Harnessing Advanced Learning for Sarcasm Detection

Prasanna Kumar M

School of Computer Science and Engineering  
 Vellore Institute of Technology  
Chennai, India  
prasannakumar.m2020@vitstudent.ac.in

Karthika Veeramani

School of Computer Science and Engineering,

Vellore Institute of Technology,

Chennai, India

karthika.v@vit.ac.in

Abstract— Sarcasm in social media and in various methods of communication attracts higher attention and create profound impact even more than the customary positive and negative responses. It is necessary to propose a prototype and henceforth a fully-fledged system as a solution to detect sarcasm, to spread positive etiquette and avoid misguidance and misinformation. Unlike other text classification tasks, sarcasm detection is bit more challenging as it relies more on contextual nuances rather than actual word sentiment. The work leverages 5 learning algorithms (Logistic Regression, Multi-head Attention Encoder, Siamese Networks, BiLSTM and BiGRU) with data extracted from SARC (Self Annotated Reddit Corpus) Dataset and News Headline dataset. The models use contextual nuances, linguistic features to detect and analyze sarcasm. An analysis on how the model’s performance varies based on the nature of text data (ie. formal and informal) has been made to learn about the nuanced nature of sarcasm.

Keywords—Sarcasm, Prototype, Contextual Nuances, Logistic Regression, BiLSTM, BiGRU, Siamese Networks, Multi-head Attention Encoder

# Introduction

Sarcasm is a type of verbal irony in which someone says something with a contradictory meaning, often using a tone of voice that indicates the intended sarcasm. Sarcasm is based on the listener's ability to recognize the difference between the literal meaning of the words spoken and the speaker's true intention. It is commonly used for humors, criticism, or expressing dissatisfaction. This study aims to counter the formidable challenges that are present in the task of detecting sarcasm in textual data, particularly on platforms such as Reddit and news headlines, where context varies greatly. Sarcasm, which relies on subtle linguistic cues and is frequently devoid of explicit markers, presents a significant challenge for both humans and machines. This work uses advanced deep learning techniques such as Multi-head Attention and Siamese Networks to capture nuanced linguistic features and contextual understanding, which are required for accurate detection. Achieving this goal will improve sentiment analysis on social media and news platforms, pushing the boundaries of NLP research and improving comprehension of textual data.

The successful detection of sarcasm in textual data has significant implications for sentiment analysis on social media and news sites. By correctly identifying sarcastic remarks, the proposed model can provide more nuanced information about public opinion, sentiment trends, and the overall tone of online discourse. This advancement not only improves the understanding of textual data, but also allows for more accurate and contextually aware sentiment analysis, which improves decision-making processes across multiple domains. Furthermore, pushing the boundaries of Natural Language Processing (NLP) research in sarcasm detection helps to advance computational linguistics and machine comprehension of human communication. Sarcasm is a complex linguistic phenomenon that is deeply embedded in human language and culture, so successfully detecting it in textual data represents a significant milestone in the field of NLP. By addressing the challenges of sarcasm detection, this study helps to develop more sophisticated NLP models capable of handling the complexities of human communication in textual form. Overall, the proposed work improves the ability to accurately interpret textual data while also pushing the boundaries of NLP research, paving the way for more advanced and contextually aware language understanding systems.

# LITERATURE SURVEY

There have been many approaches dedicated to find sarcasm in different platforms with diverse datasets. Traditional Machine Learning models are used to detect sarcasm at ease. Garg A and Duhan N mentions using Support Vector Models to detect sarcasm collected from Twitter API at a high accuracy [1]. Use of tree models such as Random Forest Classification and x Boosting also gives a good result in sarcasm detection as in the case of Kumar A and Garg G used over reddit dataset [2]. Naïve Bayes and Fuzzy clustering were used to classify sarcastic data in microblogs as done by Mukherjee S and Bala P.K [3]. Ensemble and Boosting models were used by Bagathe , R.A and Suguna R [4] to classify sarcasm at a very high accuracy rate. But one major drawback in using traditional models it lacks the capability to handle the nuances and sequential dependencies in a text.

To address these issues faced, deep learning models like LSTM and GRU were used for most of the text classification tasks. Some examples include usage of CNN-LSTMs for news headline sarcasm detection by Mandal P.K and Mahto R [5], twitter sarcasm detection using CNN-SVM by Samer Muthana Sarsam et al. [6], usage of LSTM , GRU , CNN etc. along with ensemble learning by Priya Goel et al.[7] .

Things get challenging when there is heterogenous data such as code and Pandey, R., and Singh, J. P. [8] has used BERT-LSTM model to detect sarcasm in social media text which is mixed with code. Salim S.S et al..[9] has used Deep LSTM-RNN models with word embeddings on Twitter and has also proposed a genetic optimization algorithm over CNN and LSTM for better Performance[10]. Sarcasm detection in a mashup language where elements of two or more languages are combined is a difficult task, and it was done at a high accuracy by Jain et al. [11] using BiLSTM and CNN.

Pre-trained models and Transfer Learning models having been trained over a large set of corpora and being complex, tend to perform better when tuned with right parameters. Savini, E., and Caragea, C. [12] uses BERT for building a intermediate-task transfer learning for sarcasm detection. Eke et al. [13] uses BERT with GloVe to identify sarcasm in Twitter and Internet Argument Corpus. Sometimes, comments can be in different languages, as in the case of the works of Aggarwal A. et al. [14] where bilingual word embeddings are used to detect sarcasm. Kumar. Z et al. [15] has developed a model that uses embeddings of emojis combined with words for sarcasm in Indian languages at a higher accuracy rate.

Apart from normal deep learning models, there are some advanced Deep learning models based on Attention mechanism, Contrast Learning and Dynamic Routing models which are more efficient thanks to their parallelization and ease of training. Vitman O et.al [16] proposed a sarcasm detection framework that takes important factors like context and sentiment. Attention model focuses on different parts of input making them more efficient in supervised learning tasks, as in the case of Olaniyan D. et al. [17] where an attention-based LSTM is used to detect Sarcasm in social media. Kumar A et al. [18] built a multi-headed self-attention architecture model to detect sarcasm with proof of concept from multiple datasets. Ghosh, D., Fabbri, A. R., & Muresan, S. [19] explores the role of conversational contexts in twitter for the detection of sarcastic tweets using sentence-level attention-based LSTM. The use of self-attention-based transformers for NLP tasks like text classification, machine translation, text summarization, sentimental analysis and LLMs has increased a lot ever since the model gained limelight in “Attention is all You Need” in 2017 by Vaswani et.al [20]. Transformers. Many transformer-based sarcasm and irony detection models has been proposed over the recent years. Potamias et al. [21] used transformers with the help of a recurrent CNN to detect sarcasm and irony in four datasets, and achieving SOTA performance in all the four datasets. Kumar, A., and Anand, V. [22] and Avvaru et al. [23] have used transformer-based models to detect sarcasm taking conversional context into account. Transformer models can also be used to create word embeddings with then can be used with other deep learning models like CNN as in the case of Ahuja, R., and Sharma, S. C. [24] where an encoder transformer model named LMTweets is used on SARC dataset to detect sarcasm with high accuracy. Pre-trained transformer models are also used with other conventional DL models like int the case of Mohan A. et al. [25] where a GraphCNN-BERT architecture model is used to detect sarcasm.

# METHODOLOGY

For this work, 2 publicly available datasets, SARC dataset proposed by Khodak, M., Saunshi, N., & Vodrahalli, K. [26] which contains over 1 million comments taken from reddit and the News Headline dataset proposed by Misra,Rishabh et.al [27][28] which contains around 56,700 news headlines has been used.

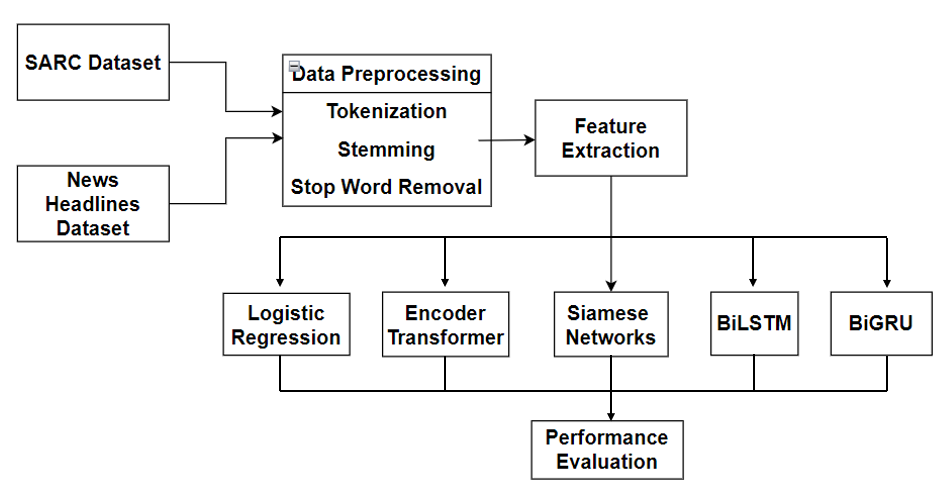


Fig. 1 Proposed System Design

Figure 1 shows a basic outline of the workflow of the system. The data is first collected from the SARC and the News Headlines dataset. The collected data in the proposed system is carefully pre-processed to make sure it is suitable for training advanced learning models. In this stage, the textual data is cleaned and standardised using a variety of techniques like tokenization, stemming, and stop word removal. Furthermore, pertinent characteristics are examined and extracted to obtain the contextual data and linguistic cues required for the detection of sarcasm. Word embeddings, syntactic features, and semantic representations are a few examples of these features.

To help with model training and evaluation, the pre-processed data is divided into three with each set for the training process, validation of the trained data and testing the model’s performance respectively. Both datasets are used to train ML and DL models, enabling thorough learning across a variety of textual data sources. To find the best-performing models, a variety of models are trained and compared, including Siamese networks, BiGRU(Bidirectional Gated Recurrent Unit), BiLSTM (Bidirectional Long Short-Term Memory), Multi-head Attention Encoder Transformer, and Logistic Regression. The models acquire the ability to precisely identify sarcasm in textual data by identifying minute linguistic cues and contextual nuances through an iterative process of model training and evaluation.

Relevant performance metrics are employed to asses these models. To gain insight into the models' performance across various evaluation criteria, metrics like the Receiver Operatic Characteristic (ROC) score, precision, recall, F1-score, and accuracy score has been employed. Through the methodical assessment of the models' performance with these metrics, researchers can ascertain the effectiveness of the sarcasm detection system and pinpoint areas that require further development. The top-performing models are ultimately chosen for system integration, guaranteeing reliable and accurate sarcasm detection across a range of textual data sources.

One thing to be noted here is that since the nature of the comments in the SARC dataset is stochastic, informal and uses a lot of figurative language, it is better not to filter-out special characters or apply any pre-processing as the models will take cues from these special characters to detect sarcasm. For the news headlines dataset however, pre-processing such as special character removal, white space removal, tokenization, lemmatization etc. are applied since the data in more deterministic, structured, and formal.

# Implementation

For building Deep Learning models, TensorFlow Keras version 2.x has been used to easily build and train a deep learning model. For Logistic Regression, the data needs to vectorized first using the Term-frequence Inverted-document Frequency (TFIDF) technique, and later scikit-learn Api for Logistic Regression is used.

## Logistic Regression

The Logistic Regression model classifies the data instances based on the probability obtained from the logistic/sigmoid function after the data is vectorized. The Logistic Regression API has 3 parameters max\_iter for maximum iterations, random\_state for reproducibility and n\_jobs for parallel processing in case of large datasets. If the probability comes to be above or equal to 0.5, then the data is classified to the positive class (ie. sarcastic) and a probability score below 0.5 indicates that the data is classified to the negative class (ie. non-sarcastic). Overall Logistic Regression is a good approach for classification especially if there is only one feature variable available.

## Encoder Transformer

The Encoder Transformer model architecture for text classification tasks, such as sarcasm detection, starts with embedding input tokens to capture semantic meaning and then adds positional encoding for sequence context. It then uses a Multi-Head Self-Attention Mechanism to analyse token relationships, which is followed by a Feedforward Neural Network for additional processing. A 1D convolutional layer with pooling captures local features. Finally, fully connected layers improve the output for classification. This architecture seamlessly integrates attention mechanisms, neural networks, and convolutional layers, making it capable for detecting like sarcasm.

## Siamese Network

The Siamese network model is designed with an initial input layer feeding into an embedding layer that converts text sequences into dense vectors. Dropout layers are used to regularise the model and reduce the risk of overfitting during training. Notably, the model has dual parallel branches, each with dense layers that reads different features like contextual feature, linguistic feature, and sentiments from the input data. Following the fusion of these branches, a final similarity layer produces a single output, providing information about the likelihood of similarity between the inputs. This architecture excels at learning nuanced embeddings and comparing textual representations, making it ideal for tasks requiring advanced text comprehension and categorization, such as sarcasm detection.

## Bidirectional Long Short Term Memory (BiLSTM)

The BiLSTM model, designed for sarcasm detection, uses a dynamic architecture to capture complex patterns in input sequences. It starts with a layer of embeddings that converts words into dense vectors, then moves on to BiLSTM layers that can understand both past and future contexts at the same time. Dropout layers and batch normalisation are strategically used to improve model robustness and avoid overfitting. This model deciphers subtle nuances in textual data, thanks to a series of dense layers for feature extraction and a final sigmoid-activated output layer for binary classification, making it ideal for accurately detecting sarcasm.

## Bidirectional Gated Recurrent Unit (BiGRU)

The BiGRU model, designed for text classification tasks such as sarcasm detection, features a dynamic framework that captures intricate patterns in input sequences. At its heart is an embedding layer that converts words into dense vectors, laying the groundwork for Bidirectional Gated Recurrent Unit (BiGRU) layers. These layers, which are intended to capture bidirectional information flow, are supplemented with dropout and batch normalisation to improve model generalisation, and reduce overfitting. Following a series of dense layers for feature extraction, a sigmoid-activated output layer enables binary classification. This configuration gives the model the ability to detect subtle textual nuances, making it an effective tool for sarcasm detection in textual data.

# Results And Inferences

The results of the 5 Machine / Deep learning models used for the News Headlines dataset with embedding dimension set to 40 and maximum length of the sequence set to 20 is provided in the Table 1

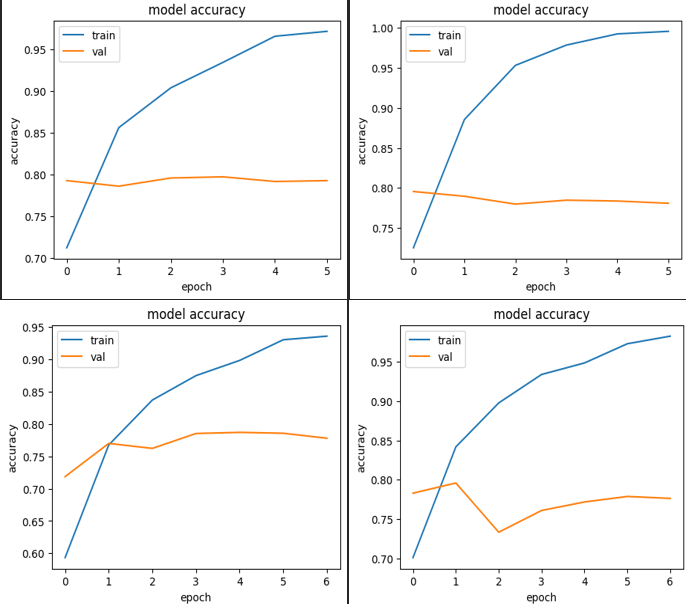
Table 1: Models performance in News Dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 | AUC |
| Logistic | 0.7865 | 0.7508 | 0.789 | 0.7695 | 0.98 |
| Transformer | 0.8011 | 0.7764 | 0.7280 | 0.7551 | 0.86 |
| Siamese | 0.7960 | 0.7643 | 0.713 | 0.738 | 0.85 |
| BiLSTM | 0.7872 | 0.7430 | 0.818 | 0.779 | 0.86 |
| BiGRU | 0.7953 | 0.7844 | 0.715 | 0.748 | 0.85 |

On observing the Table 1, the encoder transformer has given the highest accuracy closely followed by the likes of Siamese network and BiGRU can be inferred. Also, BiGRU has the highest precision score, meaning that it classifies true positives comparatively better than others. BiLSTM has the highest recall score amongst the five, meaning it effectively captures positive instances and avoids false negatives better than the rest. Logistic Regression has the highest AUC score, meaning it can distinguish better between positive and negative classes better than other models.

Figure 2 shows the accuracy plots for training vs validation data in News Headlines dataset with Transformer, Siamese Networks, BiLSTM and BiGRU.

Fig. 2 Accuracy Plots for Transformer (top-right), Siamese Networks (top-left), BiLSTM (bottom-left) and BiGRU (bottom-right) for News Dataset)



The results of the 5 Machine / Deep learning models used for the SARC dataset with embedding dimension set to 40 and maximum length of the sequence set to 20 is shown in Table 2.

Table 2 Models performance in SARC dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 | AUC |
| Logistic | 0.7051 | 0.6690 | 0.7188 | 0.6930 | 0.88 |
| Transformer | 0.6790 | 0.6841 | 0.6607 | 0.6721 | 0.75 |
| Siamese | 0.6843 | 0.6988 | 0.6446 | 0.6706 | 0.75 |
| BiLSTM | 0.6893 | 0.6782 | 0.6192 | 0.6473 | 0.70 |
| BiGRU | 0.6774 | 0.6827 | 0.6881 | 0.6753 | 0.74 |

From Table 2, it is inferred that that the logistic regression has given the highest accuracy closely followed by the likes of BiLSTM can be inferred. Also, Siamese Networks has the highest precision score, meaning that it classifies true positives comparatively better than others. Logistic Regression has the highest recall score amongst the five, meaning it effectively captures positive instances and avoids false negatives better than the rest. Logistic Regression has the highest AUC score, meaning it can distinguish better between positive and negative classes better than other models.

Figure 3 shows the accuracy plots for training vs validation data in SARC dataset with Transformer, Siamese Networks, BiLSTM and BiGRU. On observing the above results, a conclusion can be made that all the models give a good accuracy score of around 78-81% for all the models in the News Headlines dataset whereas it gives a moderate accuracy score of 67-71% for all the models in the SARC dataset. This could be possibly because the nature of sarcasm in reddit comments in the SARC dataset is more stochastic due as it is more context dependent than in the case of News Headlines dataset which is comparatively more straightforward and deterministic, making it easier for the models to learn from the data. Also, simpler models like Logistic Regression perform better than complex models like Transformers etc. in the case of SARC dataset as the reddit comments are more colloquial, unstructured and uses a lot of informal language compared to a more structured and formal news headlines dataset.

On analysing the above results, it is substantially better than the results in previous works, taking into the account of stochastic nature of the SARC dataset which makes the results more unpredictable. Though, there is a scope for improvement in the results in both the datasets. Analysing different datasets, its nature, and the impact of model’s varying performance in different datasets makes this research a clear stand out from the rest.

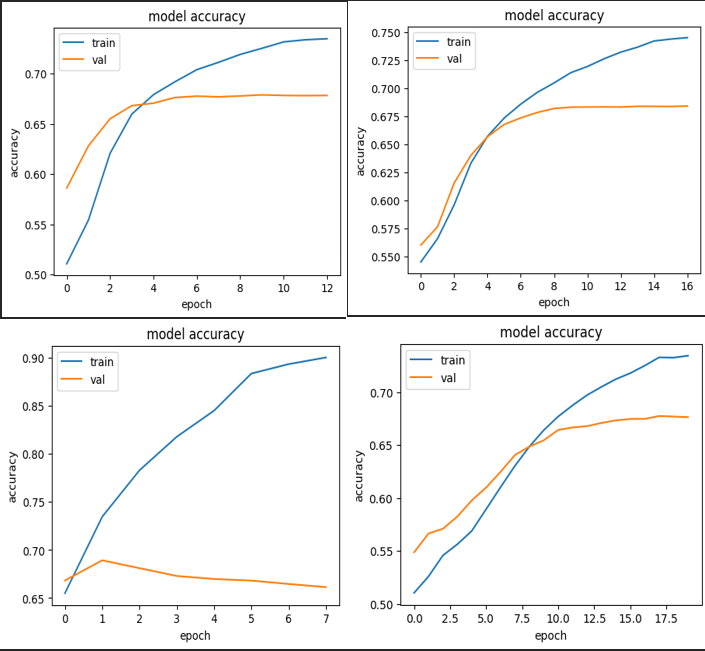


Fig. 3 Accuracy Plots for Transformer (top-right), Siamese Networks (top-left), BiLSTM (bottom-left) and BiGRU (bottom-right) for SARC Dataset

# Conclusion and Future Work

The attempt to detect sarcasm yielded promising results, demonstrating the efficacy of various machine learning algorithms on diverse datasets. The findings revealed that, while the Encoder Transformer excelled at detecting irony in the more deterministic world of News Headlines, Logistic Regression was a formidable competitor, particularly when it came to navigating the stochastic nature of the SARC dataset. Despite these successes, the research reveals some inherent limitations in existing methodologies. Although the models are good at detecting sarcasm in a specific textual context, they are not as good at extrapolating this knowledge to infer the more general contextual subtleties that influence communication dynamics over time.

The potential future research work will focus on resolving these issues and expanding the capabilities of sarcasm detection. Increasing the models' ability to transcend the limitations of their current context is a critical area for development. The techniques to improve the performance of the sarcasm detection systems by using cutting-edge modelling techniques and sophisticated pre-processing methods will be studied in future. This will allow them to detect nuances in sarcasm across a wide range of temporal and thematic dimensions. This entails developing methodologies for identifying the implicit contextual cues that give utterances meaning and intent, in addition to the overt sarcastic cues found within them. In addition to textual content, the use of additional modalities such as images, audio, and video has the potential to improve the understanding of sarcasm by identifying contextual signals and nonverbal cues that traditional text-based methods may miss. To improve the resilience and adaptability of sarcasm detection systems and expand their use across a wider range of websites and communication channels, the future research will investigate the viability and feasibility of using multi-modal datasets.

# References

1. Garg, A., & Duhan, N. (2020). Sarcasm Detection On Twitter Data Using Support Vector Machine. *ICTACT Journal on Soft Computing*, *10*(4).
2. Kumar, A., & Garg, G. (2023). Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. *Journal of ambient intelligence and humanized computing*, *14*(5), 5327-5342.
3. Mukherjee, S., & Bala, P. K. (2017). Sarcasm detection in microblogs using Naïve Bayes and fuzzy clustering. *Technology in Society*, *48*, 19-27.
4. Bagate, R. A., & Suguna, R. (2022). Sarcasm Detection with and without# Sarcasm: Data Science Approach. *International Journal of Information Science and Management (IJISM)*, *20*(4), 1-15.
5. Mandal, P. K., & Mahto, R. (2019). Deep CNN-LSTM with word embeddings for news headline sarcasm detection. In *16th International Conference on Information Technology-New Generations (ITNG 2019)* (pp. 495-498). Springer International Publishing.
6. Sarsam, S. M., Al-Samarraie, H., Alzahrani, A. I., & Wright, B. (2020). Sarcasm detection using machine learning algorithms in Twitter: A systematic review. *International Journal of Market Research*, *62*(5), 578-598.
7. Goel, P., Jain, R., Nayyar, A., Singhal, S., & Srivastava, M. (2022). Sarcasm detection using deep learning and ensemble learning. *Multimedia Tools and Applications*, *81*(30), 43229-43252.
8. Pandey, R., & Singh, J. P. (2023). BERT-LSTM model for sarcasm detection in code-mixed social media post. *Journal of Intelligent Information Systems*, *60*(1), 235-254.
9. Salim, S. S., Ghanshyam, A. N., Ashok, D. M., Mazahir, D. B., & Thakare, B. S. (2020, June). Deep LSTM-RNN with word embeddingfor sarcasm detection on Twitter. In *2020 international conference for emerging technology (INCET)* (pp. 1-4). IEEE.
10. Ashok, D. M., Ghanshyam, A. N., Salim, S. S., Mazahir, D. B., & Thakare, B. S. (2020, June). Sarcasm detection using genetic optimization on LSTM with CNN. In *2020 International Conference for Emerging Technology (INCET)* (pp. 1-4). IEEE.
11. Jain, D., Kumar, A., & Garg, G. (2020). Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN. *Applied Soft Computing*, *91*, 106198.
12. Savini, E., & Caragea, C. (2022). Intermediate-task transfer learning with BERT for sarcasm detection. *Mathematics*, *10*(5), 844.
13. Eke, C. I., Norman, A. A., & Shuib, L. (2021). Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and BERT model. *IEEE Access*, *9*, 48501-48518.
14. Aggarwal, A., Wadhawan, A., Chaudhary, A., & Maurya, K. (2020). " Did you really mean what you said?": Sarcasm Detection in Hindi-English Code-Mixed Data using Bilingual Word Embeddings. *arXiv preprint arXiv:2010.00310*.
15. Kumar, A., Sangwan, S.R., Singh, A.K. and Wadhwa, G., 2023. Hybrid deep learning model for sarcasm detection in Indian indigenous language using word-emoji embeddings. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(5), pp.1-20.
16. Vitman, O., Kostiuk, Y., Sidorov, G., & Gelbukh, A. (2023). Sarcasm detection framework using context, emotion and sentiment features. *Expert Systems with Applications*, *234*, 121068.
17. Olaniyan, D., Ogundokun, R. O., Bernard, O. P., Olaniyan, J., Maskeliūnas, R., & Akande, H. B. (2023). Utilizing an Attention-Based LSTM Model for Detecting Sarcasm and Irony in Social Media. *Computers*, *12*(11), 231.
18. Kumar, A., Narapareddy, V. T., Srikanth, V. A., Malapati, A., & Neti, L. B. M. (2020). Sarcasm detection using multi-head attention based bidirectional LSTM. *Ieee Access*, *8*, 6388-6397.
19. Ghosh, D., Fabbri, A. R., & Muresan, S. (2017). The role of conversation context for sarcasm detection in online interactions. *arXiv preprint arXiv:1707.06226*.
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*
21. Potamias, R. A., Siolas, G., & Stafylopatis, A. G. (2020). A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, *32*(23), 17309-17320.
22. Kumar, A., & Anand, V. (2020, July). Transformers on sarcasm detection with context. In *Proceedings of the second workshop on figurative language processing* (pp. 88-92).
23. Avvaru, A., Vobilisetty, S., & Mamidi, R. (2020, July). Detecting sarcasm in conversation context using transformer-based models. In *Proceedings of the second workshop on figurative language processing* (pp. 98-103).
24. Ahuja, R., & Sharma, S. C. (2022). Transformer-based word embedding with cnn model to detect sarcasm and irony. *Arabian Journal for Science and Engineering*, *47*(8), 9379-9392.
25. Mohan, A., Nair, A. M., Jayakumar, B., & Muraleedharan, S. (2023). Sarcasm Detection Using Bidirectional Encoder Representations from Transformers and Graph Convolutional Networks. *Procedia Computer Science*, *218*, 93-102.
26. Khodak, M., Saunshi, N., & Vodrahalli, K. (2017). A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*.
27. Misra, Rishabh and Prahal Arora. "Sarcasm Detection using News Headlines Dataset." AI Open (2023).
28. Misra, Rishabh and Jigyasa Grover. "Sculpting Data for ML: The first act of Machine Learning." ISBN 9798585463570 (2021).