Sarcasm Detection in Social Media Texts

Ahmed Farazi
Dept. of Computer Science
University of Calgary
Calgary, Canada
30063552

Fysal Beauferris
Dept. of Computer Science
University of Calgary
Calgary, Canada
30048436

Hafsa Zia
Dept. of Computer Science
University of Calgary
Calgary, Canada
30076662

Abstract – **This paper presents a study aimed at developing an alternative model for sarcasm detection. The objective is to achieve higher accuracy ratings on the SARC dataset by modifying the hybrid model proposed in our main paper and utilising the Sarcasm Corpus V2 dataset. The study employs GloVe, FastText, and BERT algorithms for creating word embeddings and text classification, along with SVC and Random Forest Classifier for analysis. Python libraries including Pandas, Numpy, Matplotlib, Seaborn, Spacy, Transformers, Tensorflow, Keras, and Sklearn are used for preprocessing, analysis, and visualisation. The FastText algorithm breaks input text into subwords, and each subword is represented as a vector. Our approach aims to assist individuals and researchers responsible for manually categorising text as sarcasm or non-sarcasm. The performance of our model is evaluated against the hybrid model on the SARC and Sarcasm Corpus V2 datasets, where the SARC dataset has been previously tested against the hybrid model proposed in the main paper.**

Keywords – Sentiment Analysis; Sarcasm Detection; GloVe; FastText; BERT; Random Forest; SVC.

I. INTRODUCTION

Social media has become an essential part of our daily lives, with millions of individuals using these platforms to express their opinions and thoughts. Unfortunately, this has led to an increase in abusive content, such as hate speech, cyberbullying, and harassment. As a result, it is crucial to develop automated detection systems to monitor and flag such content to maintain a safe and healthy virtual environment. One such detection system is sentiment analysis, which involves identifying and categorising opinions expressed in text as either positive, negative, or neutral.

The primary goal of this project is to enhance the detection of sarcastic sentiments in social media texts, which can lead to a betterment of sentiment analysis. Sarcasm is a unique form of communication in which a speaker says something but means the opposite, often to convey humour, criticism, or irony. Detecting sarcasm is already a challenging task for humans, and it becomes even more complex when given to machines to identify. One factor that makes sarcasm detection difficult is that it often depends on context and cultural knowledge. Another factor is that it can involve a contradiction between the literal meaning of the words used and the speaker's intention.

This project utilises FastText instead of Word2Vec for generating word embeddings, along with BERT and GloVe. We employ SVC and Random Forest Classifier for text classification to analyse social media texts and develop a more accurate sarcasm detection system. To preprocess, analyse, and visualise the data, we incorporate Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, Spacy, Transformers, TensorFlow, Keras, and Sklearn.

The findings of this project can assist researchers and individuals who are responsible for manually categorising text as abusive or non-abusive. Furthermore, this project can contribute to the development of more accurate and reliable sentiment analysis systems, leading to a safer and healthier online environment.

II. RELATED WORK

In order to detect sarcasm in social media texts, we aim to improve upon an existing methodology presented in a research paper [1] that we discovered during our initial research. The methodology used in the paper involved a hybrid ensemble model to detect sarcasm on social media platforms. This model combines the use of three popular natural language processing techniques for generating word representations, which are GloVe, Word2Vec, and BERT. While GloVe and Word2Vec are word embedding techniques, BERT is a deep learning model that can generate contextualised word representations. The ensemble model is further enhanced with the use of fuzzy logic to improve its performance on sarcasm detection. During the training phase of the existing methodology presented in the referenced paper, the context used for detecting sarcasm was based on the words surrounding the center word, except for BERT, which evaluated the context of the sentence.The

probabilities of classification from all three models were passed to the fuzzy logic layer, where the information learned from the three models was weighted as high, medium, or low based on the score of embedding vectors provided by the respective model. The fuzzy logic module then determined whether a statement was sarcastic or not based on fuzzy rules. The purpose of this approach was to combine the strengths of three different models, both word-based and sentence-based, and to utilise fuzzy logic to balance their limitations, resulting in a more precise and comprehensive method for detecting sarcasm in text.

We intend to enhance the existing methodology for sarcasm detection by presenting a hybrid machine learning model using FastText, Random Forest Classifiers and SVC's, applied to the same SARC dataset as our primary paper. The objective is to compare the performance of these models with the hybrid ensemble model presented in the referenced paper. To evaluate the accuracy of our approach, we will conduct tests to assess the precision of our data model and machine learning process in identifying sarcastic comments.

III. METHODOLOGY

Our project's stages can be broadly defined as:

I. Data gathering
II. Data cleaning
III. Sentiment analysis and Modelling
IV. Result visualisation

3.1 DATA GATHERING

A self-annotated Reddit corpus collection dataset refers to a collection of Reddit posts or comments that have been annotated or labelled by the users themselves or by a team of researchers. The dataset we used was created by Khodak et al. in 2017, the dataset contains 1.3 million sarcastic comments from the Internet commentary website Reddit. It contains statements, along with their responses as well as many non-sarcastic comments from the same source, in English language in CSV file format.

The Sarcasm Corpus V2 is a subset of the Internet Argument Corpus (AIC) which is composed of posts annotated for sarcasm. The dataset represents three distinct categories of sarcasm, including general sarcasm, hyperbole, and rhetorical questions. The Sarcasm Corpus V2 is an update to the previous version, Sarcasm Corpus V1, and it consists of a total of 6,250 posts, with 3,260 posts per class for both sarcastic and non-sarcastic posts.

3.2 DATA PRE-PROCESSING AND CLEANING

We are using Python to implement our code and a range of libraries are being utilised to aid in either preprocessing our dataset or in training our models. The libraries that are being used for preprocessing are Pandas, Numpy and Matplotlib. The Python library Seaborn is being used for checking if the dataset consists of more sarcastic comments or more non-sarcastic comments. The libraries that are primarily being used for the machine learning and training aspect of our code are Spacy, Transformers, Tensorflow, Keras and Sklearn. In order to write and execute our code we have made use of Google Colab and Jupyter Notebook as our environment as it is a convenient tool to use when working with data science and machine learning workflows. The code is being run as subsections using Jupyter Notebook in a single file where the dataset is being read using pd.read_csv from several csv files.

3.3 SENTIMENT ANALYSIS AND MODELLING

The machine learning algorithms that we are making use of in our research to refine sarcastic sentiment analysis in social media are GloVe, FastText and BERT.

The Global Vectors for Words Representation (GloVe) is an unsupervised machine learning algorithm used for creating word embeddings, which can be defined as vector representations of words in high-dimensional space. In our implementation, we will be using these word embeddings as features in natural language processing (NLP) for sentiment analysis and specifically, sarcasm detection. Our model is designed to emulate the GloVe algorithm implementation in our primary paper.

The FastText algorithm is designed for efficient text classification and natural language processing tasks and it is an extension of the Word2Vec machine learning algorithm. It takes text as input and converts it into a vector representation which is then used for classification. This model breaks an input text into subwords and each of these subwords is represented as a vector. This is how our code is utilising this algorithm to assist us in pre-processing our datasets.

Bidirectional Encoder Representations from Transformers (BERT) is a neural network-based algorithm for natural language processing tasks such as question answering, sentiment analysis, and language translation. It was developed by Google in 2018 and is currently one of the most widely used NLP models. It is a type of transformer-based model, which means that it uses a self-attention

mechanism to analyse the context of each word in a sentence.

In this research study, we carefully considered different classifiers and ultimately chose the Support Vector Classification (SVC) and Random Forest Classifier for our text classification task. We chose SVC as one of our classifiers due to its effectiveness with small datasets and its robustness to outliers compared to other classifiers. Our concern about the size of our training dataset and the possible presence of noisy or mislabeled data points led us to choose SVC as a viable option.

For our second classifier, we selected the Random Forest Classifier, an ensemble learning method that combines the results of multiple decision trees to reduce the risk of overfitting and improve the generalisation performance of the model. Given the high number of features in our text data, overfitting could be a significant issue, making Random Forest an excellent choice.

Furthermore, Random Forest provides a measure of the importance of each feature in the classification task, which helps us identify the most useful words or phrases in distinguishing between sarcastic and non-sarcastic text. This feature is valuable in understanding the nature of sarcasm in text and improving the accuracy of the model.

3.4 RESULT VISUALISATION

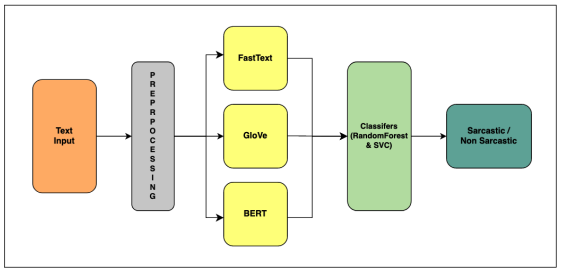

Figure 1: Architecture of our modified model.

IV. RESULTS AND DISCUSSION

Our contribution to the research on sarcastic post detection was to replace one of the three machine learning models that were used in our primary paper [1]. The FastText model was utilised instead of Word2Vec model in order to determine whether more accurate results could be achieved in comparison to the ones that came from our primary papers research.

FastText is particularly effective at capturing the meaning of rare words by breaking words down into smaller sub-word units or "n-grams" and using these sub-word units to build a language model. In contrast, Word2Vec tends to struggle with words that do not appear frequently in the training corpus.

The FastText machine learning model has the capacity to handle a much larger vocabulary than Word2Vec since it is designed to break words down into sub-word units. Because of these smaller sub-word units, FastText has shown to be faster to train than Word2Vec. This was especially important as it allows for efficiency when training large datasets, such as the ones we used in this research.

Additionally, in comparison to Word2Vec, the FastText was shown to improve the performance of downstream NLP tasks such as text classification and sentiment analysis, evident in our results.

GloVe Accuracy	0.6855828220858896
GloVe Precision	0.6956521739130435
GloVe Recall	0.6459627329192547
GloVe F1-score	0.6698872785829308
FastText Accuracy	0.4938650306748466
FastText Precision	0.4938650306748466
FastText Recall	1.0
FastText F1-score	0.6611909650924024
BERT Accuracy	0.7108895705521472
BERT Precision	0.6943231441048034
BERT Recall	0.7406832298136646
BERT F1-score	0.7167543200601051
Ensemble Accuracy	0.6301124744376277
Ensemble Precision	0.6279467828975646
Ensemble Recall	0.7955486542443064
Ensemble F1-score	0.6826108545784795

Figure 2: Random Forest Classifier results for GEN SARC dataset

GloVe Accuracy	0.6871165644171779
GloVe Precision	0.6874766876732766

GloVe Recall	0.6871165644171779
GloVe F1-score	0.6867715602047961
FastText Accuracy	0.4938650306748466
FastText Precision	0.2439026685234672
FastText Recall	0.4938650306748466
FastText F1-score	0.3265390962572908
BERT Accuracy	0.7223926380368099
BERT Precision	0.7247436535847346
BERT Recall	0.7223926380368099
BERT F1-score	0.7219264446691368
Ensemble Accuracy	0.75
Ensemble Precision	0.750105857445307
Ensemble Recall	0.75
Ensemble F1-score	0.7500088215575341

Figure 3: SVC results for the GEN SARC dataset

GloVe Accuracy	0.6125
GloVe Precision	0.6081218274111675
GloVe Recall	0.6062753036437247
GloVe F1-score	0.6071971616827166
FastText Accuracy	0.494
FastText Precision	0.494
FastText Recall	1.0
FastText F1-score	0.6613119143239625
BERT Accuracy	0.587
BERT Precision	0.5835051546391753
BERT Recall	0.5728744939271255
BERT F1-score	0.578140960163432
Ensemble Accuracy	0.5645
Ensemble Precision	0.5618756606834476
Ensemble Recall	0.72638326585695
Ensemble F1-score	0.6155500120567037

Figure 4: Random Forest Classifier results for SARC dataset using 10000 comments

GloVe Accuracy	0.6365
GloVe Precision	0.6366528884537169
GloVe Recall	0.6365
GloVe F1-score	0.636128588600453
FastText Accuracy	0.494
FastText Precision	0.244036
FastText Recall	0.494
FastText F1-score	0.32668808567603747
BERT Accuracy	0.6205
BERT Precision	0.620474579497491
BERT Recall	0.6205
BERT F1-score	0.6203381156999981
Ensemble Accuracy	0.6425
Ensemble Precision	0.6425364399924218
Ensemble Recall	0.6425
Ensemble F1-score	0.6422980928008281

Figure 5: SVC results for the SARC dataset using 10000 comments

VI. CONCLUSION

In this study, we aimed to develop a more accurate model for detecting sarcasm on the SARC dataset using a secondary Sarcasm Corpus V2 dataset. We modified a hybrid model and used GloVe, FastText, and BERT algorithms for word embeddings and text classification. For analysis, we utilised SVC and Random Forest Classifier, and for preprocessing, analysis, and visualisation, we made use of various Python libraries. Our results showed lower accuracy scores than the hybrid model on the SARC dataset, which could be due to various reasons. For instance, we may need to further improve our preprocessing step, fine-tune the word embeddings and classifiers, or use a larger secondary dataset. Nevertheless, our approach can help accurately identify sarcasm in text and

contribute to the development of more effective models.

VII. ACKNOWLEDGEMENTS

The authors thank their course coordinator, Reda Alhajj (Senior Member, IEEE), and teaching assistants, Kashfia Sailunaz and Ahmed Al Marouf, for their valuable assistance with the research paper. The authors appreciate their guidance and feedback, which were critical to the success of this project.

VIII. REFERENCES

[1] S. Khan, S. Asghar, S. Aslam, and H. Ullah, "Sarcasm Detection over Social Media Platforms Using Hybrid Ensemble Model with Fuzzy Logic,", electronics, vol. 12(4), pp. 937-958, Feb 2023.

[2] H. Chen, S. Mckeever, and S. J. Delany, "Harnessing the Power of Text Mining for the Detection of Abusive Content in Social Media," Advances in Computational Intelligence Systems, pp. 187-205, Jan. 2017.

[3] M. P. Akhter, Z. Jiangbin, S. I. R. Naqvi, M. Abdelmajeed, and T. Zia, "Abusive language detection from social media comments using conventional machine learning and deep learning approaches", Multimedia Systems, 28, pp. 1925-1940, Apr 2021.

[4] R. Cao, R. K.-W. Lee, and T.-A. Hoang, "DeepHate: Hate Speech Detection via Multi-Faceted Text Representations", WebSci '20: 12th ACM Conference on Web Science, July 2020..

[5] O. M. Singh, S. Timilsina, B. K. Bal, and A. Joshi, "Aspect based abusive sentiment detection in Nepali social media texts," in Proc. of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, pp. 908-915, 2020.

[6] M. Khodak, N. Saunshi, and K. Vodrahalli, "A Large Self-Annotated Corpus for Sarcasm," Apr. 2017. [Online]. Available: https://arxiv.org/abs/1704.055793].

[7] S. Oraby, V. Harrison, L. Reed, E. Hernandez, E. Riloff, and M. Walker, "Creating and Characterising a Diverse Corpus of Sarcasm in Dialogue," in Proceedings of the 17th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL), Los Angeles, CA, USA, 2016. [Online]. Available: https://nlds.soe.ucsc.edu/sarcasm2

IX. APPENDICES

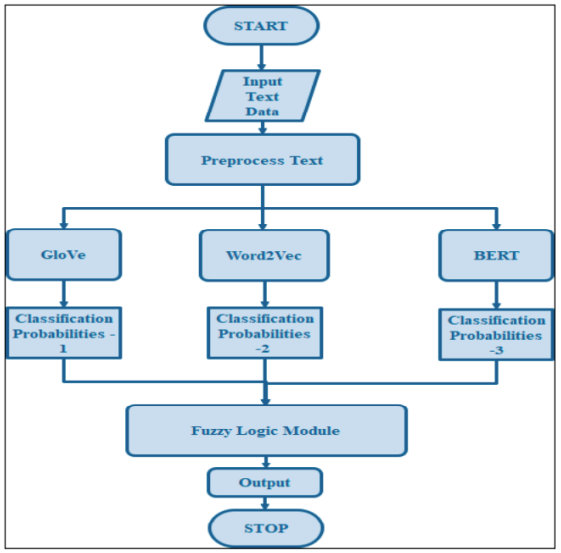

Figure 6: Flowchart of proposed model from primary paper [1]

Models	Accuracy	Precision	Recall	F1
SARC [8]	77.00	-	-	-
Multihead Attention [18]	81.00	-	-	-
RCNN-RoBERTa [22]	79.00	0.78	0.78	0.78
BERT+EmoNetSent [27]	-	-	-	0.775
CASCADE [28]	74.00	-	-	0.75
ELMo-BiLSTM [29]	79.00	-	-	-
Proposed	85.38	0.85	0.85	0.85

Figure 7: Performance comparison of the SARC dataset taken from primary paper [1]

Models	Accuracy	Precision	Recall	F1
Sarcasm Magnet [13]	72.5	0.73	0.71	0.72
Sentence-Level Attention [14]	74.9	0.74	0.75	0.74
A2 TextNet [15]	80.1	0.83	0.80	0.80
Self-Matching Network [17]	74.4	0.76	0.72	0.74
Multihead Attention [18]	81.2	0.80	0.81	0.81
Proposed	86.8	0.84	0.88	0.86

Figure 8: Performance comparison of the Twitter dataset taken from primary paper [1]

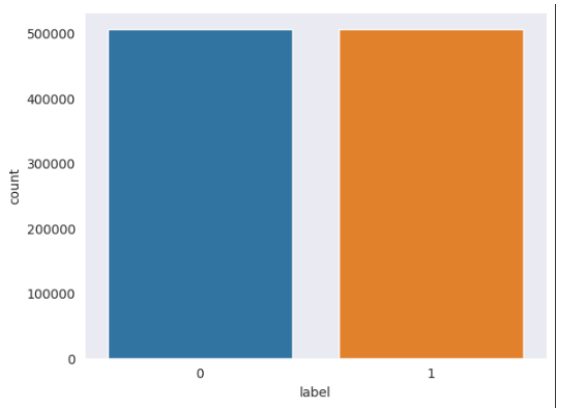

Figure 9: Seaborn visualisation showing that SARC dataset is balanced, consisting of equal number of sarcastic and non sarcastic comments

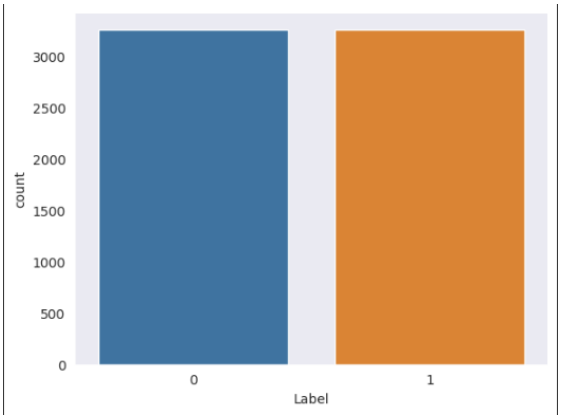

Figure 10: Seaborn visualisation showing that Sarcasm Corpus V2 dataset is balanced, consisting of equal number of sarcastic and non sarcastic comments