Weka Ranker.

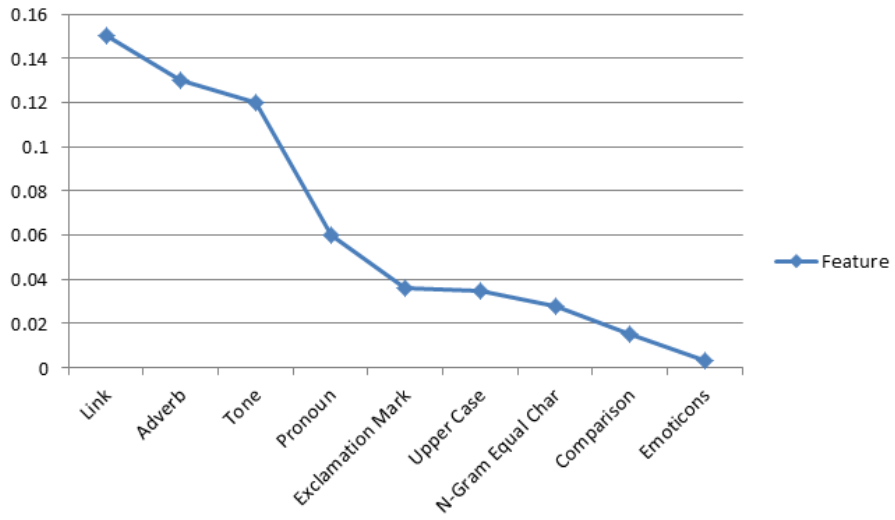

Figure 4.2: Feature importance analysis

As is shown in Figure 4.2, link is the most effective feature. Authors usually use link when they want to add a reference for showing the validity of what is said. Also adverb is shown to be the second best feature followed by Tone. In contrary to the long text in which tone proved to be the most effective feature, in short text it takes the third position. This proves that although sarcasm is heavily relied on tone, the nature of tweet texts and its complexity plays an important role in sarcasm detection. Also pronoun is the fourth important feature as is depicted. For more explanation, sarcasm targets another person as a victim in contrary to situational irony which target a situation and the best way for this purpose is to use pronoun. Furthermore, although this feature is in the list of top 4 best features, there is a noticeable gap between link as the best feature and pronoun. Also upper case, n-gram

equal character, Comparison and Emoticons showed to be effective though with a lower impact compared to the first 4 features.

4.5 Classifier

As will be discussed in Chapter 5, the performance of tree based classifiers in short texts showed to be better than the other classifiers including scoring component. This relies on the nature of tweet texts. Although using tonic words and nastiness is very common in tweet texts, it is not as widely used as long texts. Also there are many tweet texts in which even human beings may not be able to detect the sign of sarcasm. This will make the task of sarcasm detection very difficult and it is very dependant on the skills and interpretation of the tweet users about sarcasm. In this thesis, in order to detect sarcasm in short text, decision tree as a classifier is used. To do so I used Weka java API library designed for java development.

Chapter 5

Experimental Results and Discussion

5.1 Long Text Document Level

In the long text sarcasm detection in document level as is discussed in Chapter 3, scoring component is used to score each document based on the features defined. By using this classifier, an acceptable precision of 75.7% is obtained when all the features are used. Table 5.1 shows the summary of precision, recall using different classifiers.

As is clear from Table 5.1, Scoring Component performs better than all the other classifiers except tree families. Also in case of AdaBoost, SVM and NB, The precision of lower than 70% cannot be an acceptable result compared to tree families and Scoring Component. Figure 5.1 depicts the precision's

Table 5.1: Classifier's Comparison in Document Level

	Precision	Recall
Scoring Component	75.7	80.3
Random Forest	79.2	69.7
Bayes Net	72.2	72.4
Rep Tree	79.9	62.7
Decision Tree	82.9	64.2
NBtree	82.1	64.5
NB	49.5	39.5
SVM	65	15.3
Ada Boost	69.3	70.1

results obtained from different classifiers.

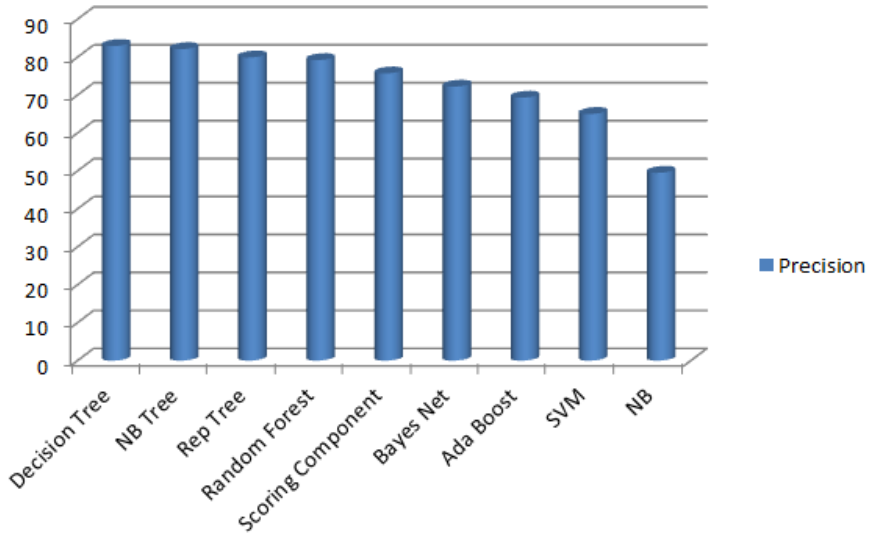

Figure 5.1: Document Level Precision

However, considering just the precision of the result is not enough since the

aim is to discover as much sarcastic documents as possible and at the same time the true positive rate is very important. So considering just the precision is not enough to make decision. Figure 5.2 depicts this fact.

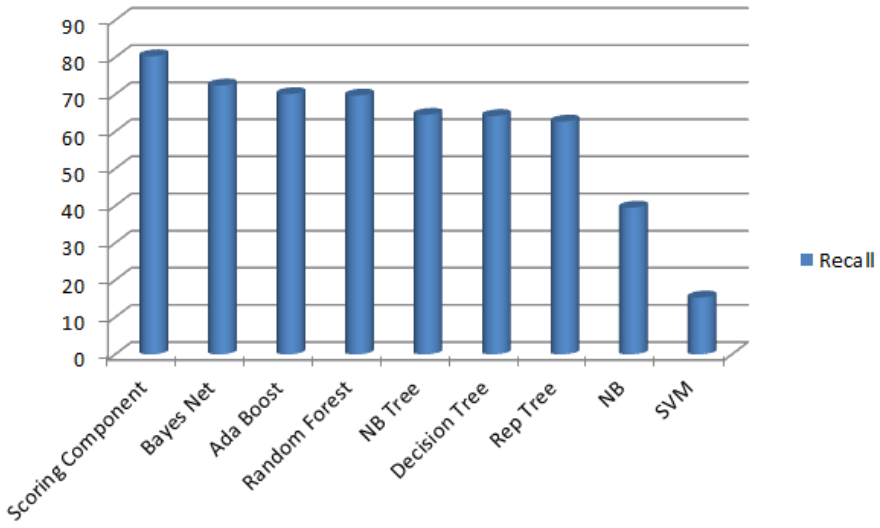

Figure 5.2: Document Level Recall

Figure 5.2 shows that the scoring component has the best recall among all the other classifiers. Also Bayes Net and AdaBoost classifiers show to have a good result in terms of recall though still there is a considerable gap compared to Scoring Component. So considering both precision and recall, it is clear that Scoring Component has the best performance among all the other classifiers. Also using Scoring Component gives us this ability to classify documents into different clusters based on their degrees. This is the noticeable advantage of using Scoring Component over all the other classifiers.

5.2 Long Text Sentence Level

As is discussed in Chapter 3, Decision Tree is used for sentence level binary classification in long texts. After finding the sarcastic documents, in order to find the sarcastic sentences in a document we need to analyze the documents in sentence level. According to the comparison made in order to compare Decision Tree, Scoring component and other classifiers, Decision Tree showed to be the best considering Recall and the second top classifier in terms of Precision. Table 5.2 depict the Precision and Recall of these classifiers:

Table 5.2: Classifier's Comparison in sentence Level

	Precision	Recall
Scoring Component	79.5	70.8
NBTree	90.3	65.5
Decision Tree	85.5	72.1
NB	76.9	26.4
SVM	80.9	39.7

In Table 5.2 it is obvious that Tree based family performs better than the other classifiers in general; also all classifiers perform better than Scoring Component considering precision except Naive Bayes. Figure 5.3 compare the precision of the different classifiers.

Classifiers such as Tree and NBTree are the top 2 classifiers. However, as is

mentioned earlier, precision is not enough to decide about the performance of the classifiers. For example, NBTree which shows to be very effective in terms of the precision, has a low true positive rate(recall) of 65.5. Figure 5.4 compare the classifiers considering the recall.

Figure 5.4 shows that Tree is the best among all the other classifiers considering recall. Also Scoring Component is the second best classifier compared to the other classifier.

5.3 Short Text Analysis

According to the comparison made between different classifiers including scoring component, Decision tree has the best performance in short text

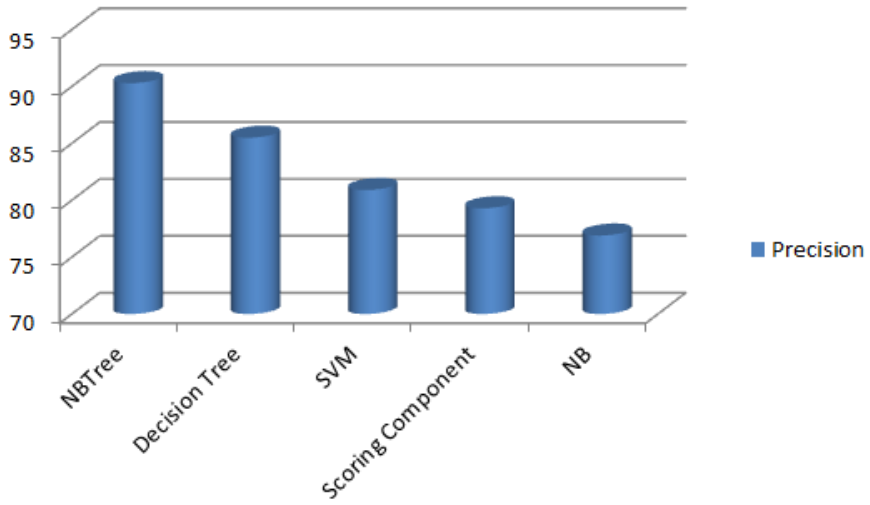

Figure 5.3: Sentence Level Precision

sarcasm detection. Table 5.3 shows the results using different classifiers.

Table 5.3: Classifier's Comparison in short text

	Precision	Recall
Rep Tree	83.4	82.9
Scoring Component	50.3	50.6
Random Forest	84.3	80.7
Bayes Net	60	83.1
Ada Boost	70.6	80.3
Decision Tree	86.6	85.6
NBtree	85.5	81.9
NB	70	48.6
SVM	81	56.2

From the Table 5.3, it is obvious that Tree families outperform all the other

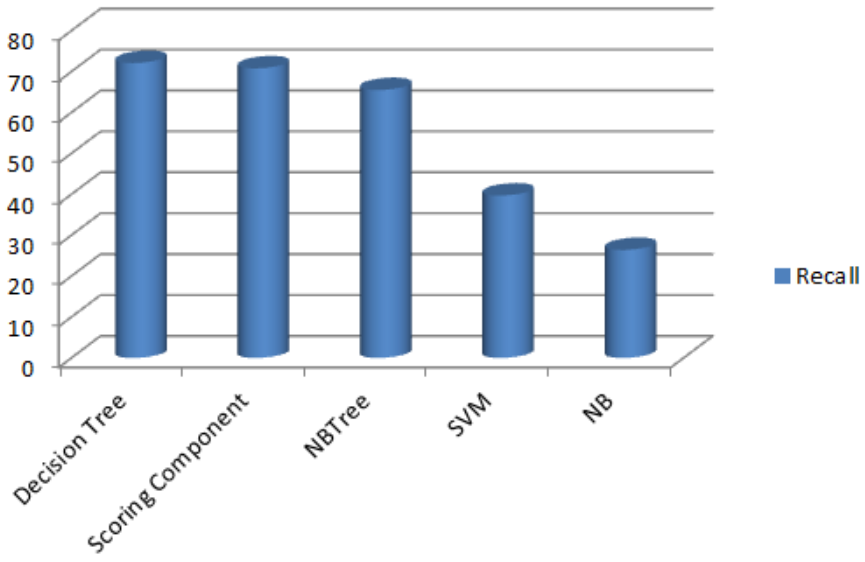

Figure 5.4: Sentence Level Recall

classifiers. Also Figure 5.5 shows that among all the classifiers, decision tree is in the top of the list considering precision.

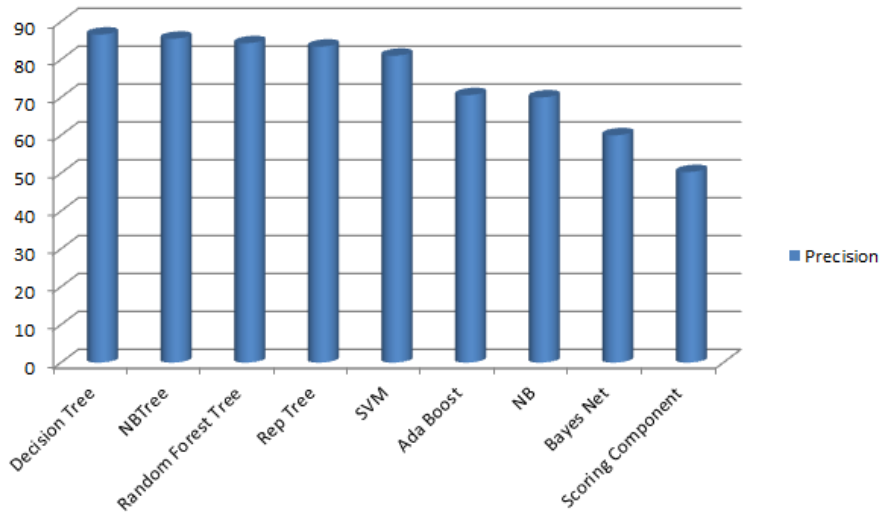

Figure 5.5: Short text precision comparison

In Figure 5.5 the top 4 classifiers are related to tree families. Also the same is true for recall as is depicted in Figure 5.6.

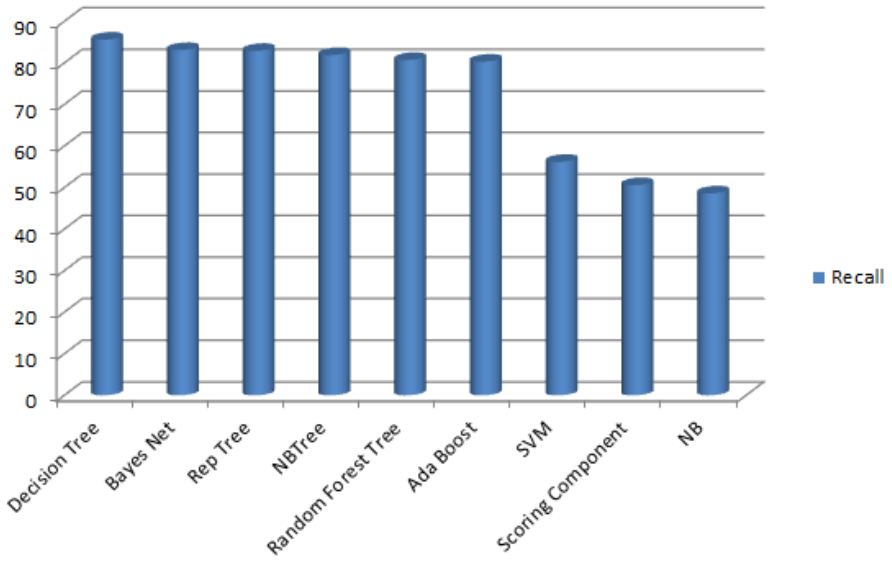

Figure 5.6: Short text recall comparison

Figure 5.6 shows that 4 out 5 top classifiers belong to tree family. Also, scoring component showed to have a poor result. The reason lies behind the nature of Twitter. Twitter as a very popular microblog has some unique features. In twitter authors are able to tag their posts to transmit their purpose easier to the audiences. This clever idea is to make up for the space limitation.However sometimes the misuse of this feature may cause trouble in the field of data mining. Table 5.4 shows some tweets which are labeled with #sarcasm but the evidence of sarcasm is even difficult for human being to detect.

Table 5.4: List of weak sarcastic tweets

number	sentence
1	A Lesson In #SocialMedia #Sarcasm via @LinkedIn https://t.co/3bAQERImEj
2	@OfficialGamerJB when's the draw ??#sarcasm
3	@SarahPalinUSA Future David Attenborough here. #sarcasm
4	Make a #hacker 's job easier by outlawing #encryption #waytogo #sarcasm Sounds legit https://t.co/UMh2TS7qRf
5	Compromise is the devil's work....and those who support this concept are evil.... #Sarcasm #ImWithHer #FeelTheBern #Sanctimonious
6	Pls, Jim. ALL Prince, ALL today. #sarcasm @JimSharpe
7	@guypbenson that'll win votes. #sarcasm
8	RT @lef106: Make soccer great again. #Sarcasm #SavetheEarthInFourWords
9	Feel bad for Martinez ... #sarcasm

As is clear from table 5.4, contrary to blogs which authors try different ways to transmit sarcasm, authors in twitter sometimes rely on the hashtags instead of clearing the sarcasm in texts. Another reason for this poor result could be the lack of manually annotated Twitter data. One might argue that it depends to the author to decide what is sarcastic, though the fact remains that it would be vey difficult and sometimes impossible to understand the sarcasm in many cases if the sarcasm hashtag were to be removed from the tweet texts.

5.3.1 Difficulty of Sacasm Detection in Short Text

As discussed before, sarcasm detection is a challenging task even for human being; as is shown in short text, this is even more obvious due to the nature of Twitter. In twitter, the space limitation and also some of the Twitter

related phenomenon such as hashtags are considered to be the key factors on the result of sarcasm detection tasks. Also this should be considered that access to context is very limited in microblogs such as Twitter.

In order to show the difficulty of sarcasm detection especially in short text, I conducted a survey. I chose 4 English speaking annotators who are born and lived in North America for their whole life. This way, I made sure that they have a good command of English and also they are familiar with the concept of Sarcasm. Furthermore, since Sarcasm is a common language feature in North America and also it is very dependant on the culture, choosing annotators all from North America reduces the human error rate in labeling the texts. I chose 18 tweets which are less obvious to be sarcastic. Also a metric called confidence degree of a text is defined which shows the likelihood of being sarcastic for the texts based on votes of annotators. Of course the greater the confidence degree is, the more sarcastic the text would be. Table 5.5 shows the tweets used for this purpose. The experiment showed very interesting results which is depicted in Table 5.6.

As is shown in Table 5.6, from all the tweets which are tagged by #sarcasm, only 3 of them have the confidence degree of 100%. This means that only in 3 out of 18 tweets, all annotators agree that tweets are sarcastic; this shows that only in 16% of tweets, all annotators are able to get the author point. On the other hand, only one tweet has a confidence degree of 75% which means that 3 out of 4 annotators agree with the author. However, the rest of the tweets have a confidence degree of 50% or less which shows that in

Table 5.5: List of Sarcastic Tweets Used in the Survey

number	Word
1	Great job governing @JustinTrudeau. https://t.co/kQAoORvTYb.
2	Compromise is the devil's work....and those who support this concept are evil
3	Let's not let down society.Let's let down ourselves instead
4	@IndianRegista @ESPNInsider Buffon competes in Serie B while De Gea is one win away from the Champions League final man
5	Make soccer great again.
6	It wasn't Patrick Kane's skill that won the @NHLBlackhawks the game, Toews leadership guided the puck in.
7	Did the #Cubs make some changes to their clubhouse or something? If so I can't find any stories about it
8	I love being able to not smell or taste anything!
9	@wheelsee watch your profanity!
10	RT @AnilaButtPTI: Nawaz Sharif buying "EXTRA TIME" to stay in power from #Rolex Rather than
11	I could be studying....or I could be at Grand Prix week. Finals week sounds a lot better!
12	Cannot wait to do my favorite thing this morning: Go through Customs!
13	Monday night @ SEATTLE?! sweet.
14	WHOOP WHOOP apparently I am owed $24718.11 wow! PS not my gmail, I don't own that
15	@Twins way to manufacture run!
16	Let's Go #Reds we can do this! Let's Rally and win this in the bottom of the ninth
17	WHO EVER DISAGREES WITH ME ARE PEDOPHILES!1!!
18	Congrats @jakearrieta! Glad things are working out for you and the Cubs!

Table 5.6: Results of the Survey

number	Annotator 1	Annotator 2	Annotator 3	Annotator 4	Confidence degree
1	Non-sarcasic	Sarcasic	Sarcasic	Non-sarcasic	50%
2	Non-sarcasic	Non-sarcasic	Sarcasic	Sarcasic	50%
3	Sarcasic	Sarcasic	Sarcasic	Sarcasic	100%
4	Sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	25%
5	Non-sarcasic	Non-sarcasic	Sarcasic	Sarcasic	50%
6	Sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	25%
7	Non-sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	0%
8	Sarcasic	sarcasic	sarcasic	sarcasic	100%
9	Sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	25%
10	Non-sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	0%
11	Sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	25%
12	Sarcasic	Sarcasic	Sarcasic	Sarcasic	100%
13	Non-sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	0%
14	Sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	25%
15	Non-sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	0%
16	Non-sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	0%
17	Sarcasic	Sarcasic	Sarcasic	Non-sarcasic	75%
18	Non-sarcasic	Non-sarcasic	Non-sarcasic	Non-sarcasic	0%

most of the cases most annotators can not get the authors point. This will be more obvious, considering that in 11 tweets out of 18(61%), less than half of the annotators label the tweets as sarcastic.

5.4 Conclusion Remarks

In this Chapter the experimental study of the proposed framework was presented. Two different data sets were used, one contained blog posts and another consisted tweet texts. In long text component, two different classifiers were used in document level sarcasm detection and sentence level sarcasm detection. According to the comparison made between different classifiers considering document level, scoring component proved to be the most effective classifier. For doing this comparison, two metrics were considered including precision and recall. Precision of 75.7 % and recall of 80.3 % are an outstanding result comparing to the other classifiers. Also in order to detect the sentences which were more likely to be sarcastic Tree classier was used. Considering both recall and precision, tree families showed to have a better performance in sentence level sarcasm detection. A precision of 85.5 % and recall 72.1% for tree classifier is a proof for the latter statement.
Furthermore, the performance of short text sarcasm detection component was studied and it was shown that the tree classifier was in the top of the list of best classifiers considering both precision and recall. Similar to sentence level sarcasm detection component in long text, tree based families showed

to have a better performance and Scoring Component did not perform well compared to the other classifiers which was due to the specific features of Twitter such as space limitation and hashtags.

The final experiment was devoted to the show the difficulty of sarcasm detection task in Twitter. To do so, a survey was conducted using 4 native annotators. Annotators were asked to label the tweets after removing #sarcasm. An interesting results were obtained: 33% of the sarcastic tweets were labeled as non-sarcastic by all the annotators which was totally contrary to what was expected. Also 61 % of the sarcastic tweets were labeled as non-sarcastic considering majority votes. This shows the fact that Sarcasm is very relied on the perception and interpretation of the authors. So one might find a text sarcastic and another might consider it as non-sarcastic.

Chapter 6

Conclusions and Future Work

After introducing the problem, giving a review of the related literature, proposing the solution, and presenting the evaluation, this is time to conclude the thesis. This chapter of the thesis realizes this task in two sections. In the first section, a review of the framework is done and the second section enumerates the possible points of extension for interested researchers and petitioners in the field.

6.1 Conclusions

Automatic detection of sarcasm is crucial not only for sentiment analysis and issue discovery, but also for many other natural language processing tasks such as question answering and textual entailment. If sarcasm detection is complex for humans, its automatic detection becomes even more challenging

with natural language processing and text mining.

In this thesis, a solution to the problem of sarcasm detection in both long text and short text were proposed. The solution was in the form of a framework. The framework consists of two components including long text sarcasm detection and short text sarcasm detection. In each framework, varied range of feature used based on the nature of the social medium environment and the nature of the text. Also the two sets of classifiers were used including Scoring Component and Decision Tree. Scoring Component was used in order to classify the documents into groups sarcastic and non-sarcastic; also for classification in sentence level and short text decision tree was applied. these classifiers showed to be the best among all the other classifiers considering both recall and precision.

In terms of data, two sets of data sets were used. For short text,Twitter as a rich source of labeled data were used. In order to crawl tweets, Twiter4j API which is a very common streaming API is applied. Also for long text, WordPress API is utilized to crawl blog posts as a representative of long texts.

6.2 Future Work

In the future, expansion of the feature sets is recommended to increase the accuracy and viability of the framework. These could include context, which would identify the cultural, blog post comment/feedback, personal profiles

information, environmental, external that provide critical data to assess sarcasm in posts. This framework also provides a good starting point in future research in differentiating irony from sarcasm since Scoring Component gives us this ability to cluster posts into different groups based on their degrees. However, more linguistic study may be needed to find a formula to distinguish these two from each other; although because of the limitations in texts in some researches including this research they are considered the same. Also bag of words approach is a reliable and common way in dealing with many natural language processing tasks. However, the static nature of bag of word approach may need some improvements and maintenance of the lists by keeping it up to date due to the ever changing nature of social media posts specially microblogs. So the latter may be expensive when dealing with big data since it is 100% manual. A possible solution in case of Twitter, for enhancing the lists and keeping it up to date may be to instantly crawl sarcastic posts and find the most common 1,2 and 3 grams. However, the accuracy of the result is a key element to consider while using more dynamic approaches rather than bag of words.

Bibliography

[1] David Bamman and Noah A Smith, *Contextualized sarcasm detection on twitter*, Ninth International AAAI Conference on Web and Social Media, 2015, pp. 574–577.

[2] Francesco Barbieri, Horacio Saggion, and Francesco Ronzano, *Modelling sarcasm in twitter, a novel approach*, ACL 2014 (2014), 50–58.

[3] Daniel M Bikel, Richard Schwartz, and Ralph M Weischedel, *An algorithm that learns what's in a name*, Machine learning **34** (1999), no. 1-3, 211–231.

[4] Mondher Bouazizi and Tomoaki Ohtsuki, *Opinion mining in twitter how to make use of sarcasm to enhance sentiment analysis*, Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, ACM, 2015, pp. 1594–1597.

[5] ———, *Sarcasm detection in twitter: All your products are incredibly amazing!!!-are they really?*, 2015 IEEE Global Communications Conference (GLOBECOM), IEEE, 2015, pp. 1–6.

[6] ———, *Sarcasm detection in twitter*, Institute of Electrical and Electronics Engineers Inc., 2016.

[7] Derek Bousfield, *never a truer word said in jest: A pragmastylistic analysis of impoliteness as banter in henry iv, part i*, Contemporary Stylistics (2007), 195–208.

[8] Penelope Brown and Stephen C Levinson, *Politeness: Some universals in language usage*, vol. 4, Cambridge university press, 1987.

[9] Paula Carvalho, Luís Sarmento, Mário J Silva, and Eugénio De Oliveira, *Clues for detecting irony in user-generated contents*, Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, ACM, 2009, pp. 53–56.

[10] Shelley Channon, Asa Pellijeff, and Andrea Rule, *Social cognition after head injury: Sarcasm and theory of mind*, Brain and Language **93** (2005), no. 2, 123–134.

[11] Dmitry Davidov, Oren Tsur, and Ari Rappoport, *Semi-supervised recognition of sarcastic sentences in twitter and amazon*, Proceedings of the Fourteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, 2010, pp. 107–116.

[12] Shelly Dews and Ellen Winner, *Muting the meaning a social function of irony*, Metaphor and Symbol **10** (1995), no. 1, 3–19.

[13] Richard O Duda, Peter E Hart, et al., *Pattern classification and scene analysis*, vol. 3, Wiley New York, 1973.

[14] Elisabetta Fersini, Federico Alberto Pozzi, and Enza Messina, *Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers*, Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on, IEEE, 2015, pp. 1–8.

[15] Erik Forslid and Niklas Wikén, *Automatic irony-and sarcasm detection in social media*, (2015).

[16] Rachel Giora, *On irony and negation*, Discourse processes **19** (1995), no. 2, 239–264.

[17] Rachel Giora, Noga Balaban, Ofer Fein, and Inbar Alkabets, *Negation as positivity in disguise*, Figurative language comprehension: Social and cultural influences (2005), 233–258.

[18] Laurence Horn and Yasuhiko Kato, *Introduction: negation and polarity at the millennium*, Negation and Polarity. Syntactic and Semantic Perspectives. Oxford University Press, Oxford (2000), 1–19.

[19] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß, *A brief survey of text mining.*, Ldv Forum, vol. 20, 2005, pp. 19–62.

[20] Thorsten Joachims, *Text categorization with support vector machines: Learning with many relevant features*, Springer, 1998.

[21] W Oller John Jr, *Scoring methods and difficulty levels for cloze tests of proficiency in english as a second language*, The Modern Language Journal **56** (1972), no. 3, 151–158.

[22] Julia Jorgensen, *The functions of sarcastic irony in speech*, Journal of Pragmatics **26** (1996), no. 5, 613–634.

[23] Raquel Justo, Thomas Corcoran, Stephanie M Lukin, Marilyn Walker, and M Inés Torres, *Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web*, Knowledge-Based Systems **69** (2014), 124–133.

[24] Mostafa Karamibekr and Ali A Ghorbani, *Sentence subjectivity analysis in social domains*, Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01, IEEE Computer Society, 2013, pp. 268–275.

[25] JP Kincaid, RP Fishburne, RL Rogers, and BS Chissom, *Derivation of new readability formulas*, Tech. report, Technical report, TN: Naval Technical Training, US Naval Air Station, Memphis, TN, 1975.

[26] Roger J Kreuz and Gina M Caucci, *Lexical influences on the perception of sarcasm*, Proceedings of the Workshop on computational approaches to Figurative Language, Association for Computational Linguistics, 2007, pp. 1–4.

[27] Roger J Kreuz and Sam Glucksberg, *How to be sarcastic: The echoic reminder theory of verbal irony.*, Journal of Experimental Psychology: General **118** (1989), no. 4, 374–386.

[28] Roger J Kreuz and Richard M Roberts, *Two cues for verbal irony: Hyperbole and the ironic tone of voice*, Metaphor and symbol **10** (1995), no. 1, 21–31.

[29] GH Landeweerd, T Timmers, Edzard S Gelsema, M Bins, and MR Halie, *Binary tree versus single level tree classification of white blood cells*, Pattern Recognition **16** (1983), no. 6, 571–577.

[30] Joan Lucariello, *Situational irony: A concept of events gone away*, Irony in language and thought (2007), 467–498.

[31] Diana Maynard and Mark A Greenwood, *Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis*, Proceedings of LREC, 2014.

[32] Skye McDonald, *Exploring the process of inference generation in sarcasm: A review of normal and clinical studies*, Brain and language **68** (1999), no. 3, 486–506.

[33] Abhijit Mishra, Diptesh Kanojia, and Pushpak Bhattacharyya, *Predicting readers sarcasm understandability by modeling gaze behavior*, (2016).

[34] Thomas M Mitchell, *Machine learning*, Machine Learning (1997).

[35] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell, *Text classification from labeled and unlabeled documents using em*, Machine learning **39** (2000), no. 2-3, 103–134.

[36] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith, *Improved part-of-speech tagging for online conversational text with word clusters*, Association for Computational Linguistics, 2013.

[37] James W Pennebaker, Martha E Francis, and Roger J Booth, *Linguistic inquiry and word count: Liwc 2001*, Mahway: Lawrence Erlbaum Associates **71** (2001), 2001.

[38] Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, and Daniele Blanc, *Enhance polarity classification on social media through sentiment-based feature expansion.*, WOA@ AI* IA, 2013, pp. 78–84.

[39] Amela Prelić, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler, *A systematic comparison and evaluation of biclustering methods for gene expression data*, Bioinformatics **22** (2006), no. 9, 1122–1129.

[40] Tomás Ptácek, Ivan Habernal, and Jun Hong, *Sarcasm detection on czech and english twitter.*, COLING, 2014, pp. 213–223.

[41] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu, *Sarcasm detection on twitter: A behavioral modeling approach*, Proceedings of the Eighth

ACM International Conference on Web Search and Data Mining, ACM, 2015, pp. 97–106.

[42] Antonio Reyes and Paolo Rosso, *On the difficulty of automatically detecting irony: beyond a simple case of negation*, Knowledge and Information Systems **40** (2014), no. 3, 595–614.

[43] Antonio Reyes, Paolo Rosso, and Davide Buscaldi, *From humor recognition to irony detection: The figurative language of social media*, Data & Knowledge Engineering **74** (2012), 1–12.

[44] Antonio Reyes, Paolo Rosso, and Tony Veale, *A multidimensional approach for detecting irony in twitter*, Language Resources and Evaluation **47** (2013), no. 1, 239–268.

[45] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang, *Sarcasm as contrast between a positive sentiment and negative situation.*, EMNLP, 2013, pp. 704–714.

[46] Patricia Rockwell, *The effects of cognitive complexity and communication apprehension on the expression and recognition of sarcasm*, Hauppauge, NY: Nova Science Publishers (2007).

[47] S Rasoul Safavian and David Landgrebe, *A survey of decision tree classifier methodology*, (1990).

[48] Robert E Schapire and Yoram Singer, *Boostexter: A boosting-based system for text categorization*, Machine learning **39** (2000), no. 2, 135–168.

[49] Dan Sperber and Deirdre Wilson, *Irony and the use-mention distinction*, Philosophy **3** (1981), 143–184.

[50] Michael Steinbach, George Karypis, Vipin Kumar, et al., *A comparison of document clustering techniques*, KDD workshop on text mining, vol. 400, Boston, 2000, pp. 525–526.

[51] Joseph Tepperman, David R Traum, and Shrikanth Narayanan, *" yeah right": sarcasm recognition for spoken dialogue systems.*, INTERSPEECH, Citeseer, 2006, pp. 1838–1841.

[52] R van Kruijsdijk, *Algorithm development in computerized detection of sarcasm using vocal cues*, (2007).

[53] Xiaojin Zhu and Andrew B Goldberg, *Introduction to semi-supervised learning*, Synthesis lectures on artificial intelligence and machine learning **3** (2009), no. 1, 1–130.

Vita

Candidate's full name: Hamed Minaee
University attended (with dates and degrees obtained):
University of New Brunswick, Fredericon, Master of Computer Science, 2013-2016
Islamic Azad University South Tehran Branch, Bachelor of Industrial Engineering, Iran, 2003-2009
Publications:
Conference Presentations: