AUTOMATIC SARCASM DETECTION USING A MULTI-FEATURE FRAMEWORK WITH BERT

By:- Team 4 – Susheel Patel, Dikshant Gupta, Xingxing Huang

INTRODUCTION

The importance of sarcasm detection lies in its ability to help us better understand the intentions of the speaker. Misinterpreting sarcasm can lead to misunderstandings and even conflicts, particularly in situations where there are cultural or linguistic differences. In the context of social media, sarcasm detection is particularly important due to the sheer volume of content that is generated on these platforms. With millions of tweets and other posts being generated every day, automated sarcasm detection systems can help identify potentially harmful or offensive content and prevent it from spreading (Vyas & Uma, 2022).

One of the key challenges in sarcasm detection is the importance of context. Sarcasm is often conveyed through tone of voice or nonverbal cues, which are difficult to capture in written communication. As a result, sarcasm detection algorithms need to consider the context in which the text is being used. This includes not only the specific words being used but also the broader social and cultural context in which the communication is taking place.

For example, consider the following tweet: "I just love sitting in traffic for hours on end #not." Without context, this tweet could be interpreted as a genuine statement of love for traffic. However, in the broader context of the tweet, the use of the hashtag #not indicates that the speaker is being sarcastic. Therefore, to accurately detect sarcasm in this tweet, it is essential to consider the context in which it was made.

Traditional machine-learning approaches to sarcasm detection often rely solely on content-based features such as sentiment polarity and word frequency. Various feature

engineering methods such as the N-gram, Bag-of-words and word embeddings have been previously utilized to identify sarcasm (Prasad et al., 2017). However, a common limitation among these studies is that they overlook the context present in the text (Eke et al., 2020). This results in a loss of relevant information for the model as importance is only given to the frequency of words. For instance, consider the sentence, "That's just what I needed today - a flat tire." A content-based approach may identify this sentence as negative due to the negative sentiment associated with the phrase "flat tire." However, a context-based approach would recognize the sarcasm in the statement, given the broader context of the speaker's frustration with the tire. Unlike static embeddings such as Word2Vec and Topic2Vec, BERT generates contextualized word embeddings where the vector for a word is determined based on its surrounding context in the sentence (Baruah et al., 2020). Our study aims to utilize BERT in conjunction with lexical, sentiment and context-based features to detect sarcasm.

The key contributions of this study are as follows:

- The study uses the Multi-Feature Model in conjunction with contextualized word embeddings using BERT to detect sarcasm and address the above-mentioned limitation. Diverse features are extracted including lexical, sentiment, context-based features (BERT), length of tweets, and part of speech tags to enhance model performance.
- Conventional Machine and Deep Learning Models such as XGBoost, SVM, Logistic Regression and LSTM are applied in two stages by first considering only semantic features in the first stage and fused features in the second stage to classify sarcasm.

The rest of the report is arranged as such: Section 2 reviews past work done on sarcasm detection, Section 3 provides details on the dataset used, Section 4 describes the methodology and approach followed, and Section 5 contains the result and conclusion of the paper and Section 6 contains the future scope of this study.

LITERATURE REVIEW

Sarcasm detection has been a research topic in natural language processing for more than a decade. Previous studies on sarcasm detection were mainly focused on content-based methods. (Reyes et al., 2012) proposed a model that considered features representing different patterns in short online texts, including ambiguity, polarity, unexpectedness, and emotional content. The study collected an evaluation corpus of 50,000 texts from Twitter. The result showed that the model had valuable insights into figurative usages of language and could be used for content-based sarcasm detection in social media. In addition to a rule-based classifier that was used to identify instances of positive sentiment contrasted with a negative situation, (Riloff et al., 2013) also used a set of patterns, specifically positive verbs and negative situation phrases as features to improve the performance. The findings indicated that identifying contrasting contexts through the phrases learned via bootstrapping enhanced the recall for sarcasm recognition. (Moores & Mago, 2022) stated that sarcasm can be categorized into three types based on the surface sentiment and intended sentiment: a positive surface sentiment with a negative intended sentiment, a negative surface sentiment with a positive intended sentiment, and a neutral surface sentiment with a negative intended sentiment. Similarly, (González-Ibáñez et al., 2011) created a corpus combining three categories, which were sarcastic, positive and negative tweets. The sarcastic category depended solely on users' judgement since it was obtained from user-annotated tags on sarcasm. The study concluded that most sarcastic tweets were negative with positive messages but aimed to express a negative attitude. The results also highlighted that using lexical features alone was insufficient for sarcasm detection.

Some other works have utilized word n-grams, which are sequences of words that often appear together, to find patterns of sarcasm in text. For example, (Barbieri & Saggion, 2014) derived new word frequency-based features that can help detect irony. (Tungthamthiti et al., 2014) divided each tweet into three types of features, which were a single word, a sequence of two words and a sequence of three words. Additionally, the weights of N-gram features are binary: 1 indicated the presence of N-gram, while 0 indicated its absence. The result of the experiments showed that including N-gram features along with all proposed features, such as contradiction, sentiment, and punctuation, improved accuracy by 3% compared to using only N-grams.

In addition to these initial content-based approaches to sarcasm detection, the researcher began to explore the use of classifier ensembles and machine-learning algorithms to improve the accuracy of sarcasm detection in social media. (Sarsam et al., 2020) performed a systematic review on sarcasm detection using machine learning algorithms on Twitter. The review considered publications until November 2018. The study revealed that support vector machine (SVM) was the most commonly used algorithm, accounting for 22.58% of the reviewed literature, followed by Logistic Regression Method at 19.35%. For instance, (da Silva et al., 2014) introduced classifier ensembles formed by Multinomial Naive Bayes, SVM, Random Forest, and Logistic Regression to classify the sentiment of tweets automatically. The experiments used the emoticons in the tweets to enrich feature sets and compared the results of a different combination of bag-of-words (BoW), feature hashing (FH), and lexicons. The results showed that classifier ensembles using bag-of-words could provide higher classification accuracy than stand-alone classifiers. However, the study did not compare their

approach with other state-of-the-art methods for tweet sentiment analysis or other methods proposed in the literature, such as deep learning models, feature engineering techniques or sentiment lexicon adaptation methods. (Bagate & Suguna, 2022) experimented with models including XGBoost, Logistic Regression, Ransom Forest, and LSTM and found XGBoost classifier outperformed other classifiers with #sarcasm present. Additionally, after removing hashtags from tweets, the XGBoost classifier was still offering comparatively good results. In general, previous approaches failed to capture the context and complexity of the presented text and thus could not detect different forms of sarcasm.

Transfer learning approaches for various tasks in natural language processing have been increasingly popular due to context's importance and data annotation's difficulties. To use transfer learning for sarcasm detection, one approach is to fine-tune a pre-trained language model on a dataset of sarcastic and non-sarcastic sentences, while another is to use a pre-trained language model to generate embeddings for each sentence in the sarcasm detection dataset. Pre-trained embeddings such as Global Vectors (GloVe), Word2Vec, and FastText are commonly utilized among these methods. Moreover, deep learning models such as Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) are often utilized for their ability to extract features automatically, making them popular choices in automatic sarcasm detection. For instance, (Mandal & Mahto, 2019) proposed an architecture for news headline sarcasm detection using deep learning techniques. The architecture comprised an embedding layer, a CNN and a bidirectional LSTM network. The CNN and bidirectional LSTM extracted relevant features and classified the news headlines. The proposed

model achieved an accuracy of 86.16%, which was considered optimal for the given dataset. However, the limitation of this study was the lack of utilizing pre-trained word embeddings or other pre-trained models, which could lead to more advanced architectures and higher accuracy.

Furthermore, (Onan & Tocoglu, 2021) proposed a sarcasm identification framework with two phases: term-weighted word embedding and stacked three-layer bidirectional LSTM processing of text documents. The study suggested that pre-trained word embeddings can enhance the performance of deep-learning models for sarcasm identification. Similarly, (Son et al., 2019) utilized GloVe and bidirectional LSTM for word representation to detect sarcasm in user-generated content and achieved sarcasm-classification accuracy of 97.87% for the Twitter dataset and 93.71% for the random-tweet dataset. (Khatri & P, 2020) also proposed using BERT and GloVe embeddings as features and concluded that word embeddings are robust for improving accuracy in automatic sarcasm detection. BERT and GloVe embeddings represent the meaning of words as vectors, making it easier for the model to understand the context and underlying sentiment of a statement.

In addition to utilizing word embeddings, multi-task learning has emerged as a promising approach in sarcasm detection, offering improved performance and efficiency that involves training a model to perform multiple tasks simultaneously. (Majumder et al., 2019) proposed a novel multi-task learning architecture that jointly aimed to improve sentiment classification and sarcasm detection. A single neural network was used to execute the sentiment classification and sarcasm detection. The network took an input sentence and assigned it both a sentiment tag (positive/negative) and a sarcasm tag

(yes/no). The input sentence is represented as a matrix using GloVe word embeddings, and a gated recurrent unit (GRU) with an attention mechanism is used to obtain sentence representation from the matrix. Lastly, the fused representation is obtained using a neural tensor network (NTN) that fuses sarcasm and sentiment-specific sentence representations. The result indicated that the multi-task learning approach enhanced performance on both tasks compared to an independent model. Furthermore, (Hagen et al., 2022) introduced an intra-model attention-based adversarial deep multi-task framework for complaint detection that considers sarcasm, sentiment, and emotion in addition to complaint identification. The model included two networks: a task-specific network and a shared feature network. The task-specific network is responsible for predicting the output for each task. In contrast, the shared feature network is responsible for extracting features from the input data and sharing across all tasks. The result demonstrated that the proposed adversarial multi-tasking framework yielded better performance for the primary task.

In recent years, there has been an increasing trend toward incorporating fused features, deep learning, and transformer-based methods. One of the most notable recent works on detecting sarcasm is by (Eke et al., 2021b). A fusion framework was proposed, combining multiple features extracted from Twitter posts to improve the performance of machine learning models like SVM, Random forest, KNN, Logistic regression and Decision tree models (Eke et al., 2021b). The proposed approach tried to improve upon limitations mentioned in past literature by addressing two key issues: the context of words and data sparsity in expression. Firstly, the framework extracted a comprehensive set of features, including lexical, length of microblog, hashtag, discourse

markers, emoticon, syntactic, pragmatic, semantic and sentiment-related features to address the context of words. Additionally, the framework employed a novel feature extraction algorithm and a two-stage classification algorithm. The first classification stage utilized lexical features only, extracting from the BoW technique and training with the five classifiers mentioned earlier. The second stage involved the fusion of lexical, sentiment, and context. The proposed approach offered a promising solution for automatically detecting sarcasm and can be applied to improve performance.

Furthermore, (Eke et al., 2021a) focused on using context embedding to consider both local and global context information to construct deep learning and BERT model features. The GloVe embedding method was used to create a word representation that captured semantic information. A deep learning model based on Bi-LSTM automatically identifies sarcasm using context information. Meanwhile, a feature fusion technique that included BERT, hashtag, sentiment-related, syntactic, and GloVe embedding features is used to evaluate two benchmarks Twitter datasets. The result demonstrated that the proposed technique outperformed the baseline methods for sarcasm detection. Another recent study discussed the problem of sarcasm detection in social media and how it can be misleading. (Kaya & Alatas, 2022) proposed a new hybrid deep neural model using CNN + BiLSTM and BERT models to detect sarcasm in a sarcastic and non-sarcastic news headline dataset. The result showed that the BERT model performed the best. The study also compared the performance of the CNN + BiLSTM Model with and without GloVe embeddings and found that the model without GloVe embeddings provided better results.

Recently, (Savini & Caragea, 2022) proposed a transfer learning framework for sarcasm detection from textual inputs. The study used the BERT pre-trained language model as the base model and fine-tuned it on three different datasets: the Sarcasm V2 Corpus, the Self-Annotated Reddit Corpus (SARC), and the SARCTwitter dataset for sarcasm detection, with different sizes and characteristics. The authors also experiment with fine-tuning the BERT model on intermediate tasks (fine-grained emotion detection, coarse-grained sentiment polarity, and sentiment classification of movie reviews) before fine-tuning the model on the sarcasm detection task. The intermediate task transfer learning models, especially with the EmoNet sentiment detection dataset (Abdul-Mageed & Ungar, 2017) show improvement over the vanilla BERT model on the SARCTwitter dataset. This is because the EmoNet intermediate models are trained from the same social media domain as SARCTwitter and are rich in polarized emotions, which helps improve the performance of the sarcasm classification task. The study also highlights the importance of sentiment in sarcasm classification and that using BERT with intermediate task transfer learning improves model performance when the dataset size is small. The BERT models that only used message content performed better than models using additional information from a writer's history. The authors established new state-of-the-art results for sarcasm detection, achieving an F1-score of 97.43% on the SARCTwitter dataset, which is an improvement of almost 11% over similar previous works.

DATASET

The dataset for detecting sarcasm was collected by fetching sarcastic and non-sarcastic tweets from Twitter. Twitter is a popular microblogging and social media platform where people, organizations and media outlets share information, express opinions, and engage with others in real-time. The Twitter Application Programming Interface (API) v2, launched in November 2021, is a set of protocols that allow users to extract data from Twitter, including tweets, user profiles and metadata. The Twitter API v2 provides several endpoints, which are specific URLs that return data based on the parameters provided (Twitter API documentation). The dataset was constructed by fetching almost equal amounts of sarcastic and non-sarcastic tweets. For obtaining the labels for each post, any tweet that contained the tag “#sarcasm” or “#sarcastic” was labelled as “Sarcastic”, and tweets that are tagged as do not contain sarcasm tags were labelled as “non-sarcastic”.

Table 1: Attributes Description fetched from Twitter API V2

Attribute	Description
Tweet text	The actual text of the tweet is limited to 280 characters.
Tweet ID	A unique identifier for each tweet is used to reference the tweet within the API and elsewhere.
User information	Information about the user who posted the tweet, including the username, name, location, and profile image.
Timestamp	The date and time the tweet was posted, including the time zone. The date and time are not relevant to the task.

Tweet metadata	Information about the tweet, including the number of likes, retweets, and replies, as well as any hashtags, mentions, and URLs included in the tweet.
Geolocation	The location of the tweet, if available, includes the latitude and longitude.
Language	The language in which the tweet was written is determined by Twitter's language detection algorithms.
Tweet entities	Information about any entities mentioned in the tweet, including mentions of other users, hashtags, URLs, and more.
Retweets and replies	Information about any retweets or replies related to the tweet, including the original tweet's ID, the user who posted the retweet or reply, and the text of the retweet or reply.

While the above-mentioned attributes are available, this study considers only the “Tweet Text” Attribute. This ensures that the approach can be applied to other domains and datasets outside Twitter. Also, the scope of the dataset is limited to the English language, and retweets (when the same tweet is re-posted by the user) have been excluded to prevent duplication in data. We also referenced the ARTK dataset used in the multi-feature fusion framework for sarcasm identification (Eke et al., 2021c) on Twitter data to evaluate our model. We merged the dataset fetched from Twitter with the ARTK dataset to train our model. The dataset is a balanced dataset with 54% sarcastic tweets and 46% non-sarcastic tweets. The total number of tweets combined is 30,000.

METHODOLOGY

The process of obtaining a data set from Twitter has the drawback of noise that comes along with it. Twitter data comes in various forms such as simple text, user mentions (@user), URLs, and hashtags (#) (Twitter API documentation). These data sets need to be pre-processed before the feature extraction and classification task. To remove noise from the data, retweets and duplicated tweets were removed. Additionally, filters were used to retain only English language tweets, and tweets with only URL content were removed. These noisy data do not contribute to the enhancement of classification accuracy and are, therefore, eliminated. After the noise is removed, the text data is pre-processed to prepare it for the classification task. The first step in this process is converting the text to lowercase. The next step is tokenization, which involves breaking the text into individual tokens or words. This process makes it easier to analyze the text and extract useful information from it. Stop word removal is another technique used in pre-processing. Stop words are commonly used words in a language such as “a”, “an”, “the”, etc. These words do not carry much meaning in the text and can be removed to make the analysis more efficient. Spell check is used to correct any spelling errors in the text. This process is necessary to ensure that the algorithm is not confused by any misspelled words that may change the meaning of the text. Lemmatizing involves reducing words to their base form using vocabulary and morphological analysis. This process ensures words are reduced to their correct root form, considering their part of speech. Part-of-speech (POS) tagging is used to identify the grammatical structure of the text. This is done to extract useful information about the text and to better understand the relationships between words in the text. POS tagging

is implemented using the Python library and Natural Language Processing (NLP) toolkit (NLTK: Natural Language Toolkit).

After preprocessing, feature engineering is done to define different features. Previous studies have relied on content-based features, for example, Bag-of-words features, in isolation for sarcasm detection without considering contextual features. Performance results obtained with content features revealed that these features alone are not sufficient to accurately capture all the sarcastic tendencies in the text (Kumar & Garg, 2019). To enhance the performance of the model, some comprehensive novel features have been proposed to augment the content features. Word embeddings representing the text in a vector form are generated using the BERT model (Devlin et al., 2019; Pennington et al., 2014). The BERT (Bidirectional Encoder Representations from Transformers) embedding model is a state-of-the-art pre-trained language model that has recently gained popularity for its superior performance in various natural language processing tasks. BERT uses a transformer-based architecture that learns contextualized embeddings by processing words in their left and right contexts simultaneously. Unlike GloVe, BERT is a context-based approach that learns the contextual relationships between words, thereby capturing the nuances of language better. BERT is pre-trained on large amounts of text data and can be fine-tuned for a variety of downstream tasks, including sarcasm detection in tweets. BERT has been shown to outperform traditional word embedding approaches like GloVe in various tasks, including sentiment analysis and named entity recognition, making it a promising choice for feature engineering in sarcasm detection in tweets. Since opposing sentiments in the same text can point to sarcasm, we utilize a sentiment lexicon,

“SentiWordNet” (Baccianella et al., 2010) to generate two features with a positive score and a negative score individually for each post. SentiWordNet is a sentiment lexicon that assigns sentiment scores to words based on their usage in context. It provides a numerical score for three types of sentiments: positivity, negativity, and objectivity. The scores range from 0 to 1, with 0 indicating no sentiment and 1 indicating the strongest possible sentiment. We extract features of positive and negative sentiment by subtracting the difference between positive sentiment and negative sentiment, then defining it into 1 and 0 based on whether it's greater or lesser than 0.5.

For the first half of the study, we only use sentiment features along with minor features like the length of the tweet. In the second half, we use word embeddings generated by BERT in addition to all of the previous features. We use classification models such as XGBoost, Logistic Regression, SVM and LSTM to classify the tweets based on features in the first half and then based on features in the second half. A brief summary on the chosen methods is as below:

Long Short-Term Memory (LSTM): LSTM is a kind of recurrent neural network (RNN) that is especially effective at capturing the sequential dependencies in text data, making it a popular option for natural language processing jobs. Additionally, it can handle variable-length inputs, which is crucial for tweets with variable lengths. LSTM is a suitable option for sarcasm detection since it is capable of accurately capturing the context and tone of the text. For the current implementation, we have utilized a Neural Network with 256 LSTM-cell hidden layer. The neural network layers are connected using a Rectified Linear Unit (ReLU), which helps avoid exploding gradient problem present with other methods (Gasparetto et al., 2022).

XGBoost: A well-liked ensemble method based on decision trees, XGBoost can handle categorical and continuous data. To increase accuracy, it functions by merging the predictions made by various decision trees. XGBoost has won several data science challenges, including those on the Kaggle platform, and is renowned for its quick training and prediction speeds (Gasparetto et al., 2022).

Support vector machine (SVM): SVM is a well-known machine learning method that works best for classification problems. It operates by locating a hyperplane that divides data into various classifications. High-dimensional data may be handled effectively by SVM, which can also capture intricate correlations between variables. It has been extensively used in sarcasm detection and other natural language processing tasks (Chih-Wei Hsu, Chih-Chung Chang, 2008).

Logistic Regression: Logistic Regression is a simple and powerful algorithm for binary classification tasks that models the probability of an event occurring based on input features. It is fast and interpretable and performs well on datasets where classes are linearly separable. In sarcasm detection, it can be effective in identifying important features by analyzing the coefficients of the model (Kantardzic, 2019).

The experiment's output is measured by performance evaluation measures known as evaluation metrics. These metrics assess the classification algorithm's output, and various measures such as accuracy, recall, precision, and f-measure are employed to evaluate the framework's performance. Each classifier's sarcasm identification potential is assessed when evaluated by these metrics. Classification accuracy (ACC) indicates the overall accuracy of the classification result, and it measures the true positive and true negative values attained by the classified instances over the whole instances.

Recall (REC) computes the sum of tweets accurately classified as sarcastic over the sum of sarcastic tweets. Precision (PRE) determines the number of tweets that have been correctly classified as sarcastic over the whole tweets that were classified as sarcastic. F-measure (F-M) is a performance evaluation that computes the harmonic mean of precision and recall and is used as the overall measurement of classifiers' performance as it considers precision and recall. F-M assumes the values of 0 and 1.

RESULTS AND CONCLUSION

RESULTS BASED ON ONLY SENTIMENT FEATURES

The results of the predictive performance analysis based on sentiment features are shown in Table 2, and a visual representation of these results can be seen in Figures 1,2,3 and 4. From the values presented in Table 2, it can be observed that the performance results for precision, recall, F-measure, and accuracy range from 61% to 65%. These values suggest that all classifiers used were able to comprehend sarcastic expressions based on sentiment features. The XGBoost classifier had the highest precision performance, with an overall score of 65.8%, outperforming all other classifiers in terms of F-measure, recall, and accuracy. The Deep Learning LSTM classifier also displayed good performance, indicating an understanding of sarcastic expressions.

However, the Logistic Regression and SVM classifiers had lower performance results, achieving precision scores of 61.4% and 61.1%, respectively. Having the lowest performance results does not imply a complete lack of understanding of sarcastic utterances by these classifiers, but rather indicates a relatively lower understanding of sarcastic expressions based on sentiment features.

Based on the results, it can be concluded that the XGBoost classifier's performance can be attributed to its ensemble properties. Figures 1,2,3 and 4 illustrate the dominance of the XGBoost classifier. The XGBoost model achieved the highest precision (65.8%), F-measure (65.9%), recall (65.8%), and overall accuracy (65.8%), surpassing the second-highest precision (61.1%) achieved by the support vector

classifier. In contrast, the LSTM classifier obtained the second-highest recall (63.1%), F-measure (63.9%), and overall accuracy (63.1%) among all classifiers. It is worth noting that sentiment features are content-based and may lack contextual information. To address this limitation, contextual features were fused with the lexical feature for classification.

Table 2: Evaluation Metric Results with Only Sentiment Features

Model	Accuracy	F1 Score	Precision	Recall
SVM	61.11667	61.22324	61.11667	61.15267
Logistic Regression	61.43333	61.65426	61.43333	61.48226
XGBoost	65.80000	65.92344	65.80000	65.83543
Deep Learning Model (LSTM)	63.192	63.959	63.161	63.192

RESULTS BASED ON CONTEXTUALIZED BERT EMBEDDINGS AND SENTIMENT FEATURES

The results of the predictive performance analysis based on sentiment features and BERT are shown in Table 3, and a visual representation of these results can be seen in Figures 1,2,3 and 4. From the values presented in Table 3, it can be observed that the performance results for precision, recall, F-measure, and accuracy range from 80% to 91%. This is a significant increase from the previous results which were obtained from just sentiment features. Using contextualized word embeddings has increased the performance of the model by almost 20% on average across all the classifiers. The

LSTM had the highest precision performance, with an overall score of 91.7%, outperforming all other classifiers in terms of F-measure, recall, and accuracy.

In the case of sarcasm detection using BERT embedding features, the input to the LSTM is a sequence of word embeddings that capture the contextual meaning of each word in a sentence. Each embedding is passed through a layer of LSTM cells, which maintain a state vector that captures the historical context of the sentence up to that point. The output of the LSTM is a prediction of whether the sentence is sarcastic or not. The reason why LSTM outperforms traditional machine learning algorithms like XGBoost, logistic regression, and SVM for this task is that these algorithms do not have the same ability to capture sequential patterns in data. While they can certainly be used for sarcasm detection, they are generally less effective at identifying the subtle nuances and contextual cues that are crucial for detecting sarcasm. Another reason why LSTM outperforms traditional machine learning algorithms is that it is capable of learning complex representations of data. This means that it can effectively capture the underlying structure of the data in a way that traditional algorithms may not be able to. In the case of sarcasm detection, this could mean that LSTM is better able to identify the subtle patterns and nuances that are indicative of sarcasm.

Thus, the result shows that contextualized word embeddings enhance the performance of each classifier.

Table 3: Evaluation Metric Results with Bert Embeddings + Sentiment Features

Model	**Accuracy**	**F1 Score**	**Precision**	**Recall**
SVM	80.29087	80.27604	80.29087	80.27644

Logistic Regression	82.94885	82.95135	82.94885	82.95001
XGBoost	87.62956	87.63885	87.62956	87.63729
Deep Learning Model (LSTM)	91.6800	91.6900	91.68000	91.72100

Figure 1: Comparison of Accuracy Result

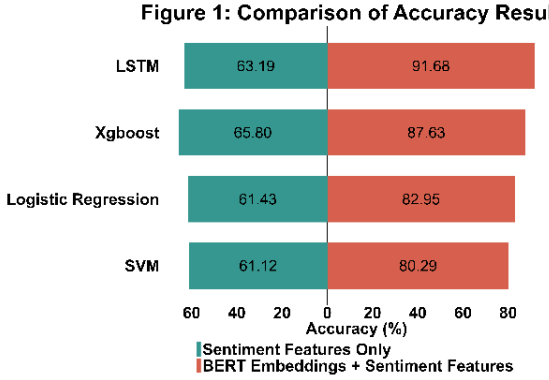


Figure 2: Comparison of F1 Score Result

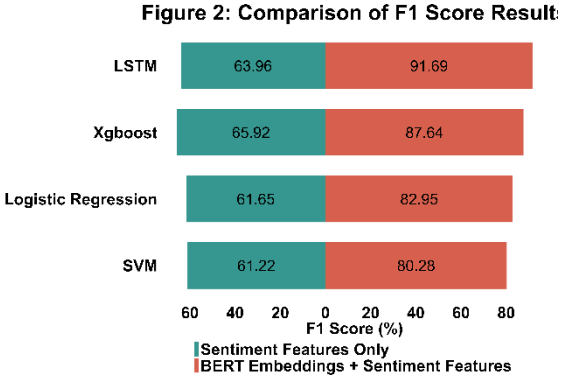


Figure 3:Comparison of Precision Result

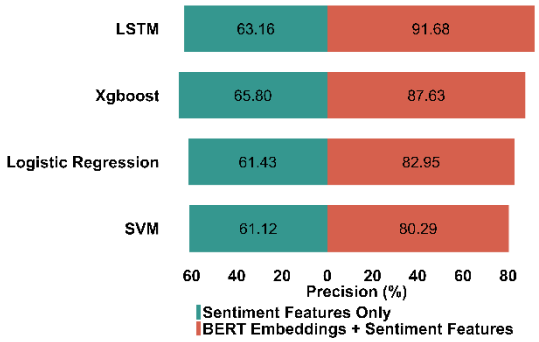
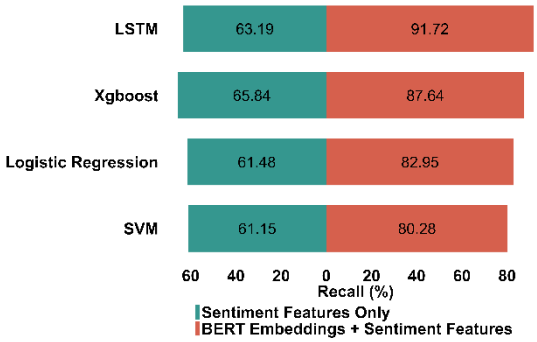


Figure 4:Comparison of Recall Results



FUTURE SCOPE

To improve the data quality for detecting sarcasm in Twitter data, a detailed spam filter can be implemented. This filter can filter out ads and non-relevant tweets from the dataset, leaving only the sarcastic tweets for analysis. By filtering out these irrelevant tweets, the overall quality of the data will be improved, allowing for more accurate detection of sarcasm. Another approach to improving the available context in word embeddings is to fine-tune BERT on Twitter data. BERT (Bidirectional Encoder Representations from Transformers) is a powerful pre-trained language model that can be fine-tuned for specific tasks, such as detecting sarcasm. By fine-tuning BERT on Twitter data, the model can learn to better understand the unique language and context of Twitter, leading to more accurate sarcasm detection. Finally, to generate larger word embeddings from BERT, more layers can be extracted. BERT consists of multiple layers, each of which generates a unique representation of the input text. By extracting more layers, larger and more detailed embeddings can be generated, which can improve the overall performance of the model. This approach can be particularly useful for detecting sarcasm, as sarcasm often relies on subtle nuances in language and context, which can be captured by more detailed word embeddings. Overall, these approaches can significantly improve the accuracy of detecting sarcasm in Twitter data by improving the quality of the data, increasing the available context in word embeddings, and generating more detailed embeddings.

REFERENCE

Abdul-Mageed, M., & Ungar, L. (2017). EmoNet: Fine-grained emotion detection with gated recurrent neural networks. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*. https://doi.org/10.18653/v1/P17-1067

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). SENTIWORDNET 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010*.

Bagate, R. A., & Suguna, R. (2022). Sarcasm Detection with and without #Sarcasm: Data Science Approach [Article]. *International Journal of Information Science and Management*, *20*(4), 1–15.

Barbieri, F., & Saggion, H. (2014). Modelling Irony in Twitter. *EACL 2014 - 14th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Student Research Workshop*. https://doi.org/10.3115/v1/e14-3007

Baruah, A., Das, K., Barbhuiya, F., & Dey, K. (2020). *Context-Aware Sarcasm Detection Using BERT*. https://doi.org/10.18653/v1/2020.figlang-1.12

Chih-Wei Hsu, Chih-Chung Chang, and C.-J. L. (2008). A Practical Guide to Support Vector Classification. *BJU International*, *101*(1).

da Silva, N. F. F., Hruschka, E. R., & Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles [Article]. *Decision Support Systems*, *66*, 170–179. https://doi.org/10.1016/j.dss.2014.07.003

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*.

Eke, C. I., Norman, A. A., Liyana Shuib, & Nweke, H. F. (2020). Sarcasm identification in textual data: systematic review, research challenges and open directions [Article]. *The Artificial Intelligence Review*, *53*(6), 4215–4258. https://doi.org/10.1007/s10462-019-09791-8

Eke, C. I., Norman, A. A., & Shuib, L. (2021a). Context-Based Feature Technique for Sarcasm Identification in Benchmark Datasets Using Deep Learning and BERT Model. *IEEE Access*, *9*, 48501–48518. https://doi.org/10.1109/ACCESS.2021.3068323

Eke, C. I., Norman, A. A., & Shuib, L. (2021b). Multi-feature fusion framework for sarcasm identification on twitter data: A machine learning based approach. *PLoS ONE*, *16*(6 June). https://doi.org/10.1371/journal.pone.0252918

Eke, C. I., Norman, A. A., & Shuib, L. (2021c). Multi-feature fusion framework for sarcasm identification on twitter data: A machine learning based approach. *PLoS ONE*, *16*(6 June). https://doi.org/10.1371/journal.pone.0252918

Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). Survey on Text Classification Algorithms: From Text to Predictions. *Information (Switzerland)*, *13*(2). https://doi.org/10.3390/info13020083

González-Ibáñez, R., Muresan, S., & Wacholder, N. (2011). Identifying sarcasm in Twitter: A closer look. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, *2*.

Hagen, M., Verberne, S., Macdonald, C., Seifert, C., Balog, K., Nørvåg, K., & Setty, V. (2022). Adversarial Multi-task Model for Emotion, Sentiment, and Sarcasm Aided Complaint Detection [Bookitem]. In *Advances in Information Retrieval* (Vol. 13185). Springer International Publishing AG. https://doi.org/10.1007/978-3-030-99736-6_29

Kantardzic, M. (2019). Data mining: Concepts, models, methods, and algorithms: Third edition. In *Data Mining: Concepts, Models, Methods, and Algorithms*. https://doi.org/10.1002/9781119516057

Kaya, S., & Alatas, B. (2022). Sarcasm Detection with A New CNN+BiLSTM Hybrid Neural Network and BERT Classification Model [Article]. *International Journal of Advanced Networking and Applications*, *14*(3), 5436–5443.

Khatri, A., & P, P. (2020). *Sarcasm Detection in Tweets with BERT and GloVe Embeddings*. https://doi.org/10.18653/v1/2020.figlang-1.7

Kumar, A., & Garg, G. (2019). Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-019-01419-7

Majumder, N., Poria, S., Peng, H., Chhaya, N., Cambria, E., & Gelbukh, A. (2019). Sentiment and Sarcasm Classification With Multitask Learning [Article]. *IEEE Intelligent Systems*, *34*(3), 38–43. https://doi.org/10.1109/MIS.2019.2904691

Mandal, P. K., & Mahto, R. (2019). Deep CNN-LSTM with word embeddings for news headline sarcasm detection. *Advances in Intelligent Systems and Computing*, *800 Part F1*. https://doi.org/10.1007/978-3-030-14070-0_69

Moores, B., & Mago, V. (2022). A Survey on Automated Sarcasm Detection on Twitter [Document]. *ArXiv.Org*.

Onan, A., & Tocoglu, M. A. (2021). A Term Weighted Neural Language Model and Stacked Bidirectional LSTM Based Framework for Sarcasm Identification. *IEEE Access*, *9*. https://doi.org/10.1109/ACCESS.2021.3049734

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.3115/v1/d14-1162

Prasad, A. G., Sanjana, S., Bhat, S. M., & Harish, B. S. (2017). Sentiment analysis for sarcasm detection on streaming short text data. *2017 2nd International Conference on Knowledge Engineering and Applications, ICKEA 2017*, *2017-January*. https://doi.org/10.1109/ICKEA.2017.8169892

Reyes, A., Rosso, P., & Buscaldi, D. (2012). From humor recognition to irony detection: The figurative language of social media [Article]. *Data & Knowledge Engineering*, *74*, 1–12. https://doi.org/10.1016/j.datak.2012.02.005

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., & Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*.

Sarsam, S. M., Al-Samarraie, H., Alzahrani, A. I., & Wright, B. (2020). Sarcasm detection using machine learning algorithms in Twitter: A systematic review [Article]. *International Journal of Market Research*, *62*(5), 578–598. https://doi.org/10.1177/1470785320921779

Savini, E., & Caragea, C. (2022). Intermediate-Task Transfer Learning with BERT for Sarcasm Detection [Article]. *Mathematics (Basel)*, *10*(5), 844. https://doi.org/10.3390/math10050844

Son, L. H., Kumar, A., Sangwan, S. R., Arora, A., Nayyar, A., & Abdel-Basset, M. (2019). Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2899260

Tungthamthiti, P., Shirai, K., & Mohd, M. (2014). Recognition of sarcasm in tweets based on concept level sentiment analysis and supervised learning approaches. *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation, PACLIC 2014*.

Vyas, V., & Uma, V. (2022). Approaches to Sentiment Analysis on Product Reviews. In *Research Anthology on Implementing Sentiment Analysis Across Multiple Disciplines*. https://doi.org/10.4018/978-1-6684-6303-1.ch011