Sarcasm Detection in Tweets

Saranya Rajagopalan
CEN-EE
Arizona State University
srajag25@asu.edu

Jad Aboul Hosn
CEN-CS
Arizona State University
jadhosn@asu.edu

Mandar Kulkarni
CEN-EE
Arizona State University
mkulka10@asu.edu

Salil Vartikar
CEN-EE
Arizona State University
svartika@asu.edu

Akshay Sonawane
CEN-CS
Arizona State University
apsonawa@asu.edu

Adithya Raju
CEN-CS
Arizona State University
aeraju@asu.edu

Abstract

Sarcasm is a subtle linguistic trait, where usually the author states the opposite of what they mean. Detecting sarcasm requires common sense and contextual knowledge which makes it a difficult problem to address. With the prevalence of sarcasm in tweets and comments online, it has become important to devise algorithms and models to distinguish between sarcastic and non-sarcastic statements. The Sentiment Analysis community has shown a great interest in Sarcasm detection given the recent advances in Natural Language Processing. In this project, we classify our set of features into four categories and then compare the performance of six different classifiers (Support Vector Machine, Logistic Regression, Naïve Bayes, Decision Trees, Random Forest and Neural Networks) using an incremental addition of one feature at a time. We also scrape, clean and pre-process our own corpus of Twitter data using Twitter Streaming API. We also introduce two new features to sarcasm detection (Passive Aggressive detection and Emoji polarity flips).

1 Introduction

"Sarcasm is the lowest form of wit, but the highest form of intelligence."
Oscar Wilde

Cambridge Dictionary defines Sarcasm as "remarks that mean the opposite of what they say" [1]. It is considered to be a form of verbal irony and is often used as a means of ridicule or comedic relief. Today, sarcasm has become a prevalent tool of communication in both, verbal and textual form. Adults recognize sarcasm on two main cues, the context in which the utterance is made and the way the speaker is speaking [2]. Other clues to verbal sarcasm might be a person's tone or other body language cues such as eye-rolling or hand gestures. On the hand, recognizing sarcasm in text, where none of these verbal cues are available, is a hot research topic with the rise in popularity of Natural Language Processing.

Over the past few years, social media websites like Twitter have become increasingly significant. Users of these websites have been expressing their ideas and opinions uninhibitedly [3]. These opinions may be on a large variety of topics including reviews on movies, products and political thoughts. They often employ sarcasm in more than one form, which makes sarcasm detection an interesting field of study. Millions of people all around the world are active on Twitter, some of them spreading a witty form of sarcasm. This informal nature of tweets and the use of an ever-evolving vocabulary with slangs words and abbreviations, along with a limit of 280 characters further adds to the challenge to sarcasm detection in tweets [3].

In this research paper, we present a series of trials to detect a combination of features that maximize model performance (F1 Score and Recall). We investigate different feature categories, and optimize existing features to fit recent changes in Twitter and Social Media trends. We begin our report with our problem statement, and the importance of sarcasm detection in Natural Language Processing in section II. Next, we explain our methodology and the intuition behind the feature engineering with in section III. Section IV explains the benchmarks, results and the reasoning behind our feature sequential testing.

1.1 Related works: methods and results

According to a recent survey on sarcasm detection [4], researchers tend to use semi-supervised approaches to extract patterns from tweets with an implicit sentiment. In [5], the author used a pattern matching algorithm to find sarcasm in online product reviews which can be related to the work of sarcasm detection in tweets. [6] is another semi-supervised approach that used bootstrapping to discover positive or negative verbs or positive or negative situations. Both these methods might not be effective in the situation of context incongruity. Some of these previous approaches that depend on detecting sarcasm using positive and negative words might fail in some cases.

In paper [7], identification of sarcasm in tweets is done using lexical and pragmatic features. The lexical features used were Parts Of Speech (POS) tags, WordNet, Interjections and punctuations, while the pragmatic features were emoticons and user mentions. Their collected data included tweets that contained hashtags – '#sarcasm (for sarcastic tweet)', '#happy (for positive sentiment)', '#angry (negative sentiment)'. The data was cleaned semi-automatically to address the concerns about corpus noisiness. They achieved an accuracy of 62-65% using Support Vector Machine and 60-63% using Logistic Regression on a specific set of features. [7] compared their models with the human labelled tweets to show the difference between the contextual understanding and sentimental analysis. Their study confirmed the necessity of contextual characteristics for sarcasm detection.

Sarcastic comments are not only found on twitter but also in product reviews and reddit. Some papers [8]-[9] considered Amazon product reviews instead of short text detection (Twitter). The considered feature set included positive and negative words, punctuation, hyperbole, inversion of polarity, ellipsis, interjection and bag of words [8]. The considered Logistic Regression, Linear SVM, Decision Trees, Random Forest and Naïve Bayes to train and test their accuracies, where Logistic regression reported the highest accuracy at 74%. In [10], the author used movie reviews. The reviews were classified into positive sentiment and negative sentiment. The used dataset consisted of 4000 reviews (2000 positive and 2000 negative). [10] achieved 89.71% using Support Vector Machine. They obtained their features using the Chi-square technique explained in [11]. [12] considered sentimental analysis on product reviews from e-commerce websites. They used opinion mining to classify a perception as positive, negative or neutral. Their features were extracted using POS tagging which detects the words with tags like NNS (noun plural), NN (Noun) and NNP (proper noun singular). Minimum support threshold was used to find all the features that the users were expressing their views on frequently. They achieved 88.13% using Support Vector Machine combined with a technique stated from [13] that classified their word vectors into two different classes.

Table 1 : Features and datasets summary

	Datasets	**Approach**			**Features**				
[14]	Collected dataset online. Training set: 8000 Test set: 4000	Naïve Bayes	SVM	Decision Tree	N-grams	POS tags	Emoticons	Sentiment Scores	Lexical

[15]	Dataset size: 19534. Equally divided in sarcastic tweets and non-sarcastic tweets	Binary Logistic Regression with l_2 regularization using ten-fold cross validation	Bigrams and Unigrams; Lexical features; POS tags; Capitalization; Whole Tweet sentiment; Tweet word sentiment; Intensifiers; Author history
[16]	Dataset size: 50000 from twitter	Parsing based Lexical Generation; Interjection Word Start	Sentiment Scores; Hyperbole
[17]	Pre-collected Dataset	Irony Detection Model	Punctuation marks; Word Length; Emoticons; Discourse Markers; POS tags; Semantic Similarity
[18]	Twitter specific dataset with size 8000	SVM; Decision Trees; Multinomial Naïve Bayes; Bayesian Networks	Emoticons; Onomatopoeic expressions; Punctuations; Lexical components
[8]	Based on Amazon reviews.	Scikit-learn; SVM; Naïve Bayes; Decision tree; Random Forest	Sentiment Score; Hyperbole; Punctuation; Ellipsis; Interjection; Emoticon
[7]	Twitter [6] data with #sarcasm tags. 900 tweets	SVM; Logistic Regression	Emoticons; Unigrams; Dictionary based
[6]	175000 tweets: 20% labelled as sarcastic and remaining non-sarcastic	Bootstrapping Learning	Positive Sentiments; Negative Sentiments
[3]	Twitter API	Behaviour modelling	Emoticons; Sentiment Score; Adjectives; Lexical Features; Capitalization; N-grams; Polysyllables
[19]	70 target words, 2,542,249 tweets. 80% training, 10% development, 10% test	SVM baseline; SVM using kernels	Positive and Negative Sentiments; Lexical Features
[20]	Seeding algorithm used. Seed positive word for positive data e.g. 'like', '#sarcasm' for negative	Random Forest; Naïve Bayes; Linear Regression	Lexical features; Pragmatic features; Positive-Negative phrases

2 Problem statement

People use twitter and other social media websites to convey their thoughts on a wide variety of topics. Sometimes, people find it difficult to recognize sarcasm as it requires the context in which the sentences were spoken along with the knowledge of the topic [21]. This intertwines with the challenges faced by machines for sarcasm detection. To better explain the difficulties in identifying sarcasm in text, consider this sentence "*Nice perfume. You must marinate in it.*" [22]. In this sentence, there is no negative words, yet it is classified as a sarcastic tweet. Sometimes, people like to convey their sarcasm openly, so they use hashtags such as "*#sarcasm*" or "*#sarcastic*". For example: "*What a nice day #sarcasm.*" However, most of the times users tend to keep their sarcasm discreet. Hence, the recognition of sarcasm by the audience/reader is important to avoid misunderstanding in everyday communication, and potentially introduce noise into our data. This autonomous labeling can also be used in the improvement of sentiment analysis.

Our goal is to tackle this problem hindering NLP. Twitter, considered short text mining, had a recent update in limit from 140 characters to 280 characters per tweet which still lacks enough length to convey the full meaning of a message and increases the ambiguity [3]. Current research on sarcasm detection using Twitter [7]- [16], [17] tend to refer to the psychological and behavioral behind sarcasm. [3] explains the importance of historical data of each author, and linking the authors to their previous tweets. 4.14% increase in accuracy was reported when 30 tweets from the same author were added to the model [3]. Yet, more than 30 tweets did not affect the accuracy. Therefore, if constrained to 30 tweets, there is a comparable performance increase.

Sarcasm detection should not depend only on the tweets and their features. In order to detect sarcasm with a higher accuracy, we should extend our range of features to a contextual feature space, as well as the author's historical data.

3 Methodology

3.1 Data collection and cleaning

Selection of datasets is one of the most important factors when it comes to the implementation of any classification model in general. The type of dataset used, its size and the balance between classes will affect the performance of the models. The reasons behind choosing this particular source [24] are that the size of this dataset is large and it provides us with the twitter IDs and not the tweets directly. One advantage in using this dataset, is the ability to retrieve the information of the author from the tweet IDs.

[23] divides the datasets into the following types and use similar techniques for our study:

3.1.1 Balanced dataset

This is a dataset that contains an equal number of tweets of each class. We obtain our dataset from [24] which contains 100,000 tweets (50,000 sarcastic and 50,000 non-sarcastic). Due to privacy settings of users being changed since the dataset was last updated in 2014, we were able to use 80,000 tweets out of this set.

3.1.2 Unbalanced dataset

This dataset has an unequal number of tweets from the 2 different classes. This dataset is also obtained from [24] and contains which contains 100,000 tweets (25,000 sarcastic and 75,000 non-sarcastic). Our dataset includes 80,000 tweets (20,000 sarcastic and 60,000 non-sarcastic).

3.2 Data preprocessing

For sarcasm detection on tweets, we extracted data real time from twitter using Tweepy API, but the major drawback of this step is that the data can be noisy. Before extracting any features, the data needs to be cleaned and filtered for less noisy features.

Some of the used data pre-processing and cleaning techniques:

- We remove tweets that start with '@User' as they are retweets and do not provide information about the original tweet which can potentially be sarcastic.

- We limit our study to English tweets as more resources are available for the processing of text in English language.
- In our project, we integrate emojis as a feature to detect sarcasm or at least a change in polarity.
- User mentions and URLs are removed from the tweet as they are not indicative of the original nature of the tweet.
- We remove duplicates that may result from retweets.

3.3 Feature engineering and extraction

In addition to the most commonly occurring Unigrams, we use 21 special features which we classify into the following categories [3]:

- Text expression-based features
- Emotion-based features
- Familiarity-based features
- Contrast-based features

3.3.1 Text expression-based features

This type of features relies upon the different connotations or ways in which a person expresses sarcasm in text. Social media users often include subtle markers within their comments (tweets) that indicate sarcasm to the reader [3]. The way these features are employed within the tweets casts doubts onto the author's sincerity.

3.3.1.1 Capitalization

As is mentioned in [26], people employ capitalization to lay emphasis on the emotion to be conveyed. We include this as one of our features since there is a high probability that an author may employ capitalize words in order to highlight sarcasm to create an extra impact. Thus, we find the number of capitalized words in each tweet and use it as a feature.

3.3.1.2 Exclamation marks

Exclamation marks are used to express emotions of a person. A user may include multiple exclamation marks to stress on a given emotion (e.g. *Wow!!!!!).* As referred to in [5], the number of exclamation marks in a tweet are counted. We then normalize the value to be in [0, 1] for each tweet by dividing the number by maximal observed value in our dataset.

Our underlying assumption behind using this as a feature is that when a person uses an extra number of exclamation marks, the emotion may be assumed to be genuine and hence not likely to be sarcastic.

3.3.1.3 Question marks

We count the number of question marks and normalize the value by dividing the number by the maximal number of question marks obtained for any tweet of our entire dataset.

We include this as one of our features under the assumption that the use of multiple question marks will show that a tweet is genuine and hence not likely to be sarcastic.

3.3.1.4 Noun and verb count

In this feature, we count the total number of nouns or verbs in the tweet. We normalize these values by dividing them by the total number of words (not tokens) in the tweet. Since emotions are not expressed using nouns, a greater percentage of nouns in a sentence might indicate a genuine statement.

3.3.1.5 Ellipsis

For this feature, similar to detection of question marks or exclamatory marks, we look for a string in the tweet that may contain consecutive ".". As mentioned by [8] this feature may often be followed by question or exclamatory marks. We assume that the presence of an ellipsis in the tweet may contribute to the existence of sarcasm.

3.3.1.6 Passive aggressiveness

Passive aggressiveness might indicate that the tweet is more serious than sarcastic. We use the presence of passive aggressiveness in the tweet as a binary feature by using regular expression (in Python). An example of passive aggressiveness: *I.AM.SO.DONE.*

3.3.1.7 Interjections

An interjection is a cry or an inarticulate utterance such as 'Alas!', 'Ouch!' or 'Wow'. This has been used as a pragmatic feature in [10] in order to identify ironies in sentences. Since sarcasm is highly related to irony, we choose to use this as a feature in our project. We assume that the no. of interjections in the tweet can be considered to make a tweet sarcastic. Instead of using the actual number of interjections as a feature, we normalize the numbers with respect to the maximum number obtained in all of the tweets.
Example of tweet containing interjections:

"Wow!! Thank you all for an amazing weekend! You can find the pictures of our event in Prague here: #TranceFamily"

3.3.2 Emotion-based features

People often use sarcasm to express their emotion indirectly. Such use of sarcasm can be seen as humor [27] or verbal aggression [28]. The following features are employed in cases where the tweets shed light on the underlying emotion of the tweet which is often negative.

3.3.2.1 Emoji sentiment:

The sentiments of the 85 most popular emojis in tweets are manually labelled as positive and negative values depending on the context in which they usually appear. The emoji sentiment of a tweet is calculatd as an average of the sentiment of all the emojis appearing in the tweet. If the emoji sentiment is opposite to the sentiment of the tweet, then the tweet could possibly be sarcastic.

3.3.2.2 Intensifiers

These are words (generally adverbs) that are used to give force or emphasis. Example: The word 'really' in "My feet are *really* cold". Here, the adverb really gives stress on the word cold in order to make the reader realize how cold it actually is.
[15] refers to intensifiers as lexical indicators and identifies if an intensifier is present in a given tweet or not. The presence or absence of an intensifier is then used as a feature in their model. The intensifiers are identified from a word list of the top 50 intensifiers drawn from Wikipedia (http://en.wikipedia.org/wiki/ Intensifier).
In our work, we identify intensifiers in each tweet. We then check the word appearing immediately after it and establish whether it has positive or negative sentiment. Using this we classify the intensifier as positive or negative and update the appropriate counter.

3.3.2.3 Words with repeated letters

Although not a very correct practice in terms of proper use of the English language and its grammar, users often tend to add extra letters in a word to stress on any emotion being expressed. Examples: lollllllll or Whaaaaat?
Similar to the use of interjections, we can identify this to be a pragmatic feature that can be used to identify sarcasm. Hence, we identify and count the number of words with repeated letters in tweets to be used as a feature in our model.
We use 'regular expression' to identify consecutive strings of letters in tokens that have more than 3 is repeated letters.

3.3.2.4 Sentiment score

Sentiment Score is the numerical representation of the sentiment polarity. In paper [3] sentiment score is calculated using SentiStrength [29]. SentiStrength detects sentiments based on the sentiment of the words in the tweets. SentiStrength assigns two scores to the words positive sentiment score and negative sentiment score. The range of the score is from -1 to -5 for negative sentiment and +1 to +5 for positive sentiment.

3.3.2.5 Skip grams

[25] explains skip-grams as sequence (pair) of words within the same tweet that are being used. These differ from N-grams such that they do not involve consecutive words. It is possible to skip words in between to consider one sequence and analyze it for irony. We extended a similar approach where we try to find words that imply sarcasm when used within the same tweet with the help of skip grams. It is important to note that the gaps between the words are limited to 2 or 3 words since the nature of tweets is such that there are on an average 12 words per tweet. Hence a higher gap will not be useful to use skip-grams as a feature. Skip-grams can skip words which do not add any meaning to the sentence like stopwords.

3.3.2.6 N-grams

N-grams are used as features in Natural Language Processing for sentiment analysis and provide useful information like most frequently used phrases and are also useful in understanding the polarity changes in the sentence. In our study, we have used most frequent bigrams and trigrams in our dataset as features as they provide valuable information for model training. We also got rid of the stopwords and punctuation as they do not add value to N-grams as features.

3.3.3 Contrast-based features

Sarcasm is often employed when a person says something while actually meaning the exact opposite. Thus, what the speaker/author meant is in complete contrast to what was spoken or written. Detecting this contrast employs the use of the following features:

3.3.3.1 Flip in polarity between sentence and emoji sentiments

Similar to the approach of flip in the sentiment of a sentence, we add this additional feature due its 'flip' nature. A tweet having a positive sentiment like happy but yet may end with a sad emoji. This shows that the author is using a positive sentence to describe something sad. This is an indicator of sarcasm. We consider two cases: positive sentence with a negative emoji or a negative sentence with a positive emoji.

3.3.3.2 Polarity flip in a sentence

A word can be said to have a certain polarity of being either positive, negative or neutral. A very basic definition of sarcasm may be said to be saying one thing and meaning the complete opposite. We use unigrams and bigrams in each tweet to check their polarity and proceed with finding out the polarity of all the unigrams or bigrams possible. If the polarity is flipped, we increment our polarity flip counter.

We check to see if there are any indicators that show that the meaning of a sentence has changed all of a sudden. Our underlying assumption for choosing this as a feature is that a flip in the polarity of a sentence may be an indicator of sarcasm.

Example: "*Late to work again. Awesome!!"*

In the example the user is late to work which implies a negative polarity. However, the word 'awesome' implies a positive polarity. Hence, we consider this a flip in polarity and use it as a feature.

3.3.3.3 Hashtag sentiment

For this feature, we attempt to find the sentiment of the # used in the tweet. The reason behind this is so that we can compare it with the sentiment of the entire tweet. We proceed with checking the polarity of both, the sentiment of the hashtag and the tweet to check if they are opposite. If they are opposite, we can say that they convey sarcasm and hence we use this as a feature.

3.3.3.4 Positive and negative word count

Positive and Negative word count may not independently add value to the analysis but when grouped with the right features, they prove to be very useful for sentiment analysis. These features added with polarity flip feature provide valuable information about the overall sentiment of the Tweet.

3.3.4 Context-based features

As mentioned by [30], one is often more comfortable with employing sarcasm when talking to people he or she is familiar to. Sarcasm is not often employed while conversing with a stranger. Also, according to [31] and [32], culture and language play a major role in terms of use and recognition of sarcasm. Thus, we imply

that familiarity is often useful when it comes to sarcasm detection and hence list the following features under this category.

3.3.4.1 User mentions

It has become common practice nowadays to address a specific tweet to a single user using the "@" symbol. As mentioned by [7], often these tweets using the '@user' can be identified to be pragmatic features when it comes to sarcasm detection. The usage of this as '@user' often implies that the current tweet is a reply to another tweet by the user. Since the user's tweet is already sarcastic, we delete all the tweets that begin with a user mention (@user). It is important to mention that we filter the tweet only if the user mention occurs at the beginning of the sentence.
Example of a tweet containing a user mention at the beginning:

"@Pokketsays I am rooting for my bed. These prex-mas days are killing me."

Table 2 : Features Summary

	Feature	**Short Explanation**
[7]	User mention	Example: @user We remove the tweet if the user mention occurs at the beginning
[5]	Exclamation count	We count the number of exclamations in each tweet and normalize the value between [0,1]
[5]	Question mark count	We count the number of question marks in each tweet and normalize the value between [0,1]
	Emoji sentiment	We use emojis from the top 85 emoji list provided by Gizmodo and manually label them as positive or negative
[8]	Interjections	We count the number of interjections and normalize the value between [0,1]
[15]	Intensifiers	Example: really, very, too We identify intensifiers and classify the words following them as positive or negative. We then update either the negative intensifier counter or the positive intensifier counter
[26]	Capital Words	We count the number of capitalizations and normalize the value between [0,1]
	Repeated letter words	Example: Lollllll, whaaaat
	Polarity Flip in a sentence	Obtain the polarity of all tokens in a tweet and look for flip in the polarity
	Sentiment score of the sentence	Obtain the sentiment score of the whole sentence using the vader package of the nltk Python library
	Flip in polarity between sentence sentiment and emoji sentiment	Check for flip in the sentiment of the tweet and the emoji to detect sarcasm. Flip may imply sarcasm.
	Noun and verb count	Larger percentage of nouns in a sentence might indicate less expression of sentiment
[33]	n-grams, n =3	Find sentiment and sum up the score Some can have chunk of polarity in sentence
	Top few most frequently used unigrams	Some unigrams are more prevalent in sarcastic tweets. So the most frequently used unigrams could be a good feature
	Passive Aggressiveness	Passive aggressive traits might have some correlation with the intention of the author

[25]	skip-grams	Skip intermediate words to check if other words are more important
	# sentiment	Check sentiment and compare with sentiment of whole sentence to check for flip
[8]	Ellipsis	Example: "…" We count the number of ellipsis in each tweet and normalize the value between [0,1]
	Positive word count	We count the number of positive words within the tweet and update the counter as a feature
	Negative word count	We count the number of negative words within the tweet and update the counter as a feature

Table 3 : Accuracy values from other sources

	Approach		**Result**	
[5]	5-fold cross validation	Human Annotators (Mechanical Turk)	Accuracy 0.896	F-Score 0.545
[7]	Polarity Based Classification		Accuracy 0.7589	
[6]	Bootstrapping Algorithm		F-1 Score- 0.51	
[3]	Behavioural Modelling		Accuracy- 0.8346	
[19]	Support Vector Machine		F-Score- 0.975	
[34]	General Architecture for Text Engineering (GATE) proposed by Cunningham et al. (2002)		F-Score- 0.91	
[10]	Support Vector Machine based on Chi-Square feature extraction [11]		Accuracy- 89.17%	
[20]	Random Forest	Weighted Ensemble (Naïve Bayes and Linear regression)	Accuracy- 84.7	Accuracy- 85.3

4 Results

4.1 Benchmarks

One of the major hurdles in Natural Language Processing is identifying sarcasm which requires contextual knowledge along with some distinct linguistic cues. Different approaches to achieve this goal of sarcasm detection have been tried over the years and appropriate feature selection still remains a big problem. In this section, we try to benchmark feature selection and model tuning based on recent state-of-the-art research papers [31] [18] [20].

Sarcastic comments generally start with a positive sentiment and ends with a negative tone or vice versa. The paper [20] exploits this particular characteristic to classify sarcastic text from non-sarcastic text. Their algorithm builds upon the sentiment polarity scores of different combinations of n-grams. These n-grams are constructed by performing a 'seeding' step which essentially identifies a particular word in the sentence which denotes a transition in the sentiment polarity flip. The authors also identify emoticons, positive words, negative words and discrepancies between sentence and emoji polarity as prominent features. We use a similar set of features using skip-grams to find the polarity flips in the tweets. We also consider emoji sentiment by manually classifying the most frequently used emojis on Twitter as positive or negative. To compare our models with [20], we train our Naïve Bayes and Random Forest models with a similar set of features and implement 10-fold cross validation on our dataset. As shown in the Table 4, our models performs equally well. The accuracies of our models are consistent with the benchmark models in [20].

Tweets can represent an entirely different set of writing styles and linguistic features like abbreviations, inappropriate punctuation and incorrect grammar. Hence, analysis of tweets in NLP poses multiple challenges and demands unique data pre-processing steps. [15] takes a novel approach to clean data and extracts unique features like intensifiers, number of capital words and pronunciation features. We train and test our Logistic Regression model for each feature individually and present a comparative study of the effect of each feature on sarcasm detection. The complexity of a Logistic Regression model is decided by the Lambda (λ) regularization parameter which balances the model complexity trade-off in order to avoid under fitting or over fitting. We select the optimum λ by calculating the validation error over a range of $[10^{-6}$-$10^{4}]$. This study shows that the value of 10^{-6} optimizes the model and fits the data with appropriate complexity. We compare the results of this model with evaluation of Logistic Regression model presented in [15]. The results as depicted in Table 4 provide a comparative analysis of our model with [15].

Irony and Sarcasm have similar linguistic features which change the meaning of sentence/comment to its exact opposite. Hence we use [8] as a reference for extracting the right set of features to analyse sarcastic tweets. We generate a similar feature set as [8] with intensifiers, punctuation, ellipsis and so on. We then compare F1-scores of our Naïve Bayes and Logistic Regression models with [8]. As we can see in the table below our F1-scores are comparable and highly consistent with [8] for Logistic Regression but not for Naïve Bayes. The reason for this might be the difference in the dataset and small differences in feature extraction techniques. The main idea behind benchmarking our models was to find some insights regarding the regularization parameters used in training. This experiment was useful in analysing the effect of model complexity on fitting our training data and optimize the regularization parameters.

Table 3 : Benchmarks

Benchmark: Contextualized Sarcasm on Twitter			**Benchmark: An Impact Analysis of Features in a Classification Approach to Irony Detection in Product Reviews**					**Benchmark: Sarcasm Detection of Tweets: A comparative Study**		
	Logistic Regression			**Logistic Regression**		**Naïve Bayes**			**Naïve Bayes**	
Features	*Accuracy*	*Benchmak Accuracy*	*Features*	*F1-score*	*Benchmark F1-Score*	*F1-score*	*Benchmark F1-Score*	*Features*	*Accuracy*	*Benchmark Accuracy*
Bigrams	64.42	69.5	All- positive words, negative words and punctuation	57.06	50.5	78.69	46.7	SentimentScore, POS, Emoji, N-grams	75.27	57
			All- positive words, negative words and ellipsis	57.05	50.9	78.97	47.8			
POS	51.103	66	All-Ellipsis Punctuation	57.11	50.3	83.09	47.9			
			All-Punctuation	57.11	50.6	82.45	45.6			
Upper Case	51.103	57.5	All-Interjection	57.11	50.6	83.33	45.6			
			All-Emoticon sentiment	57.04	50.8	78.24	46.8			
			All-Positive Word Count	57.1	18.1	82.11	11			
			All-Negative Word Count	57.7	50.9	79.45	48.3			
SentimentScore	55.8	55	Positive, Negative and Punctuation	18.86	0.9	73.13	1.8			
			Positive, Negative and Ellipsis	18.81	12.1	72.22	0			
			Ellipsis and Punctuation	0	0.8	36.18	0			
			Punctuation	0	9.8	23.45	8.6			
Intensifier	51.103	50.1	Interjection	0	0.5	13.56	0			
			Emoji Sentiment and Emoji Sentiment Flip polarity	0	0	66.48	0			
			Positive Word Count	0	50.4	46.32	45.6			

4.2 Case study: unbalanced dataset

For our study, we use the dataset from [35]. Following [23], we did a case study with imbalanced dataset in order to test the robustness of our model with 10,000 sarcastic tweets and 40,000 non-sarcastic tweets. We ran the tests for the four categories of features and for all the models. We compared the results with the same study performed using the balanced dataset from the same source with around 40,000 non-sarcastic tweets and 40,000 sarcastic tweets. Table 6 indicates that the accuracy of imbalanced datasets is surprisingly better than the accuracy of the balanced dataset. This can be explained by the poor F1 score. Accuracy denotes the ratio of correct predictions to all the classes. So if one of the class is sampled more than the other, then the accuracy would clearly be dominated by the accuracy of the dominating class. So in classification problem, we need to compare both F1 score and accuracy when we compare two models. Also, difference between the F1-scores of the Emotion based and Contrast based features and all features put together is widely different between the balanced and imbalanced datasets which indicate that these features are indeed useful in Sarcasm detection. The balanced dataset can be obtained by either oversampling sarcastic tweets or by under sampling the non-sarcastic tweets such that the ratio is 1:1. We have used oversampled data, because the more the merrier.

Table 4 : F1 Scores

SINGLE FEATURES	SVM		RandForest		LR		NB		DT		NN	
	Unbalanced	Balanced	Unbalanced	Balanced	Unbalanced	Balanced	Unbalanced	Balanced	Unbalanced	Balanced	Unbalanced	Balanced
Text Expression	0	0.65	0.36	0.53	0	0	0.22	0.6	0.5	0.63	0.13	0.67
Emotion	0.65	0.8	0.8	0.88	0.42	0.56	0.25	0.59	0.84	0.9	0.81	0.9
Contrast	0.44	0.47	0.34	0.87	0	0.46	0.48	0.84	0.68	0.88	0.65	0.87
Context	0	0	0	0	0	0	0	0.58	0	0.59	0	0.59
All	0.79	0.88	0.73	0.92	0.41	0.57	0.58	0.81	0.86	0.91	0.86	0.93

Table 5 : Accuracies

SINGLE FEATURES	SVM		RandForest		LR		NB		DT		NN	
	Unbalanced	Balanced	Unbalanced	Balanced	Unbalanced	Balanced	Unbalanced	Balanced	Unbalanced	Balanced	Unbalanced	Balanced
Text Expression	79.69	61.08	79.83	53.45	79.69	51.1	76.68	61.36	82.98	63.55	80.09	63.92
Emotion	87.47	81.56	92.07	85.16	75.25	62.83	77.18	59.46	93.56	89.51	92.34	90.4
Contrast	81.79	63.82	75.88	85.64	79.69	63.9	82.23	84.19	85.29	86.52	85.12	86.16
Context	79.69	51.1	79.69	16.92	79.69	51.1	79.69	53.23	79.69	53.29	79.69	53.27
All	89.56	89.79	94.01	92.3	75.24	64.77	80.1	79.99	94.27	90.85	94	93.05

4.3 Models comparison

In this section, we present the experimental results using different feature combinations and compare them. For each combination of features, we compare the results of Logistic regression, SVM, Naïve-Bayes, Decision Tree, Random Forest and Neural Networks. We use Macro-F1 score and accuracy as an evaluation scheme. For clarity, we have tabulated the results in Table 7. From the table below, we notice that the context based features performed worse than random or only a slightly better classifier. But emotion based features and contrast based features performed better than the rest. According to the accuracies and F1 Scores in Figures 1-2, emotion based features outperformed the other categories. We notice that in some case, having a single feature exceeded the performance of the category itself (e.g. Negative Word Count in Support Vector Machine). Also contrast-based features give consistent results for all the classifiers. We also notice that independently the features do not give very good accuracy but when we use them together they give good accuracy category-wise. Based on our results, we notice that since the emotion based features (six features) dominated the F1 score metric, we fixed these features as our base feature set and then started adding feature one category at a time. Contrary to what's expected, we noticed that even when having the full set of features, there will be a given combination of different features (that doesn't include the total 21 features) that will outperform the full set. This can be attributed to the fact that some of the tweets might not have all the features that we test for.

On the other hand, we also incrementally tested the addition of each feature category at time. As presented in Figure 3, the addition of the categories all doesn't show significant improvement. Ideally, if we add more features the accuracy has to improve. But we notice from the graphs (Figures 1, 2 & 3) that the accuracy decreases when Text-based and context based features are added. The poor accuracy while adding more features in some cases may be attributed to our binary classification problem, knowing that F1 score is a better metric in such cases. The most critical features are: trigram sentiments, positive word count, negative word count and emoji sentiment, positive word count, negative word count and emoji sentiment.

Table 6 : Feature Comparisons

SINGLE FEATURES	SVM				RandForest				LR				NB				DT				NN			
	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A
Text Expression based features	**0.58**	**0.75**	**0.65**	**61.08**	**0.52**	**0.53**	**0.53**	**53.45**	**0**	**0**	**0**	**51.1**	**0.61**	**0.59**	**0.6**	**61.36**	**0.62**	**0.65**	**0.63**	**63.55**	**0.61**	**0.73**	**0.66**	**63.92**
Noun count	0	0	0	51.1	0.46	0.57	0.51	46.78	0	0	0	51.1	0.57	0.81	0.67	61.22	0.58	0.8	0.67	61.37	0.57	0.8	0.67	61.16
Verb count	0	0	0	51.1	0.38	0.41	0.4	38.84	0	0	0	51.1	0.58	0.62	0.6	59.77	0.57	0.7	0.63	60	0.57	0.72	0.64	59.93
Exclamation	0	0	0	51.1	0.05	0.05	0.05	3.82	0	0	0	51.1	0.54	0.21	0.3	52.8	0.55	0.22	0.31	52.89	0.55	0.22	0.31	52.89
Questionmarks	0	0	0	51.1	0.1	0.11	0.11	6.6	0	0	0	51.1	0.56	0.13	0.22	52.66	0.57	0.14	0.22	52.66	0.56	0.13	0.21	52.5
Ellipsis	0	0	0	51.1	0.04	0.04	0.04	3.11	0	0	0	51.1	0.56	0.14	0.23	52.64	0.56	0.14	0.23	52.63	0.56	0.14	0.23	52.64
Interjections	0	0	0	51.1	0.06	0.07	0.07	4.75	0	0	0	51.1	0.66	0.07	0.14	52.94	0.67	0.07	0.13	52.95	0.66	0.08	0.14	52.94
Passive aggressive count	0	0	0	51.1	0	0	0	1.4	0	0	0	51.1	0.53	0.01	0.03	51.19	0.55	0.01	0.03	51.21	0.58	0	0.01	51.21
Uppercase	0	0	0	51.1	0	0	0	14.19	0	0	0	51.1	0.5	0.95	0.65	50.37	0.51	0.79	0.62	52.33	0.5	0.73	0.6	52
Emotion-based features	**0.87**	**0.73**	**0.8**	**81.56**	**0.89**	**0.87**	**0.88**	**88.16**	**0.67**	**0.48**	**0.56**	**62.83**	**0.58**	**0.51**	**0.6**	**59.46**	**0.9**	**0.88**	**0.89**	**89.51**	**0.9**	**0.91**	**0.9**	**90.4**
Positive Intensifier	0	0	0	51.1	0.06	0.06	0.06	4.42	0	0	0	51.1	0.68	0.69	0.12	52.89	0.68	0.07	0.12	52.89	0.68	0.07	0.12	52.89
Negative Intensifier	0	0	0	51.1	0.02	0.02	0.02	2.05	0	0	0	51.1	0.59	0.02	0.04	51.4	0.59	0.02	0.04	51.4	0.59	0.02	0.04	51.4
Repeat Letters	0	0	0	51.1	0.58	0.59	0.59	1.37	0	0	0	51.1	0.54	0.04	0.08	51.43	0.55	0.04	0.07	51.42	0.51	0.01	0.03	51.13
Sentiment score	0.58	0.57	0.58	58.69	0.48	0.51	0.49	48.48	0.59	0.32	0.41	55.81	0.59	0.45	0.51	57.53	0.59	0.67	0.63	60.76	0.57	0.76	0.65	60.23
Bigrams	0.73	0.43	0.54	64.43	0.74	0.63	0.68	71.16	0.73	0.43	0.54	64.43	0.73	0.43	0.54	64.42	0.77	0.74	0.76	76.66	0.78	0.7	0.74	75.88
Trigram	0.47	0.5	0.47	47.9	0.75	0.77	0.76	76.21	0.72	0.43	0.54	63.81	0.72	0.43	0.54	63.79	0.76	0.79	0.77	77.47	0.76	0.78	0.77	77.29
Skipgrams	0.47	0.51	0.49	48.32	0.81	0.8	0.8	81.16	0.71	0.45	0.55	64.32	0.61	0.45	0.52	58.89	0.82	0.81	0.81	81.89	0.81	0.81	0.81	81.51
Emoji Sentiment	0	0	0	51.1	0	0	0	4.51	0	0	0	51.09	0.5	0.98	0.66	51.55	0.5	0.98	0.66	51.55	0.5	0.69	0.58	51.4
Contrast-based features	**0.83**	**0.32**	**0.47**	**63.82**	**0.78**	**0.97**	**0.87**	**85.64**	**0.86**	**0.31**	**0.46**	**63.9**	**0.83**	**0.85**	**0.84**	**84.19**	**0.86**	**0.98**	**0.88**	**86.52**	**0.8**	**0.96**	**0.87**	**86.16**
polarity flip	0.83	0.32	0.47	53.41	0.83	0.74	0.79	80.4	0.8	0	0	51.33	0.83	0.75	0.79	80.4	0.84	0.75	0.79	80.4	0.84	0.75	0.79	80.4
Emoji Tweet Polarity flip	0	0	0	51.1	0	0	0	1.48	0	0	0	51.1	0.49	0.99	0.66	49.15	0	0	0	51.1	0	0	0	51.1
Positive_word_count	0	0	0	51.1	0.56	0.72	0.63	59.06	0	0	0	51.1	0.58	0.38	0.46	56.47	0.57	0.75	0.65	60.48	0.57	0.74	0.65	60.41
Negative_word_count	0.84	0.38	0.52	66.34	0.78	0.99	0.87	86.28	0.87	0.11	0.19	55.48	0.85	0.38	0.52	66.34	0.78	0.99	0.88	86.28	0.78	0.99	0.88	86.28
Context-based features	**0**	**0**	**0**	**51.1**	**0**	**0**	**0**	**16.92**	**0**	**0**	**0**	**51.1**	**0.51**	**0.67**	**0.58**	**53.23**	0.52	0.59	0.59	**53.29**	**0.52**	**0.69**	**0.59**	**53.27**
User mentions	0	0	0	51.1	0	0	0	16.92	0	0	0	51.1	0.51	0.67	0.58	53.23	0.52	0.59	0.59	53.29	0.52	0.69	0.59	53.27

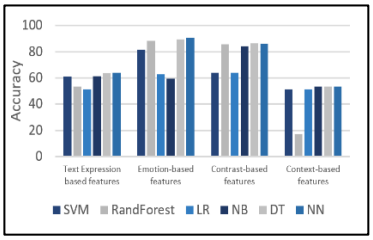

Figure 1 : Accuracy for feature categories

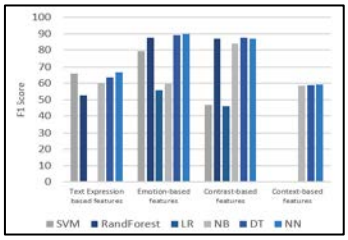

Figure 2 : F1 Score for feature categories

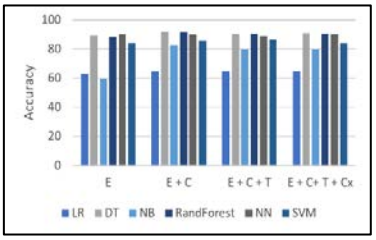

Figure 3 : Incrementally adding features

Legend: SVM - Support Vector Machines, Randforest: Random Forest, LR: Logistic Regression, NB: Naive Bayes, DT: Decision Tree, NN: Neural Networks; P - Precision, R - Recall, F1 - F1 Score, A - Accuracy

5 Conclusion

Sarcasm Detection requires unique feature engineering techniques by considering contextual characteristics. Our study shows that just increase in the number of features is not enough to achieve high accuracy, but selection of the right set of features is the basis of successful classification. These set of features will change with the change in dataset and data source. Sarcastic and non-sarcastic comments have different linguistic characteristics, hence the model needs to be trained with a balanced dataset. An unbalanced dataset may classify with higher accuracy but will produce low F1-score as the class with higher number of samples dominates the model training. Sarcasm is a behavioural trait and requires contextual analysis, hence expanding the feature space with author background and topic understanding will be an interesting area of future research. Additionally, previous tweet history from the same authors might lead to a more accurate contextual analysis.

6 References

[1] "dictionary.cambridge.org/us/," [Online]. Available: https://dictionary.cambridge.org/us/dictionary/english/sarcasm. [Accessed 27 April 2018].

[2] P. Rockwell, "Lower, slower, louder: Vocal cues of sarcasm.," *Journal of Psycholinguistic Research,* vol. 29, no. 5, pp. 483-495, 2000.

[3] A. Rajadesingan , R. Zafarani and H. Liu, "Sarcasm detection on twitter: A behavioral modeling approach," *Eighth ACM International Conference on Web Search and Data Mining*, pp. 97-106, 2015.

[4] A. Joshi, P. Bhattacharya and M. Carman, "Automatic sarcasm detection: A survey"," *ACM Computing Surveys (CSUR)*, 2017.

[5] O. Tsur , D. Davidov and A. Rappoport, "A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews," *InICWSM* , pp. 162-169, 2010.

[6] E. Riloff , A. Qadir, P. Surve, L. De Silva, N. Gilbert and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," *Conference on Empirical Methods in Natural Language Processing*, pp. 704-714, 2013.

[7] R. González-Ibánez, S. Muresan and N. Wacholder, "Identifying sarcasm in Twitter: a closer look," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers*, vol. 2, pp. 581-586, 2011.

[8] K. Buschmeier, P. Cimiano and R. Klinger, "An impact analysis of features in a classification approach to irony detection in product reviews," *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 42-49, 2014.

[9] D. Dmitry, O. Tsur and A. Rappoport, "Semi-supervised recognition of sarcastic sentences in twitter and amazon," in *Proceedings of the fourteenth conference on computational natural language learning*, 2010.

[10] N. Zainuddin and A. Selamat, " Sentiment analysis using support vector machine," *Computer, Communications, and Control Technology (I4CT)*, pp. 333-337, 2014.

[11] L. Ladha and T. Deepa, " Feature selection methods and algorithms," *International journal on computer science and engineering*, pp. 1787-1797, 2011.

[12] D. Devi, C. Kumar and S. Prasad, "A feature based approach for sentiment analysis by using support vector machine," *Advanced Computing (IACC), 2016 IEEE 6th International Conference*, pp. 3-8, 2016.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Handbook of Machine Learning*, pp. 273-297, 1995.

[14] A. Ghosh, L. Guofu, T. Veale, P. Rosso, E. Shutova, J. Barnden and A. Reyes, "Semeval-2015 task 11: Sentiment analysis of figurative language in twitter," *In Proceedings of the 9th International Workshop on Semantic Evaluation(SemEval 2015)*, pp. 470-478, 2015.

[15] D. Bamman and N. A. Smith, "Contextualized Sarcasm Detection on Twitter," *ICWSM*, pp. pp. 574-577, 2015.

[16] S. K. Bharti, S. B. Korra and S. K. Jena, "Parsing-based sarcasm sentiment recognition in twitter data," *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. pp. 1373-1380, 2015.

[17] D. I. H. Farías, V. Patti and P. Rosso, "Irony detection in Twitter: The role of affective content," *ACM Transactions on Internet Technology (TOIT)*, p. p.19, 2016.

[18] E. Fersini, F. A. Pozzi and E. Messina, "Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers," *Data Science and Advanced Analytics (DSAA), IEEE International Conference*, pp. pp 1-8, 2015.

[19] D. Ghosh, W. Guo and S. Muresan, "Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words," *Conference on Empirical Methods in Natural Language Processing* , pp. 1003-1012, 2015.

[20] T. Jain, N. Agarwal, G. Goyal and N. Aggarwal , "Sarcasm detection of tweets: A comparative study," *Contemporary Computing (IC3)*, pp. 1-6, 2017.

[21] M. Razali, A. Halin, N. Norowi and S. Doraisamy, "The importance of multimodality in sarcasm detection for sentiment analysis," *Research and Development (SCOReD), IEEE 15th Student Conference*, pp. 56-60, 2017.

[22] P. Chaudhari and C. Chandankhede, "Literature survey of sarcasm detection," *Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2041-2046, 2017.

[23] S. Poria, E. Cambria, D. Hazarika and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks," *arXiv preprint arXiv:1610.08815*, 2016.

[24] T. Ptáček, I. Habernal and J. Hong , Sarcasm detection on czech and english twitter, Dublin, Ireland: Dublin City University and Association for Computational Linguistics, 2014.

[25] A. Reyes , P. Rosso and T. Veale, "A multidimensional approach for detecting irony in twitter," *Language resources and evaluation*, vol. 47, no. 1, pp. 239-268, 2013.

[26] I. Hernández-Farías, J. Benedí and P. Rosso , "Applying basic features from sentiment analysis for automatic irony detection," *InIberian Conference on Pattern Recognition and Image Analysis*, vol. Springer, no. Cham, pp. 337-344, 2015.

[27] M. Bavasavanna, Dictionary of psychology, Allied Publishers, 2000.

[28] M. Toplak and A. Katz, "On the uses of sarcastic irony," *Journal of pragmatics*, vol. 32, no. 10, pp. 1467-1488, 2000.

[29] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai and A. Kappas , "Sentiment strength detection in short informal text," *Journal of the Association for Information Science and Technology*, pp. 2544-2558, 2018.

[30] P. Rockwell, " Empathy and the expression and recognition of sarcasm by close relations or strangers," *Perceptual and motor skills,* vol. 97, no. 1, pp. 251-256, 2003.

[31] H. Cheang and M. Pell, "Recognizing sarcasm without language: A cross-linguistic study of English and Cantonese.," *Pragmatics & Cognition,* vol. 19, no. 2, pp. 203-23, 2011.

[32] P. Rockwell and E. Theriot, "Culture, gender, and gender mix in encoders of sarcasm: A self-assessment analysis," *Communication Research Reports,* vol. 18, no. 1, pp. 44-52, 2001.

[33] A. Reyes and P. Rosso , " Making objective decisions from subjective data: Detecting irony in customer reviews," *Decision Support Systems,* vol. 53, no. 4, pp. 754-760, 2012.

[34] D. Maynard and M. Greenwood, "Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis," *InLrec,* pp. 4238-4243, 2014.

[35] T. Ptáček, I. Habernal and J. Hong , "Sarcasm detection on czech and english twitter," in *COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers,* 213-223, 2014.

[36] S. Kannangara, "Mining Twitter for Fine-Grained Political Opinion Polarity Classification, Ideology and Sarcasm Detection," *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining,* pp. 751-752, 2018.