SARCASM DETECTION IN REDDIT COMMENTS

Ryan Adoni[1], Glen Farbanish Jr.[1]

[1]Stevens Institute of Technology

radoni@stevens.edu, gfarbani@stevens.edu

ABSTRACT

As most communication happens over the internet in the form of written media, it is imperative that readers are able to fully understand, not only the content of a message or piece of text, but the tone and intention of what they are reading. Oral communication is more expressive than written media as tone can be conveyed through loudness, inflection, and physical cues. Specifically, sarcasm is one such facet of oral communication often lost in written form—can sarcasm be detected to a high degree using natural language processing? It is already well known that punctuation and context play important roles in detecting sarcasm in text, therefore this paper leverages real-world data from Reddit. Moreover, the primary goal of this paper is to contribute to the detection of sarcasm in text to mitigate the lack of inflection written text has compared to speech. Investigating the amount of context needed to detect sarcasm and the effects of cleaning data has, this paper sets out to answer how sarcasm can be detected and to what degree. Using BERT and DistilBert, this paper compares and contrasts these models on sarcasm detection and uses an ensemble model capable of achieving higher average accuracy than any of the models individually. Building upon previous methodologies, this paper hopes its findings will further research in the field of sarcasm detection and sentiment analysis.

1 Introduction

A common problem of written text is the loss of tone and inflection when compared to speech. While spoken words carry loudness, inflection and physical cues, written text lacks the ability to fully communicate advanced semantic abstractions such as sarcasm. Plainly speaking, Miriam-Webster defines sarcasm to be "the use of words that mean the opposite of what you really want to say especially in order to insult someone, to show irritation, or to be funny" [8]. Sarcasm can be seen in many different place in written media, one such being online forums. Here is a real-world example of sarcasm from Reddit [12]—an online internet forum commonly used for news, social discussions, and web content rating:

Table 1: Sarcasm Example from Reddit [9]

Comment 1	Have you tried addressing what makes you feel that way?
Comment 2	No, I've just been drifting through life not trying to solve any of my problems.

This example comes in the form of two comments. The first comment is a simple question asking whether someone has ever attempted to fix why they feel a certain way. The second comment is a direct response to the first comment falsely—and sarcastically—claiming that the commenter has never actually attempted to solve any of their problems. This is false and the tone and context in which the commenter says this makes the reader infer they are not actually serious and are joking. For more examples and explanation for how sarcasm can be seen on the internet, read section 3 and Table 2.

While this can be apparent to some users on the spot, sarcasm can be difficult to pick up on if one is not looking for it, which can lead to misunderstandings.

This problem plagues internet forums and common solutions have sprung up to try and lessen miscommunications. Regarding sarcasm, Reddit utilizes a user-added tag, '/s,' to denote if a post contains sarcasm.

While this can be extremely helpful, user-added tags can be incorrectly applied and abused. Would it not be helpful to automatically detect sarcasm in a piece of text and label it as such?

Sentiment analysis is already a large field in natural language processing (NLP) for quantifying and studying subjective texts. This can be labeling text as "positive," "negative," or "neutral" or in the case of this paper, "sarcastic" or "not sarcastic." In an effort to help reader comprehension of tone within written media, specifically real-world media, such as internet forums, this paper aims to predict whether a short piece of text is "sarcastic" or "not sarcastic."

Research in the field of sentiment analysis as it relates to sarcasm detection has already been well documented, so carefully examining past research is a necessary first step in improving previous methods and building a successful classifier.

2 Background

Lots of research has been done regarding sentiment analysis and there are a large amount of papers regarding the detection of sarcasm in text. Much of the work that has occurred surrounding the investigation of sarcasm detection can be classified into three main groups: semi-supervised approaches such as [2], supervised approaches such as [1], and rule-based approaches such as [14]. Many approaches also cast a wide net by training multiple models and seeing which ones yield the best results [11] or using all the models in an ensemble; ensemble approaches have also been used in conjunction with rule-based approaches to detect sarcasm such as [14] and [10].

A lot of data for detecting sarcasm has been compiled from Twitter. For example, Bouazizi et al. [1] approaches sarcasm detection by utilizing data from Twitter and supervised learning that learns "based on the part-of-speech (PoS) of words" used as well as taking into account punctuation-related features such as the number of exclamation and question marks, dots, capitalized words, and quotes within a specific piece of text. Rappoport et al. [2] also makes use of data from Twitter and punctuation-related features, but employs a semi-supervised approach; even between different methods, there is a lot of overlap between the ideas being used in detecting sarcasm. Since real data can be full of non-standard English, Prasad et al. [11] makes use of slang and emoji dictionaries to specifically train multiple models—random forest, gradient boost, decision tree, as well as others—to be aware of this, which noticeable improve their models. Motivation for this idea comes from the fact that a standard bag-of-words approach will fail to understand the negative nature in the following quote if the emoji is ignored: "I'm so pleased mom woke me up with vacuuming my room this morning. :)" [11]. This toy example captures the importance of modeling emoticons and other punctuation usually removed in standard preprocessing as this quote could very easily be classified differently with the emoticon ":)" removed from preprocessing. Using sentiment analysis, it is likely that this example would be classified as positive because of the positive denotations of the word "pleased," however if the ":)" emoticon is not removed, a model may be able to capture the sarcastic nature associated with using an emoticon in this linguistic setting and then correctly classify this quote as negative.

Sarcasm is an extremely high level abstraction in language and therefore requires advanced sentiment analysis algorithms to be able to be detected effectively as discussed previously. It is necessary to evaluate models using raw and unprocessed text as the smallest punctuation in a sentence can make a difference when attempting to classify a piece of text as "sarcastic" or "not sarcastic." This motivation leads directly into the data set that will be used in this paper as it is extremely raw, unprocessed, and from real internet interactions on Reddit.

3 Data Set

The data set this paper will be working with [9] can be found on Kaggle and was collected and annotated by researchers at Princeton [5]. The data set has balanced and imbalanced, i.e., the true distribution, versions. For this paper's experiments, the balanced data set will be used, which has 1,010,827 data points where each point consists of 10 features: label, comment, author, subreddit, score, upvotes, downvotes, creation time, UTC creation time, and parent comment. For this paper's purposes, only the label, comment, and parent comment features will be used. The parent comment is the first comment asked; this provokes the child comment. The label represents whether the child comment is "sarcastic" (denoted by a 1) or "not sarcastic" (denoted by a 0). The label distribution is approximately equal, consisting of 505,443 comments labeled as "sarcastic" and 505,384 labeled as "not sarcastic."

Moreover, this data set contains themes and text with non-standard linguistic features such as caps, italics, and elongated words. e.g., "Yeahhh, I'm sure THAT is the right answer." Lastly, the data set contains comments with "subjective determinant, racism, conditionals, sentiment heavy words, 'Internet Slang' and specific phrases" [9]. While this data set contains many challenges, it models real conversations and comments, allowing for better generalizations for real-world applications.

Two example data points from this data set can be seen in Table 1—one with a "sarcastic" label (denoted by 1) and one with a "not sarcastic" label (denoted by 0). Recall, the parent comment initiates the response from the child comment and the labels measure the sarcasm in the child comment, not the parent comment.

Table 2: Example Data Points

	Parent Comment	Child Comment	Label
Sample 1	Yahoo discloses hack of 1 billion accounts	3 years later and they finally disclose this?	0
Sample 2	That's the best part about our generation. We have the internet and youtube videos to help us with almost anything.	Almost no parenting needed!	1

As clearly shown in Table 2, sample 1 exhibits a parent comment where a user shares some recent news about Yahoo disclosing a data leak that occurred three years ago; the child comment is a non-sarcastic remark directly responding to the parent comment about that news, questioning why they would release this information so late. Similarly, sample 2 shows a child comment wittily—and sarcastically—responding to the parent comment as clearly parenting is still needed in today's age even if the "internet and youtube videos" are extremely helpful. Hopefully, these examples better make sense of the data set being used and illustrate what sarcasm on online forums is like.

As emphasized in the background, this data set is extremely unclean. The uncleaned data will be used as input to the models used in this paper and while cleaned data will be produced to measure the effects of cleaning the data, this paper expects the uncleaned data to be of more use as this is what prior research indicates.

4 Methods

As mentioned before, the goal of this paper is to solve sarcasm detection. To do this, research was conducted on sentiment analysis and previous findings with sarcasm detection to better understand the problem. Originally, simple classifiers were going to be constructed using RNNs and CNNs with different architectures such as LSTM, GRU, Bidirectional LSTM, and attention to gain an understanding how these models perform. However after looking at a few of the most popular attempts on Kaggle, these investigations were already looked into [7]. Using Bidirectional LSTM, [7] achieved the best performance with 70.37% accuracy on test data. Therefore, more advanced models are investigated in this paper to achieve a higher performance.

For sarcasm detection, pre-trained BERT models fine-tuned on the Reddit sarcasm data are used to solve this binary classification problem. In order to obtain the best models, four sets of factors are tested. The first is different input number of tokens (comment length). Due to large RAM requirements and expensive training times, finding the best comment length while still staying within the bounds of computational power is essential. How long of an input can the models afford? The next is cleaned input vs uncleaned input. Do numbers and punctuation help the models? The third factor is the added context of the parent comment. Does the additional context improve the models' performances? Lastly, what types of BERT architecture, whether BERT or DistilBERT have significant advantage over others. Which BERT model performs the best for sarcasm detection? After all of these factors are explored, the best results will determine the models that are trained to be used in an ensemble in hopes of boosting performance.

5 Experimental Design

In order to solve sarcasm detection BERT models are being used. BERT, which stands for Bidirectional Encoder Representations from Transformers, is an advanced NLP machine learning framework based on transformers. The BERT model originated from the model described in "Attention Is All You Need" [18]. In this paper, the model is composed of 12 transformer layers with 6 identical encoder layers and 6 identical decoder layers. Researchers took the encoder transformer layer from the Attention paper and stacked it to create the BERT model. BERT consists of 12 identical transformer layers as seen in Figure 1. At the start of every input sequence there is a [CLS] token. This token is used for classification in classification tasks. Additionally, BERT has the [SEP] token which separates and marks the ends of sentences. The

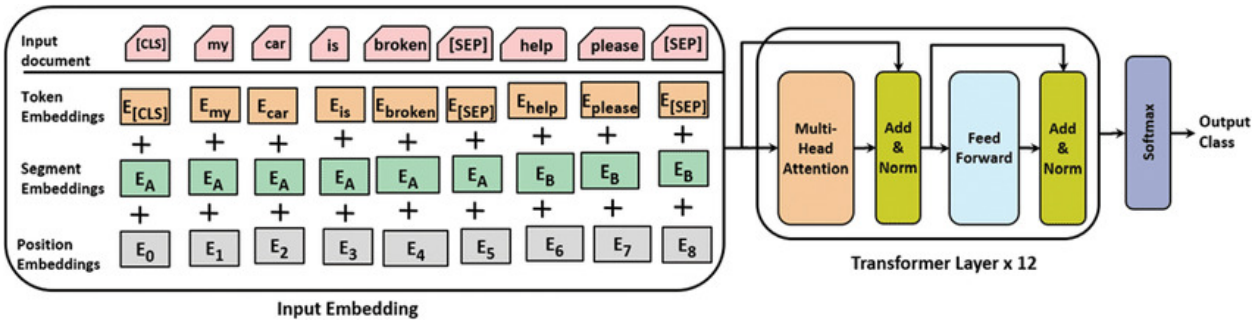


Figure 1: BERT Fine Tuning Classification Architecture [4]

last layer of BERT is a fully connected layer with a softmax activation function. As a result, categorical cross entropy is used. BERT base version features 110 million trainable parameters [3].

Alternatively, there exists the DistilBERT model. DistilBERT is a smaller, faster, and cheaper version of BERT. DistilBERT is 40% smaller and 60% faster than BERT while still retaining 97% of BERT's performance. DistilBERT is trained using knowledge distillation in which the smaller student model, DistilBERT, is trained to mimic the performance of a larger teacher model, BERT [13]. Since DistilBERT eases the computational requirements, this model will be used to explore comment length, prepossessing data, and parent context to see the effects it has on sarcasm detection accuracy.

Accuracy is defined as being binary and can defined as such where i and j are the ground truth and predicted class labels respectively, "sarcastic" or "not sarcastic" and can be defined easily as such:

$$Accuracy = \delta_{ij} = [i = j] \tag{1}$$

where the Kronecker delta function from (1) is defined in the usual way:

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{2}$$

This paper, then, defines average accuracy to be the total sum of the accuracy values for all data points in the test set divided by the the total number of data points in the testing set. Taking $\mathcal{T}$ to represent the test set where each value in $\mathcal{T}$ is a triple (p, c, g, l), where p is the parent comment, c is the child comment, g is the ground truth value for the label of "sarcastic" or "not sarcastic," for c, and l is the predicted value for the label of c. Average accuracy is, therefore, defined as such:

$$Average\ Accuracy = \frac{\sum_{(p,c,g,l)\in\mathcal{T}} \delta_{gl}}{|\mathcal{T}|} \tag{3}$$

The sarcasm detection code this paper refers to consists of three notebooks: Data Generation, Modeling, and Ensembling. The code was run on Google Colab Pro accounts to increase computational performance. The data generation file is where the input data is shuffled and converted into jsonl files. The Reddit sarcasm data is read in and parent comment, child comment, and label data fields are kept. A 70% training, 20% validation, and 10% testing split is created with approximately 700,000 training entries, 200,000 validation entries, and 100,000 testing entries. A cleaned version of this dataset is created where punctuation, numbers, and urls are removed. These two sets of data are then saved to be used by the next notebook.

The model code is where the BERT models are created and fine-tuned. Using the Hugging Face library, we are able to import BERT [15] and DistilBERT [17] models. From these classes, a pre-trained tokenizer, model, and task type can be chosen. BERT uses the "bert-base-uncase" pre-trained model and distilBERT uses the "distil-base-uncased" pre-trained model. Additionally, some trials were run using the DistilBERT "distil-base-cased" pre-trained model. Both use sequence classification as the task type. The class used is the Trainer API for fine-tuning the BERT models [16]. Here is where most of the parameters were initialized. The Adam optimizer is used with a learning rate of 5E-5. All models are fine-tuned on 4 epochs with a batch size of 16. Every 20,000 batches, the models are evaluated on the validation set and the model weights are saved. This allows the best model to be used after fine-tuning is completed.

The model notebook can fine-tune a number of different models by setting the proper flags, paths, and variables. The first factor tested using DistilBERT is comment length. The sizes: 20, 32, 64, 96, and 128 are tested to determine which offer the best performance while staying in the bounds of system limitations (See Figure 2). Next, eight DistilBERT models are run varying between parent vs. only child, unclean vs. clean, and length 64 vs. length 96 to see how each factor affects average accuracy (See Figure 3). Lastly, BERT vs. DistilBERT is tested to see how the different models vary in average accuracy (See Figure 4). Each DistilBERT model takes approximately 4.5 hours to run and each BERT model takes 10 hours to run. After the models finish fine-tuning, they are saved to be used for ensemble.

The last notebook, ensemble, is where the fine-tuned models are loaded to see if running certain combinations of models can result in improved performance. From here, precision and recall are also calculated.

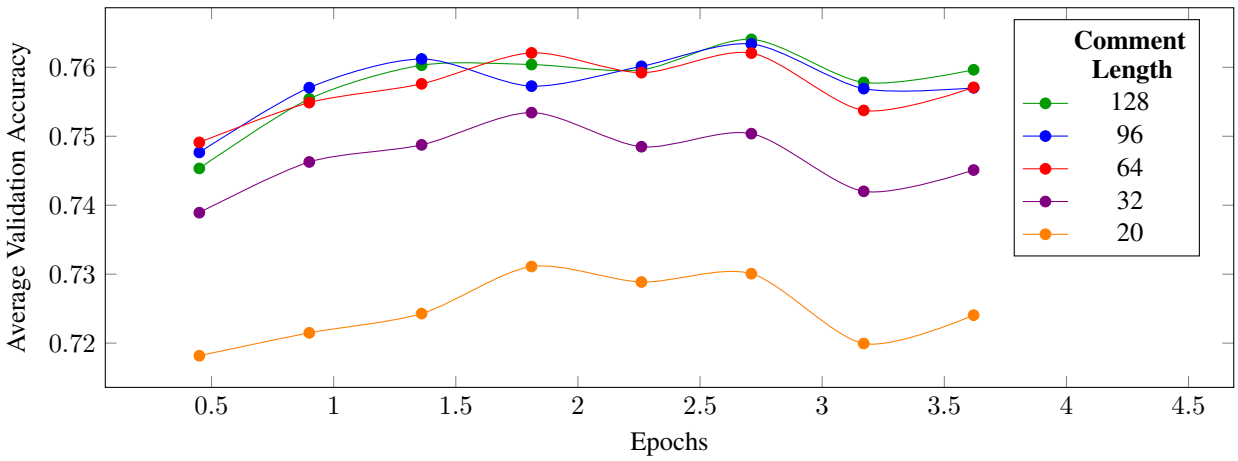
6 Experimental Results

Since the data is labeled, the main evaluation metrics will be testing and validation accuracies calculated by evaluating the testing and validation data sets on the trained models. Additionally, the top models will be combined in ensemble approaches for further improved evaluation. Because sarcasm may be more recognizable by humans, we will look at specific examples the models misclassify after training to recognize shortcomings and seek possible improvements.

In the experiment for determining the best comment length size, the sizes looked at are 20, 32, 64, 96, and 128 on DistilBERT Unclean Parent models. Larger sizes such as 192 were attempted but RAM limitations prevented these models from being run. The average length of the uncleaned parent comments are 24.04 tokens and the average of the uncleaned child comments are 10.36 tokens. Looking at Figure 2, the validation accuracies at checkpoints vs. epochs are plotted. At each checkpoint the weights are saved, thus the highest average accuracy on the graph is saved as the best result. Length 20 was the worst by a significant amount with length 32 also showing low validation accuracies. These results are likely caused by some of the longer comments needing to be truncated to fit into the tokenizer. The lengths 64, 96, and 128 all had approximately the same best results with 128 being the best by a slim margin. Diminishing results are returned with sizes 64 and up. This is likely caused by the models taking in all or enough context to determine if the comment is sarcastic or not. Since 64 and 96 had similar results to 128, they will be used for the next experiment to save computation time.

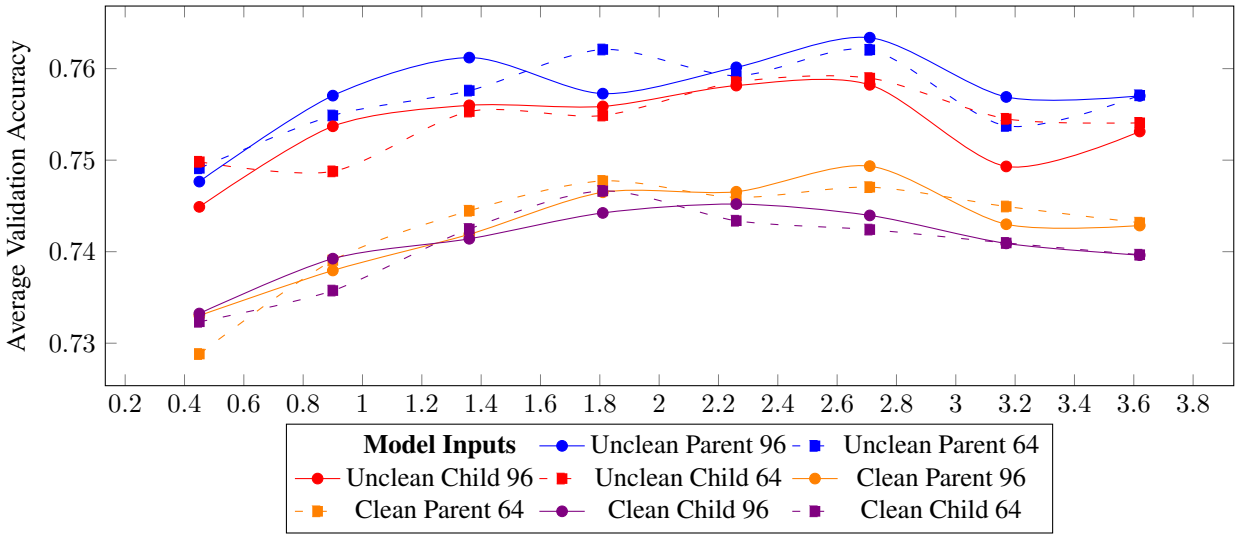
The next experiment conducted looks at context and data preprocessing. On DistilBERT models three factors are looked at. The first is context with some models including the parent comment and others only having the child comment. The next factor is data preprocessing with some models having uncleaned data passed in and others having cleaned data passed in. The last factor is further insight on comment length with 64 and 96 being tested. Looking at Figure 3, the validation accuracies at checkpoints vs. epochs are plotted. The most noticeable divide in the graph is preprocessing. All models with uncleaned data performed better than those with cleaned models. This is likely caused by the BERT models being able to handle punctuation and removing it gives the models less context to work with. The next finding is with parent

Figure 2: Average Validation Accuracy Versus Epochs for Varying Comment Lengths for DistilBERT



These models were all DistilBERT Unclean Parent Models.

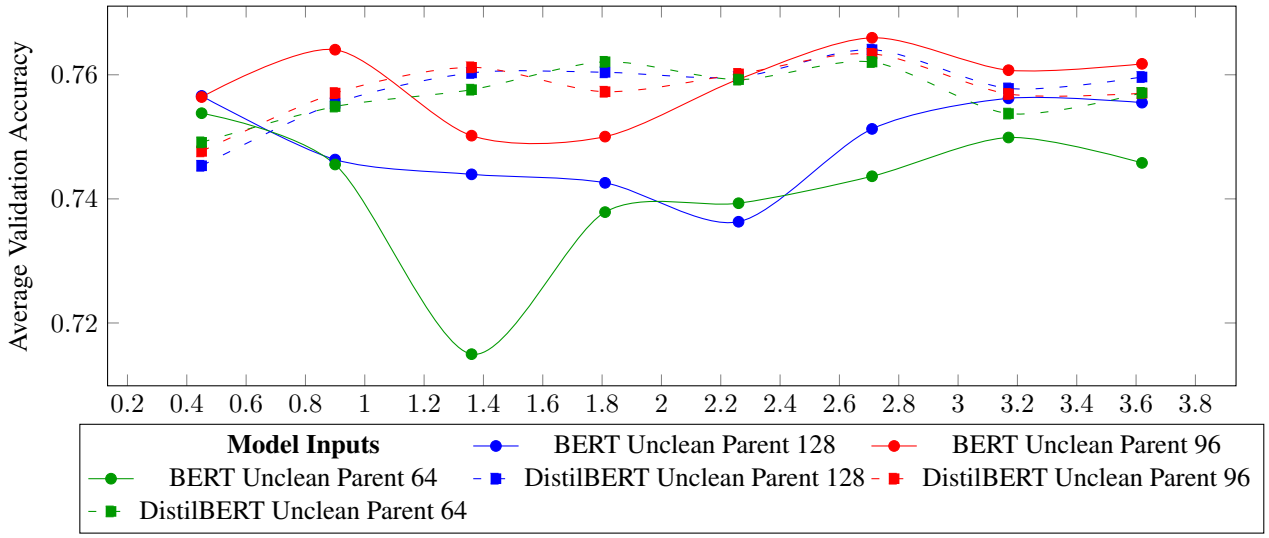
Figure 3: Average Validation Accuracy Versus Epochs for Varying Data Inputs for DistilBERT



vs. only child comments. In every pair of models, the parent context added models have a higher average accuracy than child only. This is again likely due to the added context the parent comments provide giving the models more insight on which label to predict. Lastly, the comment lengths have mixed results with certain pairs having 96 outperform 64 and vice versa. Overall, in the last experiment, uncleaned parent data will be used as these saw the best results.

The last experiment run is comparing the DistilBERT model to BERT model. The models are run with unclean parent data with the lengths 64, 96, and 128. Looking at Figure 4, the validation accuracies at checkpoints vs. epochs are plotted. Out of the two sets of models, DistilBERT with length 128 performed the best and BERT with length 96 performed overall the best with the highest average validation accuracy. More insights between the two models show DistilBERT is consistent while BERT models have high levels of randomness. While BERT 96 had the highest average validation accuracy, BERT 64 and 128 had the lowest. Other tests run with BERTs resulted in heavy overfitting occurring during fine-tuning with the model deprecating to random guess after the first epoch. Thus, while BERTs are capable or achieving higher performance they introduce large levels of randomness.

Figure 4: Average Validation Accuracy Versus Epochs Varying for BERT and DistilBERT Models



After running the experiments and finding the factors with the best results fifteen models were created. Now it is time to create different ensembles of the models. Eight ensembles are created. Ensemble 1 includes all the BERT models. Ensemble 2 consists of all DistilBERT models excluding the models with length 20 and 32 which had poor average validation accuracy. Ensemble 3 looks at all uncleaned data models excluding the DistilBERTs with length 20 and 32. Ensemble 4 looks at the models that use cleaned data. Ensemble 5 consists of the models with the best validation accuracies. Ensemble 6 consists of all models except DistilBERT with length 20 and 32. Ensemble 7 consists of all models. The ensembles use majority vote to select which label to predict. If an ensemble contains an even number of models, the weights are reweighted to have the most complex model break ties. The tiebreak models are denoted by ⋆'s in the ensemble while the other models included have ✓'s. The models and ensembles are evaluated on the test set with average accuracy, precision, and recall reported. The results can be seen in Table 3. The best model performance seen was DistilBERT Uncleaned Parent 64. The average test accuracy is 0.807 which is significantly higher than the other models. This model is considered an outlier given its unusually high performance. Ensembling the models had mixed results. For some ensembles, improvements were seen having a higher average test accuracy than each of the individual models as seen by Ensemble 1. Other ensembles appeared to average the testing accuracies of the models seen with the ensembles including the DistilBERT outlier. The precision and recalls across the models are equivalent such that no model overly predicts one class. To test to see the effects the outlier had on the ensemble results, Ensemble 8 was created with all models excluding the outlier model. Ensemble 8 has an average test accuracy of approximately 0.800 which is still above all the combined models. Overall, these results of models and ensembles show significant improvement over the previous attempts using simpler architectures.

Table 3: Individual Model and Ensemble Metrics

	BERT Unclean Parent 64	*BERT Unclean Parent 96*	*BERT Unclean Parent 128*	*Distil. Clean Child 64*	*Distil. Clean Child 96*	*Distil. Clean Parent 64*	*Distil. Clean Parent 96*	*Distil. Cased Unclean Parent 64*	*Distil. Unclean Child 64*	*Distil. Unclean Child 96*	*Distil. Unclean Parent 20*	*Distil. Unclean Parent 32*	*Distil. Unclean Parent 64*	*Distil. Unclean Parent 96*	*Distil. Unclean Parent 128*	*Avg. Accuracy*	*+ Precision*	*- Precision*	*+ Recall*	*- Recall*
Ensemble 1	✓	✓	✓													.789	.800	.779	.772	.806
Ensemble 2				✓	✓	✓	✓	✓	✓	✓			✓	✓	⋆	.801	.809	.793	.789	.813
Ensemble 3	✓	✓	✓					✓	✓	✓			✓	✓	✓	.802	.803	.800	.800	.803
Ensemble 4				✓	✓	✓	⋆									.779	.811	.754	.730	.829
Ensemble 5		✓	⋆					✓	✓	✓			✓	✓	✓	.802	.804	.800	.799	.804
Ensemble 6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	.804	.812	.796	.791	.816
Ensemble 7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	.802	.814	.791	.785	.820
Ensemble 8	✓	✓	⋆	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	.800	.812	.788	.781	.818
Avg. Accuracy	.767	.785	.779	.792	.758	.795	.759	.780	.795	.795	.766	.767	.807	.777	.759					
+ Precision	.768	.805	.783	.811	.764	.804	.804	.765	.783	.791	.806	.788	.803	.770	.771					
- Precision	.766	.767	.776	.775	.751	.786	.722	.798	.808	.800	.735	.749	.812	.784	.748					
+ Recall	.767	.753	774	.762	.747	.781	.680	.811	.817	.804	.749	.732	.815	.791	.739					
- Recall	.767	.817	.784	.822	.768	.809	.834	.745	.773	.787	.831	.801	.799	.763	.780					

Note that for ensembles with an even number of models, a small value was added to the most complex model. The most complex model for even ensembles are marked with a ⋆ and not a ✓. Additionally, '+' represents "sarcastic" and '-' represents "not sarcastic."

7 Conclusion & Future Work

This paper has accomplished its goals of detecting sarcasm in text to mitigate the lack of inflection written text has compared to speech. This paper has also produced fantastic findings investigating the amount of context needed to detect sarcasm and the effects of cleaning data. Using BERT and DistilBert, this paper compares and contrasts these models on sarcasm detection and constructs an ensemble model capable of achieving approximately 80% average accuracy, achieving and surpassing the goals of this paper. While this average accuracy can most likely be improved, there are significant problems with the data set which will be discussed in the coming paragraphs as they relate to future work.

The authors of this paper hope it will be useful to others and that the attached code can be adapted for other data generation, fine-tuning, and ensembling tasks that can be applied for future research in the field of sarcasm detection and sentiment analysis.

Moreover, a summary of this paper's findings is that there is a noticeable increase in average accuracy when models are fine-tuned using unclean data rather than clean data. Additionally, it was found that token length or comment length has an effect on the performance of the models and found that for this data set a comment length for 64 was adequate to achieve the best results for the least amount of fine-tuning. Further, it was found that the BERT architecture yielded more varied results than the DistilBERT models and context always helped the models differentiate "sarcastic" from "not sarcastic." Ensembling models increased performance unless there was an outlier and overall none of the models over-predicted on a specific class. Cased models did not perform better than uncased models, however extensive testing was not done similar to the other experiments conducted—this is a potential future area of research.

While achieving relatively high average test accuracy over 80%, this paper finds many avenues for future work: larger and more complex models, more hyper-parameter tuning, better metrics and methodologies for selecting models for an ensemble, examining other syntactic patterns, and cleaner data.

Larger and more complex NLP models have generally been proven to solve problems more accurately, so it makes sense as a future line of research to attempt to detect sarcasm using larger models to see if higher performances can be reached. Different architectures for BERT could also be attempted such as RoBERTa [6] as only two architectures were attempted in this paper.

Further, while much care was taken in tuning the hyper-parameters to achieve the best results, the authors of this paper did not have unlimited amounts of time nor resources to tune every hyper-parameter independently, so while some were tuned individually and those results are displayed in the figures 2 to 4, a more holistic approach is taken to determine the effects of different hyper-parameters. In future works, a more rigorous approach could be taken if more time and resources were available.

Additionally, better metrics and methodologies could be attempted to find out what models should go into the ensemble rather than just grouping based on type and average validation accuracy results.

Finally, different syntactic patterns can be examined. Discussed in the coming paragraphs, some sarcasm can come from misspellings so examining syntactic patterns can be a future line of research. Tokenizing with emoticons is also a great idea and one that could use more research as sometimes these are indicators of sarcasm [11].

The largest future line of work this paper uncovers is cleaner data. This paper finds significant issues with the data set being used [9] and believes better data to be a strong future line of research. While, unclean data clearly provides a boost in accuracy, this paper finds some significant issues and believes some selective cleaning could improve results. Looking at Table 4, there are three main issues found with the data set that the authors believe hindered accuracy: spelling mistakes, non-English comments, and misclassification of labels. Cleaner data could be compiled in future research, as while this data set does provide advantages in being unclean, the old adage goes: "garbage in, garbage out."

Primarily, while this paper shows how punctuation and grammar can denote sarcasm, and while noting that some misspellings, such as elongated words, do indicate sarcasm like "reeeeallly" or "yeeeeah, this paper proposes that most sarcastic misspellings can be detected differently than actual misspellings—most sarcastic misspellings have many repeated letters. While, this paper does not make use of character recognition to differentiate between sarcastic misspellings and true misspellings, this can be a future research interest. Most likely true misspellings should be corrected to proper English to remove unnecessary data obfuscation. However, because this is not applied in this experiment, performance and, most likely, average accuracy suffers because of the extra tokens needed from misspelled words. An example of gross misspellings can be seen in Table 4 anomaly 1.

Further, when scanning the data, non-English comments were found. Comments in Spanish and Dutch (an example appears in Dutch as anomaly 2 of Table 4) appear more often than expected, which is to say more than zero. Because of the extra tokens needed, this, like the problem of misspellings, not only adds extra complexities into the models, but increases

Table 4: Example Data Anomalies

	Parent Comment	Child Comment	Anomaly
Anomaly 1	Tender... I just... Eh no	u kno u like dem chicken tenders do amirite?	Spelling Mistakes
Anomaly 2	gelukkig zijn die scandinaviers allemaal zo anti-confrontatie, ze zullen zichzelf nooit durven verdedigen helemaal niet tegen onze lange en sterke landgenoten!	Volgens duitsland gaf scandinavie meer weerstand.	Non-English
Anomaly 3	Florida man walks out of his house with a semi-automatic rifle and a four-pack of beer and begins shooting at passing cars	He was clearly intimidated by the cars and was merely standing his ground.	Misclassification

tokens and reduces performance. Since most sarcasm is language specific, models should only be trained using data from one language or if more than one language was being trained as an experiment, it is likely many more comments in other languages would be needed, unlike this data set, which only has relatively sparse examples of non-English comments.

Moreover, the worst issue found with this data, is misclassifications in the data set. Take anomaly 3 from table 4. The child comment clearly indicates sarcasm, however, the ground truth label is 0, meaning "not sarcastic." Even worse, ensemble 7 from table 3, predicts this child comment to be of label 1, meaning "sarcastic." While this should be classified as correct and show how the models are correctly predicting, it incorrectly hurts the evaluation metrics.

While misclassifications and other anomalies are bound to happen in large data sets, a cleaner and more refined data set is a future line of research as while syntactic patterns contribute to sarcasm detection, there are many selective cleaning methods that can be applied: removing true misspellings, removing non-English comments, and fixing misclassifications.

While many avenues of research still exist for this problem, this paper is quite content with an average accuracy above 80% as this outperforms every top-rated model using this data set to detect sarcasm the authors of this paper have found on Kaggle by approximately 4% [9].

References

[1] Mondher Bouazizi and Tomoaki Otsuki Ohtsuki. A pattern-based approach for sarcasm detection on twitter. *IEEE Access*, 4: 5477–5488, 2016. URL `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7549041`.

[2] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon, 2010. URL `https://aclanthology.org/W10-2914.pdf`.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL `http://arxiv.org/abs/1810.04805`.

[4] Shaheen Khatoon, Majed Alshamari, Amna Asif, Md Maruf Hasan, Sherif Abdou, Khaled Elsayed, and Mohsen Rashwan. Development of social media analytics system for emergency event detection and crisis management. *Computers, Materials and Continua*, 68:3079–3100, 03 2021. doi: 10.32604/cmc.2021.017371.

[5] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A large self-annotated corpus for sarcasm, 2017. URL `https://arxiv.org/abs/1704.05579`.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[7] Fahad Mehfooz. Saracsm eda//bidirectional/cnn/logistic/deployment. `https://www.kaggle.com/fahadmehfoooz/saracsm-eda-bidirectional-cnn-logistic-deployment`, 2021. Accessed: 2021-12-13.

[8] Merriam-Webster. Sarcasm. `https://www.merriam-webster.com/dictionary/sarcasm`, ... Accessed: 2021-12-13.

[9] Dan Ofer. Sarcasm on reddit, 2018. URL `https://www.kaggle.com/danofer/sarcasm?select=test-balanced.csv`. 2021-11-03.

[10] Pradeesh Parameswaran, Andrew Trotman, Veronica Liesaputra, and David Eyers. Detecting target of sarcasm using ensemble methods. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 197–203, Sydney, Australia, 2019. Australasian Language Technology Association. URL https://aclanthology.org/U19-1027.

[11] Anukarsh G. Prasad, S. Sanjana, Skanda M. Bhat, and B. S. Harish. Sentiment analysis for sarcasm detection on streaming short text data. In *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, pages 1–5, 2017. URL https://ieeexplore.ieee.org/document/8169892.

[12] Reddit. Reddit. https://www.reddit.com/, ... Accessed: 2021-12-13.

[13] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

[14] Karthik Sundararajan and Anandhakumar Palanisamy. Multi-rule based ensemble feature selection model for sarcasm type detection in twitter. *Hindawi*, 2020. URL https://www.hindawi.com/journals/cin/2020/2860479/.

[15] Hugging Face Transformers. Bert. https://huggingface.co/docs/transformers/model_doc/bert, 2019. Accessed: 2021-12-09.

[16] Hugging Face Transformers. Trainer. https://huggingface.co/docs/transformers/main_classes/trainer, 2019. Accessed: 2021-12-09.

[17] Hugging Face Transformers. Distilbert. https://huggingface.co/docs/transformers/model_doc/distilbert, 2020. Accessed: 2021-12-09.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.