– Independent Work Report Fall 2018 –

Automatic Sarcasm Detection on Twitter and Reddit through Sentiment Analysis

Hyeong Joon Suh
Adviser: Christiane Fellbaum

Abstract

Automatic sarcasm detection is a challenging task that has important applications in sentiment analysis. Previous works have implemented a wide variety of approaches such as using lexical, pragmatic and sentiment features of text. We model sarcasm as being expressed through a juxtaposition of contrasting sentiments and define sarcasm detection as a binary classification task of deciding if a piece of text is sarcastic or not sarcastic. We introduce a new method of engineering sentiment features and train five different machine learning classifiers to detect sarcasm in Twitter and Reddit data. We find that of the features engineered, the most important indicators of sarcasm are those that capture the contrast of sentiments of phrases in a sentence. Our best performing model obtains an F1 score of 0.65 on Twitter data, beating previous state-of-the-art performance by 7%. The same approach on Reddit data obtains an F1 score of 0.60 and falls short of state-of-the-art performance by 6%. The positive results show sentiment features are valuable indicators of sarcasm. The difference in results obtained from Twitter and Reddit data in spite of implementing the same approach illustrates that there is a difference in how sarcasm is expressed on different platforms.

1. Introduction

Sarcasm is a part of human interaction that is very subtle and expressed in many different ways. It is difficult to identify sarcasm since it often involves expressing one's opinion using language that implies the opposite. In spoken language, people rely on the tone of their voice or facial expressions to clarify their statement, but this is not possible in written text. This is why users on various social media platforms have developed a shared rule of indicating that they are being sarcastic. Twitter users use "#sarcasm" or "#sarcastic" and Reddit users use "/s" at the end of their sentences. The

development of these rules is testament to the fact that conveying and identifying sarcasm in text is difficult even for us humans.

Teaching machines to detect sarcasm in text has unsurprisingly proved to be an incredibly challenging task and has attracted a lot of interest. The main motivation for researching this topic is to improve the accuracy of sentiment analysis. Sarcastic comments can easily confuse sentiment analysis models. For example, the tweet "So excited to be returning to work tomorrow #sarcasm" is expressing a negative sentiment but contains the phrase "so excited", making the tweet's sentiment difficult for machines to decipher. Sentiment analysis has a wide range of important applications. Using sentiment analysis, businesses can better understand customers' opinions on their products, politicians can gauge people's views on certain situations, and financial institutions can anticipate market movements. In the past, the US Secret Service sought engineers to build a software system to detect sarcasm on Twitter to accurately analyze social media data.[1]

The primary goal of our research is to develop a machine learning model based on sentiment analysis that accurately detects sarcasm in text. In doing so, we seek to make three main contributions. First, we introduce new approaches to engineering sentiment features of text that can be used to train machine learning models. Secondly, we evaluate the effectiveness of using only sentiment features to detect sarcasm. Lastly, we examine if sarcasm is expressed differently on Twitter and Reddit by assessing our model's performance on two different datasets.

2. Related Work

Sarcasm has been studied extensively and in a variety of ways as a linguistic phenomenon. Camp (2012) studied sarcasm in terms of meaning inversion, where a speaker expresses a position while actually communicating the opposite. The paper divided sarcasm into four different types, perlocutionary, propositional, 'like'-prefixed and lexical, by discussing how each type differs in the manner in which meaning inversion is manifested. [3] Kreuz and Caucci (2007) asked college students to read excerpts from published works and then to classify the text as being sarcastic or

[1] https://www.bbc.com/news/technology-27711109

not sarcastic. The study found that lexical factors such as interjections (e.g. “gee”, “gosh”), certain formulaic expressions (e.g. “thanks a lot”, “good job”) and repetitions (e.g. “perfect, just perfect”) were significant predictors of the participants’ rating of sarcastic comments. [10]

Research in automatic sarcasm detection formulates the problem as a classification task, where given a piece of text, the goal is to correctly classify it as sarcastic or not sarcastic. Many approaches base their methodology on linguistic studies of sarcasm. Gonzalez-Ibanez et al. (2011) investigated the effectiveness of using lexical and pragmatic factors in building machine learning models to identify sarcasm in text. Lexical features include interjections and punctuations mentioned in Kreuz and Caucci (2007) among others and pragmatic features comprise of positive and negative emojis and Twitter’s ‘@user’ mention, which marks a tweet as being a reply to another tweet. [5] Reyes et al. (2012) constructed a model based on four types of features: signatures (e.g. punctuation marks, emojis), unexpectedness (e.g. temporal and contextual imbalance), style (e.g. character-grams, skip-grams, polarity skip-grams) and emotional scenarios (e.g. imagery, pleasantness). [14] Liebrecht et al. (2013) claimed that sarcasm is often signaled by hyperbole and that explicit markers such as “#sarcasm” on twitter are used by people to convey sarcasm when using non-hyperbolic language. [11]

Among various wide-ranging approaches to automatic sarcasm detection, methods that incorporate sentiment features are some of the most successful. Riloff et al. (2013) targeted a specific type of sarcasm that consists of a positive sentiment contrasted with a negative situation. They proposed a bootstrapping algorithm that learns positive sentiment phrases and negative situation phrases from sarcastic tweets and use them to detect sarcasm. Table 1 shows a subset of the positive phrases and negative situations learned. This method achieved an F1 score of 0.51 on a dataset of 3,000 tweets. [15] Joshi et al. (2015) based their approach on a linguistic theory called context incongruity. In addition to lexical features consisting of unigrams from the training corpus and pragmatic features such as capitalization, emojis and punctuation marks, they implement the same bootstrapping method from Riloff et al. to learn sentiment phrases. They report an F1-score of 0.61 on the same dataset used by Riloff et al., 0.89 on a different dataset of tweets and 0.64 on a dataset

of discussion forum posts.[7]

Type (Count)	Phrase
Positive Verb Phrases (26)	missed, loves, enjoy, cant wait, excited, wanted, can't wait, get, appreciate, decided, loving, ...
Positive Predicative Expressions (20)	great, so much fun, good, so happy, better, my favorite thing, cool, funny, nice, always fun, ...
Negative Situations (239)	being ignored, being sick, waiting, feeling, waking up early, being woken, fighting, staying, writing, being home, cleaning, not getting, crying, sitting at home, being stuck, ...

Table 1: Positive phrases and negative situation phrases learned in Riloff et al (2013). with their bootstrapping algorithm [15]

There are several limitations to these two approaches. First, Riloff et al. (2013) addresses only a subset of ways sarcasm is expressed on Twitter. Sarcasm is not limited to being expressed through a positive sentiment contrasted with a negative situation. Rather, it can be captured more generally as a contrast of positive and negative sentiments. Furthermore, sarcasm is often expressed through the use of hashtags and emojis, especially on social media platforms. [13] Joshi et al. (2015) addresses these limitations to an extent by learning negative sentiment and positive situation phrases in addition to positive sentiment and negative situation phrases. However, it is difficult to infer how much of their impressive performance can be attributed to sentiment features, because they report similar F1 scores when their model is trained only on lexical and pragmatic features.

In this paper, we attempt to address these limitations. We fully embrace the approach of modeling sarcasm as being expressed through a contrast of sentiments and seek to test this approach by isolating sentiment features as the only features used to train our machine learning models. Furthermore, we do not use a bootstrapping algorithm and use our custom method of extracting a wider range of sentiment features from text.

3. Approach

The core idea behind our approach is that sarcasm is expressed through a juxtaposition of contrasting sentiments. We hypothesize that if adjacent phrases have opposite sentiment scores or if the variation of sentiment scores of phrases is large across a piece of text, the text is likely to be sarcastic. We use

sentiment model developed by Socher et al. (2013) [16] to decompose a sentence into a binary tree of phrases with sentiment annotations. From this tree, we engineer five features: sentence sentiment score, maximum sentiment score, minimum sentiment score, sentiment score range and adjacent sentiment contrast score. We will use the phrase "I love how terrible my day has been" to explain what each feature represents. Figure 1 is a visual representation of how the sentence is decomposed using Socher et al.'s model.

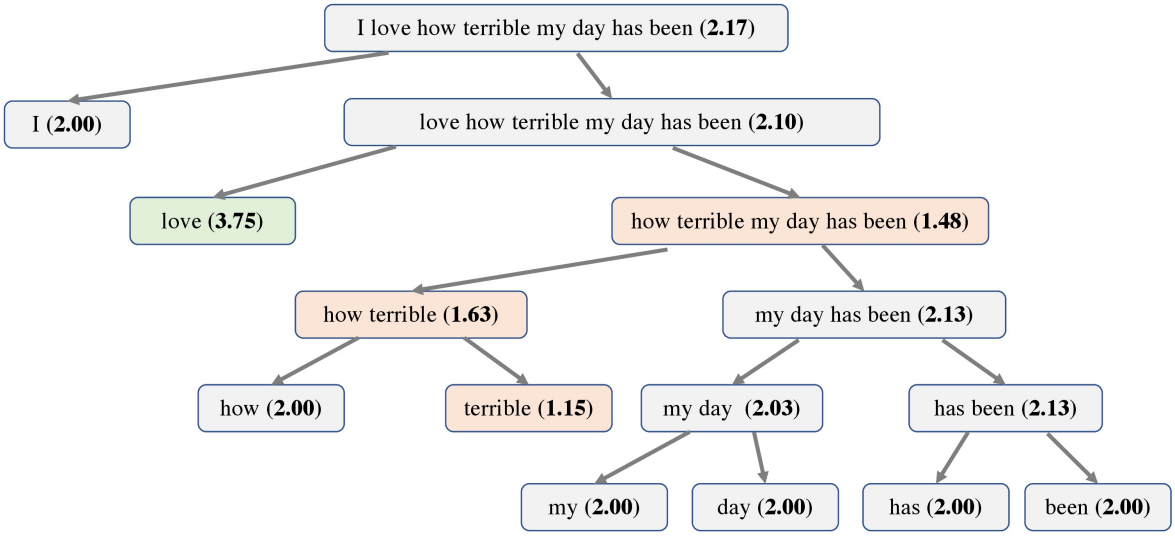

Figure 1: Sentiment annotated tree of phrase "I love how terrible my day has been." Each node contains a phrase and its sentiment score in parentheses. Sentiment score ranges from 0 to 4, where 0 is very negative, 2 is neutral, and 4 is very positive.

The sentence sentiment score is simply the sentiment score of the whole sentence, which in this case is 2.17. The maximum and minimum sentiment scores are the highest and lowest sentiment scores of the whole tree. In our example, the phrase "love" has the highest with 3.75 and the phrase "terrible" has the lowest with 1.15. The sentiment score range is the difference between the maximum and minimum sentiment scores $(3.75 - 1.15 = 2.60)$ and is designed to capture the variations of sentiment across a sentence. Lastly, adjacent sentiment contrast score is the maximum difference of sentiment scores of adjacent phrases or, in other words, the difference of sentiment

scores of two children nodes with the same parent. In the example, the two adjacent phrases with the largest contrasting sentiments are “love” and “how terrible my day has been.” Therefore, the adjacent sentiment contrast score is 2.27. This feature is designed to detect if there are phrases of opposite sentiments juxtaposed together.

The way sarcasm is expressed is often different across websites because the nature of interaction differs depending on the websites’ design and purpose. In our study, we use datasets from Twitter and Reddit and we engineer additional sentiment features specific to each platform to address the different ways sarcasm may be expressed on these websites.

Twitter, like most social media platforms, has seen emojis become an essential part of user communication. Since Twitter has a character limit on a single post (140 traditionally, 280 as of 2018) and emotions are at times difficult to convey with only traditional text, users often turn to emojis to convey rich emotional content with a single character. It was important to include the sentiment analysis of emojis in our study because emojis can be a way for users to express sarcasm. Consider the following tweet as an example: “I fully support the government’s new immigration policies ☹☹☹.” Without the emojis, the tweet can be mistaken as expressing genuine support for the government. It is clear that the emojis represent the user’s true sentiment and that the text itself is sarcastic. We try to capture this form of sarcasm by unpacking the emotional content of emojis using an emoji sentiment lexicon and analyzing how much this differs from the sentiment score of the text.

Reddit is best described as a discussion forum rather than a social media platform. The design of the website is optimized for users to post their opinions and for lengthy discussions to follow in the form of chains of comments. We worked with a large dataset of sarcastic and non-sarcastic comments on posts and identified that sarcasm is often used to make a comment about the parent post. For example, a user commented sarcastically “Thanks Trump!” on the post “US airstrikes hit Syrian mosque, Human Rights Watch says.” On discussion forums like Reddit, it is important to understand the context of a comment to evaluate whether the comment is sarcastic, more so than on Twitter. This is why we felt sentiment analysis on comments by itself is not enough. Therefore, we

measure the sentiment score difference of a comment and its parent post to try to assess if Reddit users convey sarcasm through conveying a contrasting sentiment to a post.

We model sarcasm detection as a binary classification task of deciding whether a piece of text is sarcastic or not sarcastic. Using the features described, we train machine learning classifiers to detect sarcasm and we evaluate the performance by comparing F1 scores with results of previous studies.

4. Implementation

The implementation can be broken down into three steps: data collection, feature engineering and machine learning model training. The code used to implement all steps in this section can be found at https://github.com/hjsuh18/sarcasmNLP.

4.1. Data

In this study, we used two datasets taken from Twitter and Reddit. We chose to implement the same approach to two datasets to evaluate the approach on different types of text data and to analyze how sarcasm is expressed in different forms depending on the environment. There were a range of datasets that we could choose from, including movie reviews, Amazon product reviews and excerpts from books and TV shows. We chose Twitter because it has been studied and tested extensively as a data source for sarcasm detection [6] and we chose Reddit because Khodak et al. (2018) [8] created the largest corpus for sarcasm research from Reddit called the Self-Annotated Reddit Corpus (SARC) [1]. Table 2 presents the number of data points in the final datasets used.

	Twitter	**Reddit**
Sarcastic	1209	160,874
Non-sarcastic	1209	160,874
Total	2418	321,748

Table 2: Breakdown of number of data points in the Twitter and Reddit datasets used

4.1.1. Twitter Data Collection

We used Twitter's streaming API to collect tweets. We collected tweets from November 17, 2018 to

December 28, 2018 and limited the tweets to only English tweets tweeted from within the United States. We managed to collect over 10 million clean tweets within this period of which we labeled 1209 to be sarcastic. To create a balanced dataset, we randomly selected 1209 non-sarcastic tweets from a dataset containing all tweets collected.

4.1.2. Twitter Data Cleaning

Tweets streamed using Twitter's API needed to be cleaned in multiple ways. There were many tweets that we simply did not include in our dataset. First, there were retweets, which are re-postings of another user's tweet. To avoid duplicate entries in our data, we removed any tweets that started with "RT @", which is the mark for a tweet being a retweet. Secondly, we removed any tweets that started with the mention tag "@", which is used to direct a tweet to another user. In the case where a tweet is a direct reply to another tweet, the contents of the tweet are likely to be heavily related to contextual information from the other tweet. The lack of context makes the tweet too ambiguous to judge whether it is sarcastic and so we decided to remove such tweets altogether. Lastly, we removed any tweets with URL's for a similar reason. URL's usually link to an image or a news article and without analyzing this extra contextual information, these tweets would only add noise to our data.

We cleaned the remaining tweets in several steps. First, we converted all our text to lowercase to treat same words with different capitalizations as being the same. Second, we removed "#sarcasm" and "#sarcastic" from the text since these are used as our sarcasm labels and we did not want our model to learn from these hashtags. Third, we removed the "#" character from hashtags. We initially planned on removing all hashtags from the text but found that hashtags often contain strong sentiment words that play an important role in conveying sarcasm. Fourth, we removed user mentions in the form of "@user". User mentions in the middle of tweets (not user mentions at the start of tweets) are often used to refer to people or as a substitute for the name of a user. We decided these mentions would not affect our sentiment features and only confuse the text parser. Fifth, emojis were removed from the tweet but saved to be analyzed separately. Finally, the characters "&", ">" and "<" were represented as "&", "<" and ">" respectively in the streamed data. To

help our model understand these characters, we replaced “&” with “and” and simply removed “<” and “>” from the text. “<” and “>” can take on many different meanings, such as “less than” and “greater than” in mathematical notation, as part of arrows, “->”, and as part of a textual representation of a heart, “<3”. Instead of arbitrarily choosing one of these interpretations, we decided to simply remove the characters. Figure 2 demonstrates how tweets were cleaned following these steps.

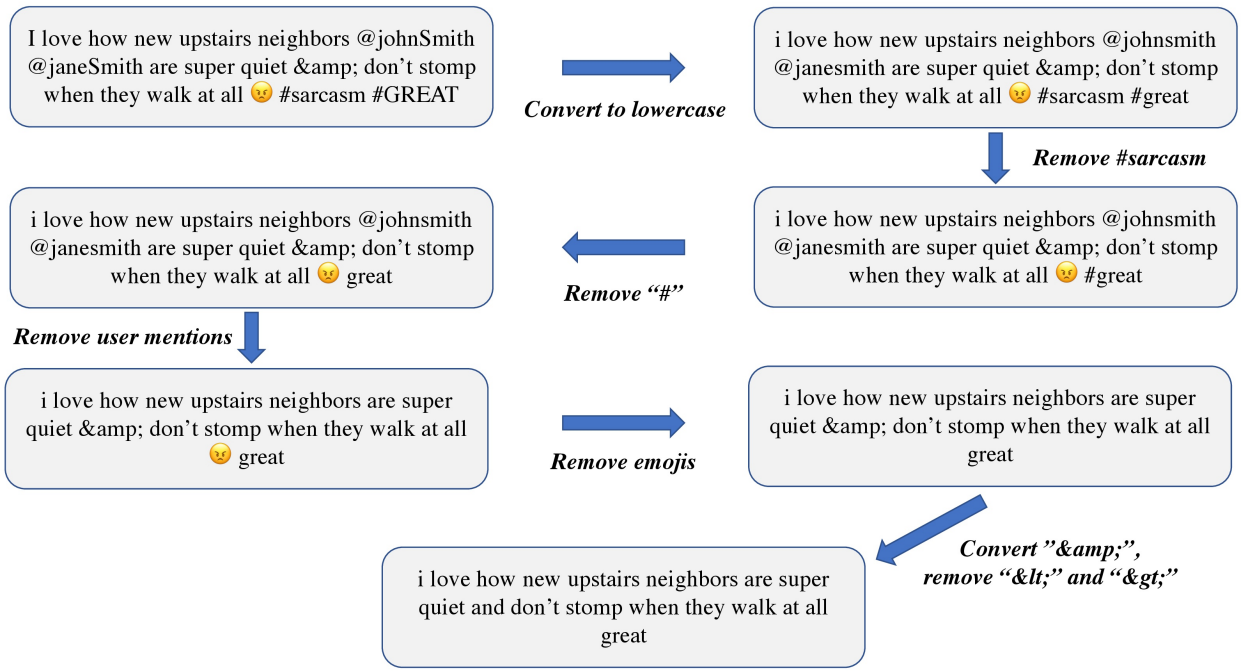

Figure 2: Data cleaning pipeline for streamed tweets

4.1.3. Twitter Data Labeling

We labeled tweets as being sarcastic if the tweet contained either “#sarcasm” or “#sarcastic” hashtags and not sarcastic if the tweet did not. This is a widely adopted approach in sarcasm detection tasks since it allows researchers to collect a large number of sarcastic tweets quite easily. [6] Since no one but the author can truly verify that a tweet was intended to be sarcastic, it is valuable to collect self-annotated data using this method. However, we cannot fully trust that users always correctly label tweets with the “#sarcasm” tag. There are tweets that are not sarcastic that contain “#sarcasm” and tweets that are sarcastic but not tagged with “#sarcasm”. To overcome this challenge, there have

been studies that use manually annotated data, but we simply opted use “#sarcasm” and “#sarcastic” tags to collect a larger dataset of sarcastic tweets, albeit an imperfect one. [6]

4.1.4. Reddit

There was little work to be done to prepare the Reddit dataset compared to the Twitter dataset. This is because the Self-Annotated Reddit Corpus (SARC) [1] contains data that is already cleaned and labeled. The dataset contains 533 million comments in total, 1.34 million of which are sarcastic. We decided to use a balanced filtered subset of the corpus, which consists of a total of 321,748 comments, 160,874 of which are sarcastic. This dataset is self-annotated (the authors use the “/s” label in Reddit as an indicator for sarcasm) and its format reflects Reddit’s tree-like conversation structure. Table 3 is a small subset of the dataset that demonstrates the dataset’s structure.

Parent Post	**Comment**	**Sarcastic**
"...two-income families often have even less income left over today than did an equivalent single-income family 30 years ago, even when they make almost twice as much."	Chalk it up to the ever-increasing cost of freedom.	True
	We’re about to finally get affordable housing, and now the politicians are doing everything they can to keep prices *high*.	False
Heath Ledger Wins Oscar!	oh wow I am so surprised I never saw this coming	True
	It’ll look so good IN HIS COFFIN NEXT TO HIS CORPSE!	False
atheist toast	Everything I need to know I learned from toast.	False
	You guys are completely lost, God obviously made that toast.	True

Table 3: Small subset of Reddit dataset

Reddit is designed such that there is an initial post, which we refer to as the “parent post”, and there are chains of comments about the post. The balanced dataset is engineered such that there is one sarcastic comment and one non-sarcastic comment for each parent post.

4.2. Feature Engineering

As outlined in Section 3, there are sentiment features extracted from text that is common to both datasets and there are features that are engineered to capture certain characteristics specific to

Twitter and Reddit. Table 4 gives a clear overview of all the features.

Feature Name	Description (range)
1. Sentence sentiment score	Average sentiment score of sentences in text (0 - 4)
2. Maximum sentiment score	Sentiment score of most positive phrase (0 - 4)
3. Minimum sentiment score	Sentiment score of most negative phrase (0 - 4)
4. Sentiment score range	Max. sentiment score – min. sentiment score (0 - 4)
5. Adjacent sentiment contrast score	Absolute difference of sentiment scores of adjacent phrases with largest sentiment contrast (0 - 4)
6. Emoji sentiment score (Twitter)	Average sentiment score of emojis in tweets (0 - 4)
7. Emoji-text contrast score (Twitter)	Absolute difference of sentence sentiment score and emoji sentiment score (0 - 4)
8. Parent sentiment score (Reddit)	Average sentiment score of parent posts (0 - 4)
9. Parent-comment contrast score (Reddit)	Absolute difference of sentence sentiment score and parent sentiment score (0 - 4)

Table 4: Overview of all sentiment features used

4.2.1. Sentiment Features using Socher et al.'s Sentiment Tree Model

We rely heavily on Socher et al. (2013)'s sentiment analysis model [16] to engineer sentiment features. This deep learning model understands the grammatical structure of sentences and annotates the sentence with sentiment scores based on how words compose the meaning of longer phrases. We use the model to generate a sentiment-annotated tree like Figure 1 and we simply traverse through the tree to generate sentiment features 1 - 5. Thankfully, the model is easily implemented through Stanford's CoreNLP [12], which is a large collection of natural language processing tools.

We initially planned on splitting sentences into phrases of equal length and using SentiWordNet [2] to obtain sentiment scores for each phrase. SentiWordNet labels synsets in WordNet with a sentiment and an objectivity score. The sentiment score of a phrase or sentence can be obtained by taking the average of the sentiment scores of words. While this method may also have worked, we valued Socher et al. (2013)'s model's ability to understand the grammatical composition of sentences since this allowed us to parse sentences into phrases in a more intelligent manner.

4.2.2. Twitter-specific Features

We use an emoji sentiment lexicon created by Novak et al. (2015) [9] to evaluate the sentiment score of an emoji. The study presents a dataset of 969 emojis and each emoji has a count of how

many positive, neutral and negative tweets it has appeared in. A sentiment score is engineered by giving each positive occurrence 4 points, neutral occurrence 2 points and negative occurrence 0 points and taking the average. For example, the most popular "tears of joy" emoji has occurred in 6845 positive tweets, 4163 neutral tweets and 3614 negative tweets according to the dataset. Therefore, its sentiment score is $\frac{6845 \cdot 4+4163 \cdot 2+3614 \cdot 0}{6845+4163+3614} \approx 2.44$. We calculate the absolute difference of this score and the sentiment score of the rest of the tweet to obtain the emoji-text contrast score.

4.2.3. Reddit-specific Features

We simply use Socher et al.'s model to obtain the average sentiment score of a comment's parent posts. The parent-comment contrast score is the absolute difference between this score and the comment's sentence sentiment score.

4.3. Machine Learning

We approached sarcasm detection as a binary classification task. This is the most popular approach, especially with Twitter data. However, Khodak et al. (2018) took a different approach to test their Reddit sarcasm corpus and trained their models to distinguish which of two comments of a Reddit post is sarcastic, given that one is sarcastic and the other is not. [8] We decided not to follow Khodak et al. for a couple of reasons. First, we wanted to follow the same approach taken with the Twitter dataset to allow for a more direct comparison of results. Secondly, Khodak et al.'s model may be able to detect which of two comments is more likely to be sarcastic but is incapable of actually distinguishing whether a comment is sarcastic or not.

The Twitter dataset was randomly divided into training (80%) and testing (20%) datasets. The Reddit data was provided with training and testing data already divided as 80% and 20% of the complete dataset respectively. First, only features 1 - 5 on Table 4 were used to train and test classifiers. Then, we trained models with the dataset-specific features (features 6 and 7 for Twitter, 8 and 9 for Reddit) in addition to features 1 - 5. We trained five different classifiers: support vector machine with linear kernel, support vector machine with RBF kernel, gaussian process, random forest and multilayer perceptron neural networks. These classifiers were chosen based on their

good performances in initial testing and the best parameters for each classifier were found after hyperparameter optimization with 5-fold cross-validation.

5. Results & Evaluation

5.1. Twitter Dataset

5.1.1. Statistical Analysis

Before examining the performance of the trained models, it is informative to perform basic statistical analysis on the sentiment features. Figure 3 shows boxplots of all sentiment features grouped in pairs such that the difference between features of sarcastic tweets and non-sarcastic tweets are clear. Table 11 in Section 8, the Appendix, also provides various statistics of the dataset.

The goal of engineering a wide variety of sentiment features was to identify if any of them can be used as a differentiating indicator between sarcastic and non-sarcastic tweets. A couple of features that stand out as potential candidates are the score range and adjacent contrast features. For both features, sarcastic tweets have a mean score that is roughly half a standard deviation above the mean score of non-sarcastic tweets. These statistics support our approach of modeling sarcasm as being expressed through a contrast of sentiments.

Another feature that stands out is the emoji score. For both sarcastic and non-sarcastic tweets, most data points that are visible on the box plot are marked as outliers and the box that marks the middle half of the data is hardly visible. The reason for this is the bad quality of our data. Only 352 out of 2418 tweets in our dataset had an emoji in them. When there was no emoji, we set the emoji score as neutral (2.0), which is why the majority of the data points on the box plot are concentrated on 2.0. This also impacts the emoji-sentence contrast score, which was originally designed to capture how sarcasm is expressed by emojis' contrasting sentiment to the remainder of a tweet.

5.1.2. Machine Learning Classification Models

Table 5 shows the performance of five different machine learning models that were trained using features 1 - 5. The F1 score is used to compare and evaluate our models against each other and

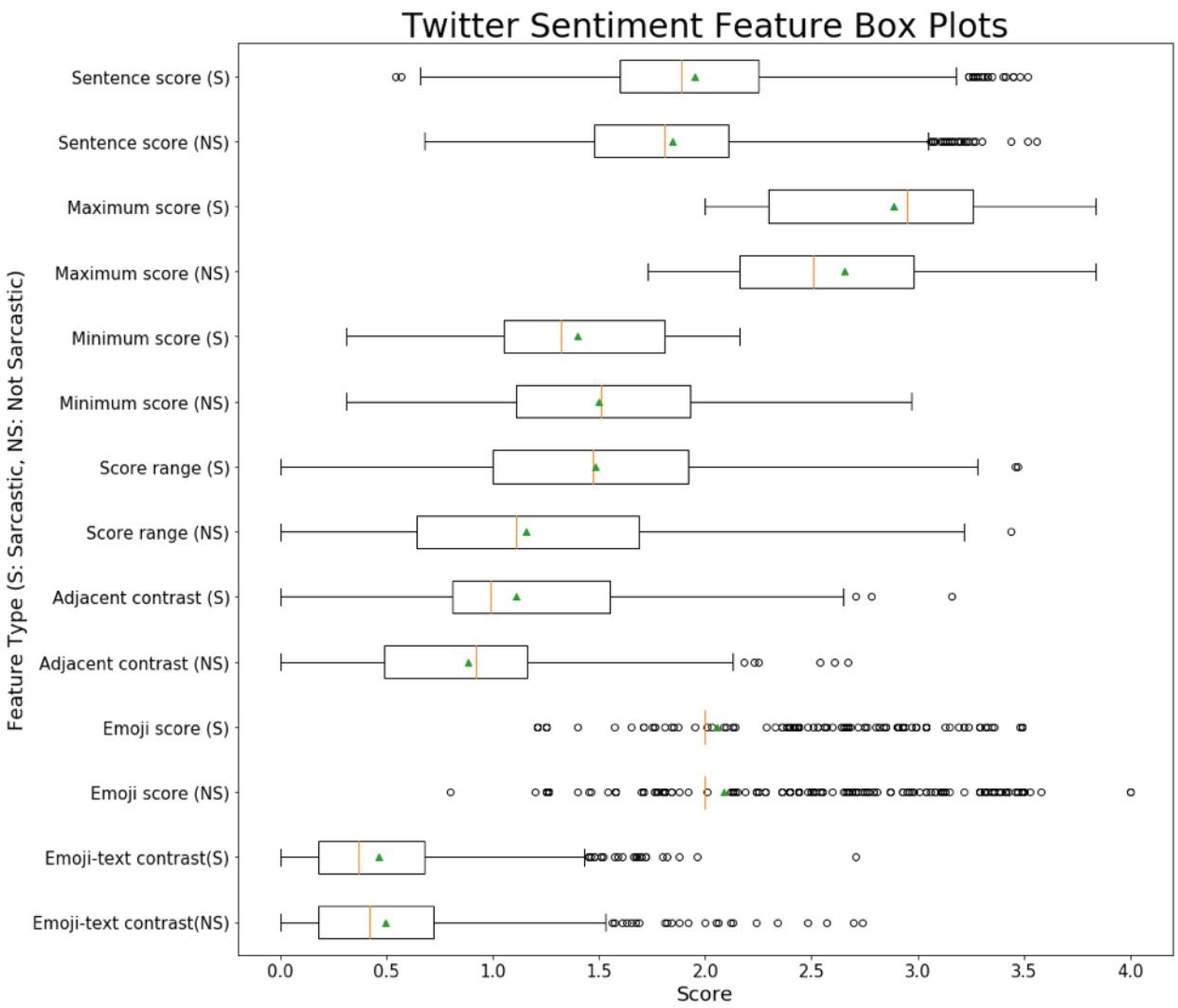

Figure 3: Boxplot illustrating the statistical distribution of Twitter sentiment features. The red line represents the median and the green triangle represents the mean. Circles outside of the boxplot's whiskers are outliers.

also against results from previous research. For comparison, a model that guesses randomly would achieve an accuracy, precision, recall and F1 score of around 0.5 since our dataset is balanced. All of our models achieve superior F1 scores, which suggests that the models have all partially learnt how to detect sarcasm in tweets. The multilayer perceptron classifier obtained the highest F1 score, although the RBF kernel support vector machine and random forest classifier also performed well.

Given the low quality of the two emoji sentiment features, it is no surprise that the addition of these features in training did not have a large effect on performance metrics. Table 6 shows the results, including the percentage improvement from the results in Table 5.

Model	Accuracy	Precision	Recall	F1 score
SVM (Linear)	0.610	0.642	0.498	0.561
SVM (RBF)	0.629	0.632	0.614	0.623
Gaussian Process	0.612	0.623	0.568	0.594
Random Forest	0.616	0.613	0.631	0.622
MLP	0.629	0.625	0.643	0.634

Table 5: Performance of machine learning models trained on Twitter data using features 1 - 5 from Table 4

Model	Accuracy (%change)	Precision (%change)	Recall (%change)	F1 score (%change)
SVM (Linear)	0.612 (+0.33)	0.632 (-1.56)	0.535 (+7.43)	0.580 (+3.39)
SVM (RBF)	0.620 (-1.43)	0.624 (-1.27)	0.606 (-1.30)	0.615 (-1.28)
Gaussian Process	0.614 (+0.33)	0.626 (+0.48)	0.568 (+0.0)	0.596 (+0.34)
Random Forest	0.624 (+1.30)	0.624 (+1.79)	0.627 (-0.63)	0.625 (+0.48)
MLP	0.620 (-1.43)	0.604 (-3.36)	0.701 (+9.02)	0.649 (+2.37)

Table 6: Performance of machine learning models trained on Twitter data using features 1 - 7 from Table 4

On average, accuracy changed by -0.18%, precision changed by -0.82%, recall changed by +2.90% and F1 score changed by +1.06%. The improvement in F1 score is negligible, but it is interesting to see how the additional features boosted recall and hurt precision by significant margins for the linear-kernel support vector machine and the multilayer perceptron classifier. Overall, this means that the best model trained using Twitter data was the multilayer perceptron classifier trained with all seven sentiment features.

In an attempt to understand what the classifiers learned from our data, we examined the importance given to the seven features in our random forest classifier. The importance of features is retrievable from the linear kernel support vector machine and random forest classifier and since the random forest classifier performed well, we believed that this data would provide some valuable insights. Figure 4 is the bar chart illustrating this data.

It is surprising to see how all features apart from emoji score are considered more or less equally important. As we hypothesized when examining the boxplots of these features, the score range and adjacent contrast features are the two most important, even if they are not by a large margin.

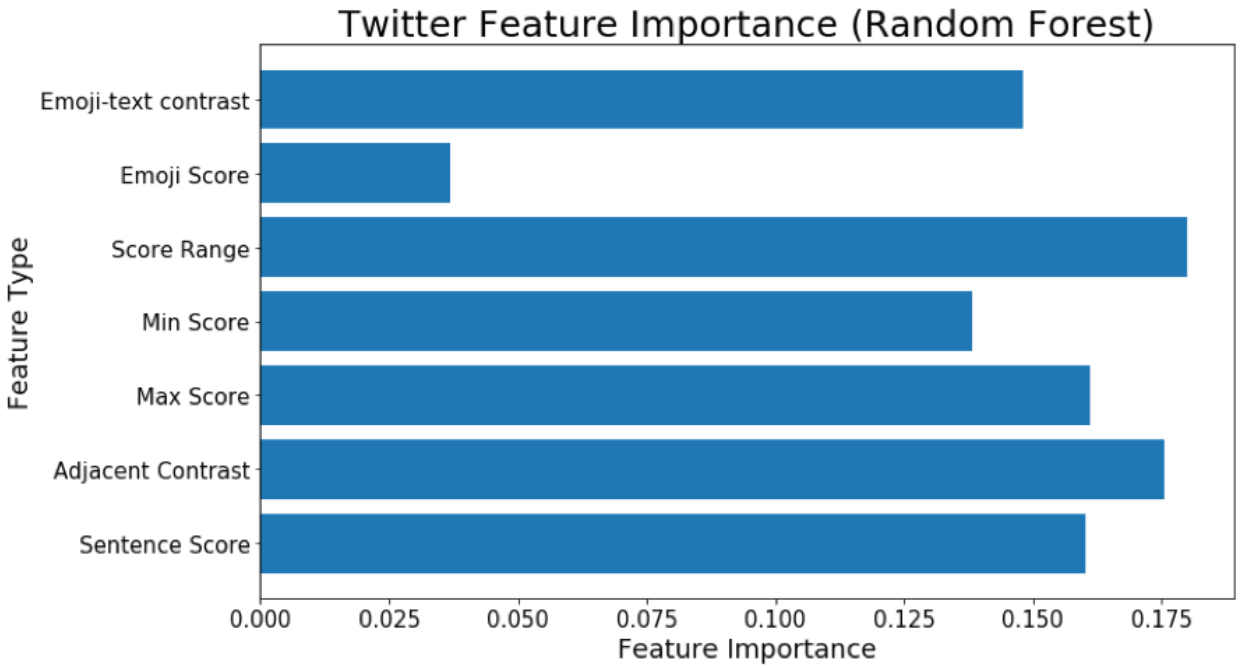

Figure 4: Importance of each feature in trained random forest classifier

5.2. Reddit Dataset

We follow the same line of analysis with Reddit data, but obtain very different results.

5.2.1. Statistical Analysis

Figure 5 is a boxplot of sentiment feature values obtained from the Reddit data. Table 12 in Section 8, the Appendix, also provides various statistics of the dataset.

Unlike the boxplot of Twitter data, we notice that the boxplots for sarcastic and non-sarcastic comments are very similar for every feature. The adjacent contrast and score range features of sarcastic and non-sarcastic tweets differed by half the standard deviation. With Reddit data, the means only differ by roughly a tenth of the standard deviation. This is a very stark difference given that our approach is the same, and only the dataset has changed.

5.2.2. Machine Learning Classification Models

Table 7 presents the performance of the classifiers with five features excluding the parent score and parent-text contrast features. Table 8 presents the performance of the models trained with all seven features.

Given that a random classifier obtains an F1 score of 0.5, the performance of most of these models

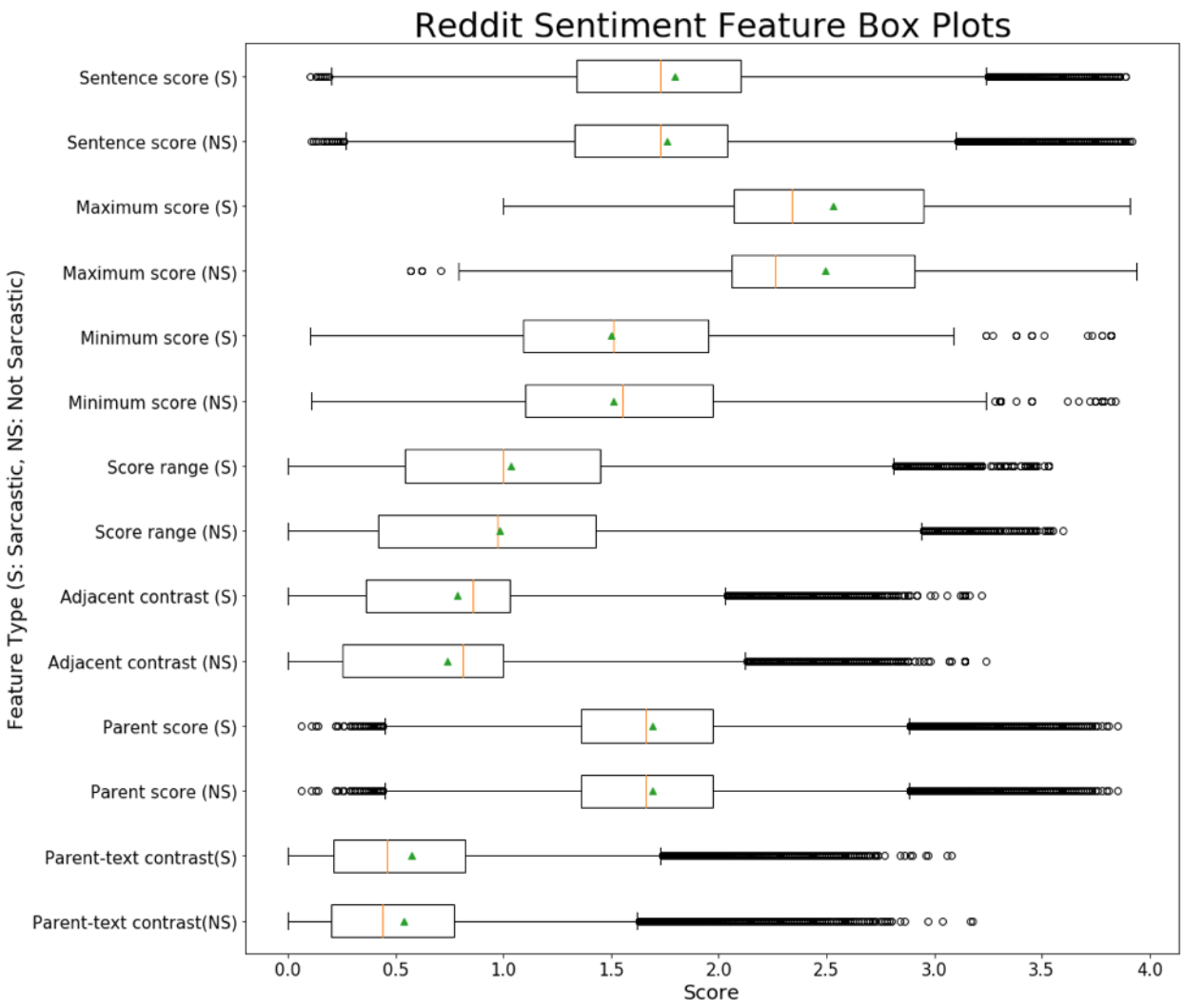

Figure 5: Boxplot illustrating the statistical distribution of Reddit sentiment features. The red line represents the median and the green triangle represents the mean. Circles outside of the boxplot's whiskers are outliers.

are disappointing. On average, the addition of features 8 and 9 changed accuracy by +0.248%, precision by 0.362%, recall by -2.39% and F1 score by -0.696%. The fact that the two additional

Model	Accuracy	Precision	Recall	F1 score
SVM (Linear)	0.523	0.546	0.271	0.362
SVM (RBF)	0.529	0.521	0.712	0.602
Gaussian Process	0.527	0.523	0.601	0.559
Random Forest	0.517	0.517	0.510	0.514
MLP	0.542	0.532	0.696	0.603

Table 7: Performance of machine learning models trained on Reddit data using features 1 - 5 from Table 4

Model	Accuracy (%change)	Precision (%change)	Recall (%change)	F1 score (%change)
SVM (Linear)	0.522 (-0.19)	0.541 (-0.92)	0.282 (+4.06)	0.371 (+2.49)
SVM (RBF)	0.538 (+1.70)	0.533 (+2.30)	0.61 (-14.33)	0.569 (-5.48)
Gaussian Process	0.521 (-1.14)	0.517 (-1.15)	0.623 (+3.66)	0.565 (+1.07)
Random Forest	0.532 (+2.90)	0.532 (+2.90)	0.526 (+3.14)	0.529 (+2.92)
MLP	0.531 (-2.03)	0.525 (-1.32)	0.637 (-8.48)	0.576 (-4.48)

Table 8: Performance of machine learning models trained on Twitter data using features 1 - 5, 8, 9 from Table 4

Reddit-specific features made little impact and negatively influenced the F1 score illustrates that we failed to capture any form of sarcasm with these features.

The two models with the best performance are the RBF kernel support vector machine and multilayer perceptron classifier, both trained with only five features. They both achieve an F1 score of just over 0.6, but it is concerning that the precision and accuracy are very low in both cases. If having a lot of false positives is not an issue, these models may be attractive choices, but otherwise the models do not show great performance.

5.3. Comparison of Twitter and Reddit Models

The average F1 score of the models trained and tested with Twitter data was 0.610. The average was 0.525 with Reddit data. Therefore, the Twitter models were 16% better on average than Reddit models. Given that the same basic approach was implemented to both datasets, this difference suggests that there is a fundamental difference in the way sarcasm is expressed on the two platforms. The two platforms are very different in their design, purpose and userbase, so it is not surprising that our approach had different performances on the two datasets.

We had hoped to capture platform-specific ways sarcasm is conveyed by engineering platform-specific features. Twitter's characteristics as a social media platform were captured by analyzing emojis and Reddit's characteristics as a discussion platform were captured by examining the relationship between the comments and their parent posts. However, neither of these features made a significant impact on performance. For the Twitter dataset, we still believe that emojis are important features that must be considered, and it is likely that the lack of and quantity of tweets

with emojis in our dataset was the reason why results were different from what we expected. As for Reddit, it is possible that discussions contain less emotive language than social media posts, rendering our approach less effective.

The most performant classifiers were RBF kernel support vector machines and multilayer perceptron classifiers. Support vector machines have been widely adopted in previous research due to their good performance [6, 15] and neural networks are gaining popularity [16, 4]. Their good performance on both datasets is more evidence that the two classifiers are effective in natural language processing tasks.

5.4. Comparison with state-of-the-art models

We have chosen Riloff et al. and Joshi et al.'s results as the benchmark to evaluate our models. This is because both teams also incorporate sentiment features as their most important, if not their only, features for training their models. Table 9 and Table 10 show how the results obtained from our best models compare with the results reported in these two research papers. It should be noted that each group used different datasets to obtain these results, so these comparisons should be taken with a grain of salt.

Approach	F1 score
Riloff et al.	0.51
Joshi et al.	0.61
Twitter MLP (7 features)	0.65

Table 9: Comparison of F1 scores of Twitter models [6, 15]

Approach	F1 score
Joshi et al.	0.640
Reddit MLP (5 features)	0.603

Table 10: Comparison of F1 scores of discussion forum models. Joshi et al. used data from a discussion forum different from Reddit.[6]

On a dataset of tweets, our multilayer perceptron model trained on all seven sentiment features outperforms Riloff et al. by 27% and outperforms Joshi et al. by 7%. On the discussion forum dataset, we were short by 6% to Joshi et al.'s state-of-the-art performance. Given that there is still plenty of room for improvement in our approach, these are very encouraging results.

6. Conclusion & Future Work

Our approach to sarcasm detection exceeded the state-of-the-art performance on Twitter data and fell slightly short on discussion forum data. We adopted a previously tested method of modeling sarcasm as being expressed through a juxtaposition of contrasting sentiments but implemented this idea in a novel way. We introduced a way to use Socher et al.'s sentiment analysis model [16] to generate features that can capture contrasting sentiments of phrases in a sentence. By making a conscious decision to only use sentiment features to train our models, we have shown how effective sentiment analysis can be in predicting sarcasm in text. The same approach was taken to detect sarcasm in tweets and reddit posts and the noticeable difference in performance suggests that sarcasm is expressed in different ways depending on the environment. We hypothesize that users use less emotive language on discussion forums like Reddit compared to social media platforms like Twitter.

There are a couple of questions our research fell short of answering that we hope to address in the future. First, the lack of tweets with emojis in our dataset made it difficult to extract any meaning from the emoji sentiment features we engineered. We would like to test our method using a larger dataset of tweets to find what role emojis play in expressing sarcasm on social media platforms. Given the short time we had to complete our research, there was a limited number of tweets that could be streamed into our dataset. Secondly, we would like to further analyze how sarcasm takes different forms in different platforms. Our results only show that there exists a difference between Reddit and Twitter but offer little insight into exactly how sarcasm is expressed differently on the two platforms. Our initial hypothesis is that social media platforms are more suited to expressing personal emotions whereas discussion forums are less personal and more focused on sharing opinions about topics of common interest. A study comparing the most frequently used n-grams in sarcastic tweets and sarcastic Reddit comments may be able to qualify how different language is used on the two platforms to convey sarcasm.

We relied heavily on Socher et al.'s sentiment analysis model [16] to break down sentences into

phrases and to attach sentiment labels to these phrases. This means that the performance of our model is dependent on the performance of the underlying sentiment analysis model and we believe our approach can be improved in this area. Since Riloff et al.'s bootstrapping algorithm [15] is effective in learning positive and negative sentiment phrases, future work may be able to combine the two approaches to identify positive and negative sentiments in phrases more effectively. Our approach and its performance will certainly benefit from using a more refined sentiment analysis model to generate the sentiment features.

7. Acknowledgements

I would like to thank Professor Fellbaum and fellow members of the Natural Language Processing Seminar for their invaluable guidance and feedback throughout the research process.

References

[1] "A large self-annotated corpus for sarcasm," 2017. [Online]. Available: https://arxiv.org/abs/1704.05579

[2] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *in Proc. of LREC*, 2010.

[3] E. Camp, "Sarcasm, pretense, and the semantics/pragmatics distinction," vol. 46, no. 4, 2012, pp. 587–634.

[4] B. Felbo *et al.*, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," 2017.

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

[6] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," vol. 50, no. 5. ACM, 2017, p. 73.

[7] A. Joshi, V. Sharma, and P. Bhattacharyya, "Harnessing context incongruity for sarcasm detection," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, vol. 2, 2015, pp. 757–762.

[8] M. Khodak, N. Saunshi, and K. Vodrahalli, "A large self-annotated corpus for sarcasm," 2017.

[9] P. Kralj Novak *et al.*, "Sentiment of emojis," *PLOS ONE*, vol. 10, no. 12, pp. 1–22, 12 2015. Available: https://doi.org/10.1371/journal.pone.0144296

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

[11] C. Liebrecht, F. Kunneman, and A. van Den Bosch, "The perfect solution for detecting sarcasm in tweets# not." New Brunswick, NJ: ACL, 2013.

[12] C. D. Manning *et al.*, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[13] D. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis," in *LREC 2014 Proceedings*. ELRA, 2014.

[14] A. Reyes, P. Rosso, and T. Veale, "A multidimensional approach for detecting irony in twitter," vol. 47, no. 1. Springer, 2013, pp. 239–268.

[15] E. Riloff *et al.*, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 704–714.

[16] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

8. Appendix

Feature Name (S: sarcastic, NS: not sarcastic)	Mean	Std. Dev.	Skewness
Sentence score (S)	1.95	0.52	0.52
Sentence score (NS)	1.84	0.51	0.53
Maximum score (S)	2.88	0.58	0.12
Maximum score (NS)	2.65	0.56	0.66
Minimum score (S)	1.40	0.42	0.013
Minimum score (NS)	1.50	0.42	-0.21
Score range (S)	1.49	0.67	0.15
Score range (NS)	1.16	0.66	0.29
Adjacent contrast (S)	1.11	0.52	0.20
Adjacent contrast (NS)	0.88	0.51	0.20
Emoji score (S)	2.06	0.24	3.09
Emoji score (NS)	2.09	0.33	2.74
Emoji-text contrast (S)	0.46	0.37	1.18
Emoji-text contrast (NS)	0.50	0.40	1.43

Table 11: Statistical overview of sentiment features from Twitter dataset

Feature Name (S: sarcastic, NS: not sarcastic)	Mean	Std. Dev.	Skewness
Feature (S: sarcastic, NS: not sarcastic)	Mean	Std. Dev.	Skewness
Sentence score (S)	1.80	0.59	0.66
Sentence score (NS)	1.76	0.55	0.61
Maximum score (S)	2.53	0.50	0.75
Maximum score (NS)	2.49	0.50	0.85
Minimum score (S)	1.50	0.43	-0.24
Minimum score (NS)	1.51	0.43	-0.28
Score range (S)	1.03	0.62	0.42
Score range (NS)	0.98	0.66	0.48
Adjacent contrast (S)	0.79	0.50	0.36
Adjacent contrast (NS)	0.74	0.52	0.45
Parent score (S)	1.69	0.45	0.66
Parent score (NS)	1.69	0.45	0.66
Parent-text contrast (S)	0.57	0.46	1.13
Parent-text contrast (NS)	0.54	0.43	1.18

Table 12: Statistical overview of sentiment features from Reddit dataset