SARCASM DETECTION (IN SENTIMENT ANALYSIS): A REVIEW USING DEEP LEARNING METHODS.

Abstract

Detection of sarcasm in textual data is crucial in modern natural language processing (NLP), as it can change the meaning of a statement entirely. Inability to detect satirical comments in NLP work will complicate classification algorithms. This problem has its consequences. Much research has been conducted on sarcasm detection (Norman et al. 2021). Although, most of them are content based cutting off circumstantial information. In this report, deep learning models with embedding were compared based on their respective performance metrics to address the concerns raised earlier. We used a publicly available dataset from Kaggle comprising of sarcastic and non-sarcastic words for training and evaluating our models. Our results suggest that Bidirectional Representations from Transformers (BERT) outperformed former networks based on accuracy, F1-score, receiver operating characteristic (ROC) area under curve score (AUC). Our pre-trained BERT model on our sarcasm detection task gave us an accuracy (0.94) and F1-measure (0.94) with an ROC-AUC score (0.97) respectively.

SECTION 1. INTRODUCTION

Sarcasm is a common way of expression for human beings, often used to convey criticism or humour in an indirect manner. Sarcasm is a form of irony that involves saying something but meaning the opposite. In recent times, sentiment analysis and affective computing has gathered a lot of recognition (Pak & Paroubek 2010, p.1320). Recent breakthroughs in deep learning techniques have enhanced NLP tasks, including sarcasm detection. The concept behind this approach is spotting the conflicting emotions within the written texts. This analysis of people's opinions identifies implicit subjective information in the files. The act of recognizing people's opinions about certain products, services, entities is of great benefits to the demand organizations. (Aquino J. 2012) Thus, it is an essential tool as it generates structured knowledge and patterns that support decision systems both local and global respectively. Imagine having an affective computing tool that refines recommendation systems and customer service management by disclosing clients' opinions and emotions. Further this approach can as well mitigate certain negative feedbacks from the customers. Detecting sarcasm in textual data can be challenging due to its highly contextual nature and the subtlety of language cues as it requires understanding the context, the speaker's tone, recognizing the use of irony, ambiguity, or exaggeration in the language. However, in recent years, deep learning (DL) techniques have shown promising results in sarcasm detection (Zhang et al. 2016, p.1562). Thus, several models have been reviewed by researchers to identify textual sarcasm, involving Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Transformer-based models. We recommended BERT model for sarcasm detection while comparing it with other embedding-based models such as Bi-RNN(GRU).

SECTION 2. RELATED WORKS IN SARCASM DETECTION (SENTIMENT ANALYSIS)

Many NLP research works have come up with many ways to automate sarcasm detection with different models. Some features were either revealed via DL or manual handcrafting method – feature engineering (Abulaish et al., 2020, p.152), never together. There has been so much dependence on DL by some scholars and vice-versa thus leaving rooms for more experimentations.

Some group of works believed that hashtags were the first indicators to site sarcasm (Riloff et al. 2012, Davidov et al., 2010, Bamman & Smith, 2015). Following the trend of CNN in solving NLP problems,

further studies were done by (Poria S. et al., 2016, Majumder N. et al., 2019). According to the former (Poria) four datasets were used to extract four feature set using convoluted neural network (CCN). They were sequentially combined and classified by a support vector machine (SVM) classifier. Aside this reliance on hashtags, few were able to add rulesets. For instance (Barbieri F. et al., 2014) used frequency and word sparseness as her main features. It was utilised by (Bouazizi & Ohtsuki, 2016) in extra rules on removing sarcastic words. This involved summing positive and negative tweet word and tallying it with the number of words with high emotion states subsequently. Seed phrase was implemented by Shmueli et al., 2009) such as "being cynical". This approach was then used to pick derisive comments online. Aside hashtag and rules related detection tack, old tweets from users were used to make features for other methods as initially implemented by (Bamman & Smith, 2015, p.104) with a thorough analysis on most relevant factors in trails of tweets from the same user. Figure 1 demonstrated it.

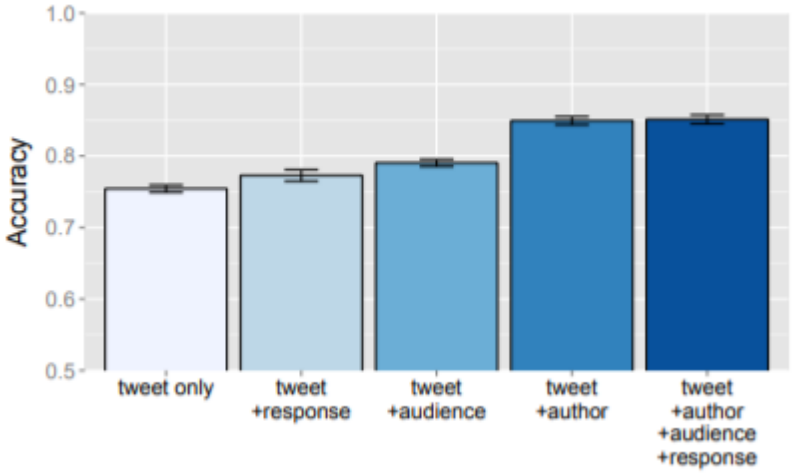


Figure 1.

It should be worth noting that the increased accuracy can be achieved when there is a combination of all the relevant features while considering their respective rules. Lately, (Baruah A., 2020, p.83) used a BERT architecture to test the use of historical data to identify sarcasm using conversational cues like response, consecutive previous utterances. Moreover, a second study from (Joshi et al. 2015) which based on contextual differences yielded a far better result using existing historical data. Simultaneously, the subsequent works on context-based sarcasm detection acknowledged the historical information (Riloff et al. 2012, Davidov et al., 2010, Bamman & Smith, 2015). Alternatively, some scholars have used glossary to detect sarcasm in the best possible ways recently. They designed lexicons solely on positive verbs and negative instances via bootstrapping (Riloff et al. 2012). Long-texts classification projects were carried out by some people using online product reviews and discussions demonstrating features in-forms of N-grams (Reyes et al. 2012 & Hazarika et al. 2018). Though this work was blurred by the fact that many accepted that sarcasm is once meant to be malicious anomaly, because of its temporal structure. Thus, hoping that in a short time the situation will change (Ebrahimi M. Et al. 2017, p.7075).

According to (Razali, MD et al., 2021), using deep learning extracted features they were able to recognize sarcasm in tweets. Convolutional neural network was used to extract feature set before it was merged with handmade set. While combining these elements, logistic regression was used as the classifier.

Table 1 below shows results on some existing works on sarcasm detection.

Author	Accuracy	Precision	Recall	F1-Score
Ilic S. et al. 2018.	88%	0.87	0.87	0.87
Kumar A. et al. 2015.	87%	0.87	0.91	0.86
Razali, MD et al. 2021	94%	0.95	0.94	0.94
Shmueli et al. 2020	87%	0.87	0.91	0.86

Table 1.

SECTION 3. Work Methodology

3.1 Dataset Collection

3.1.1 Data Details

New headlines (Asad & Sharif,2018) dataset used was taken from two Json files found on Kaggle. Being a self-contained data, they consist of three categories namely is_sarcastic, headline and article link (not required) respectively. Fig.2 shows the count

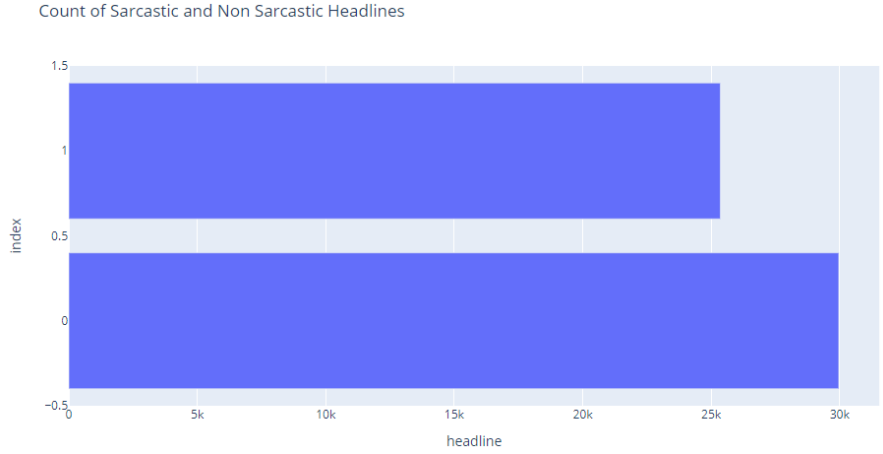


Fig.2

distribution for the dataset. Figs. 3 and 4 show Word clouds for the two categories respectively.

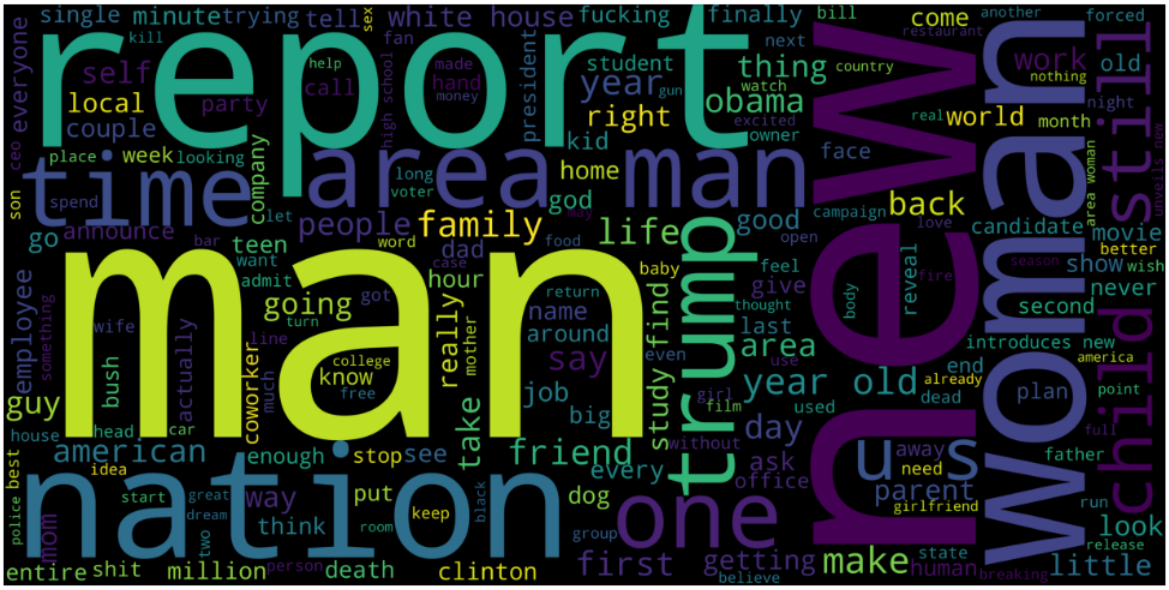


Fig.3.

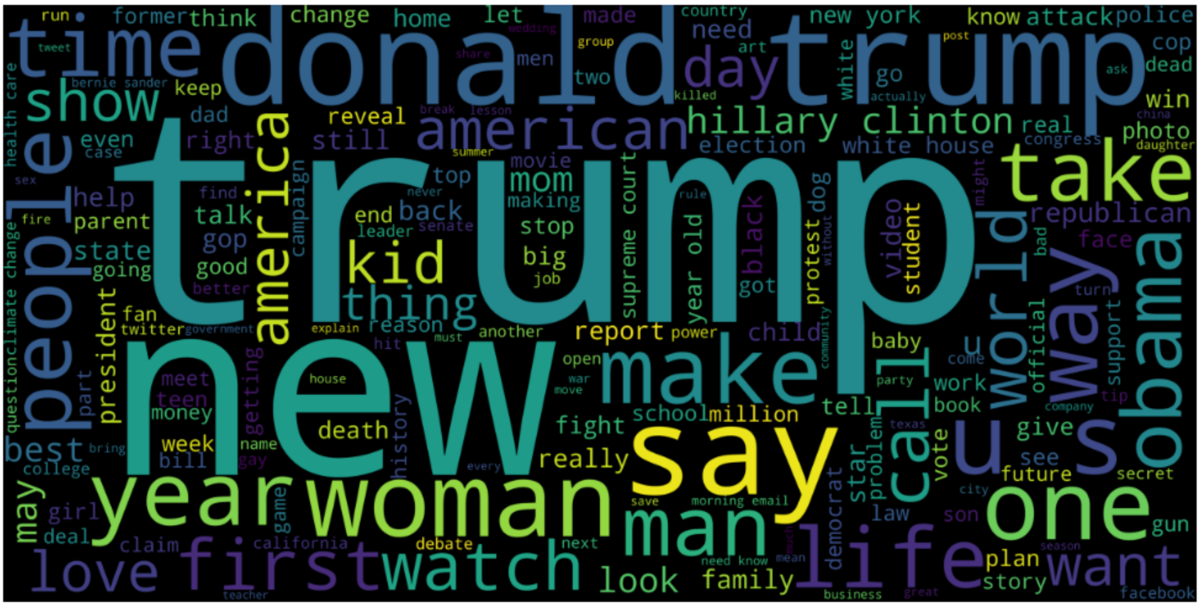


Fig.4.

3.2.1 Pre-processing of Data for Sarcasm Detection

For every automated NLP activity, text pre-processing is required. This step is key because it reduces computing difficulties, eases data handling complexity and ensures quality result for classification purposes. The steps taken are briefly explained below.

Data Cleaning / Punctuation Removal - This is done to ensure that that data is fit for analysis and that it will not disrupt word tokenization and eliminate noise.

Lowercasing - the text conversion to a lowercase usually done to standardize the data and ensure consistency in the data.

Stopword Removal – removing common words considered to be of little value for text analysis.

Tokenization – splitting a piece of text into smaller parts (tokens).

Lemmatization – reducing words to their base form or lemma.

Stemming – quite like lemmatization but the latter offers more accurate result.

Handling Emoticons and Parts -of-Speech tagging – replacing the emoticons and emojis with textual equivalents while removing text trends based on how many times word occurs within a given word class.

3.2.2 Exploratory Data Analysis using Latent Dirichlet Allocation (LDA)

This is an unsupervised statistical model used in NLP to identify topics within text corpus commonly used for topic modelling, document clustering and information retrieval. It was first implemented by (David et al.,2003). According to our analysis, it could be inferred that:

- In the entire dataset, the mean and median length of headlines is around 10 words.

- 45.8% and 54.2% are sarcastic and non-sarcastic headlines respectively.
- There are some headlines with 2 / 3 / 4 words.
- 85% of the headlines contain numbers approximately the same for the respective sarcastic and non-sarcastic headlines.
- Most headlines use numbers to attract viewership.
- LDA10, topic 8 is generally about social media
- Topic 9 is about politicians especially Donald Trump.
- Topic 2 is about children, violence, climate change.
- Sarcastic headlines tend to be pervasive across classes of news.

3.3 Sarcasm Detection (Deep Learning Frameworks)

3.3.2 Proposed Model - BERT

This work uses deep learning methods to detect sarcasm (Mehndiratta et al., 2017) in an unbalanced dataset as shown in Fig.2. This experimentation uses a BERT model as the proposed model with references to Embedding baseline models. BERT (BBC. "Hana Kimura,2020) has revolutionised the field of NLP since its inception in late 2018, producing latest results across many NLP tasks, including text classification.

BERT uses bidirectionality to learn the language by applying Self-Attention layers in both directions. The idea behind bidirectionality is to capture context from both directions for better performing in NLP tasks. (Zargaryan & Iskandaryan 2021) The architecture of BERT is a stack of Encoder blocks of the Transformer model; it does not use the whole Transformer architecture. It also uses the Encoder of the Transformer architecture to capture the semantic and syntactic information of the sequences. Fig.5 below shows a simple BERT architecture.

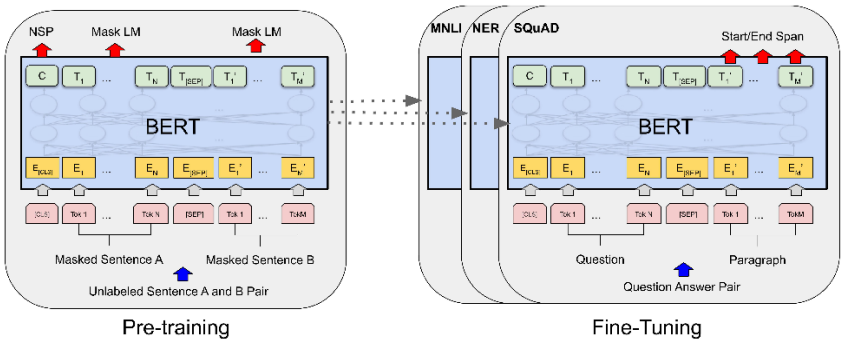


Fig.5.

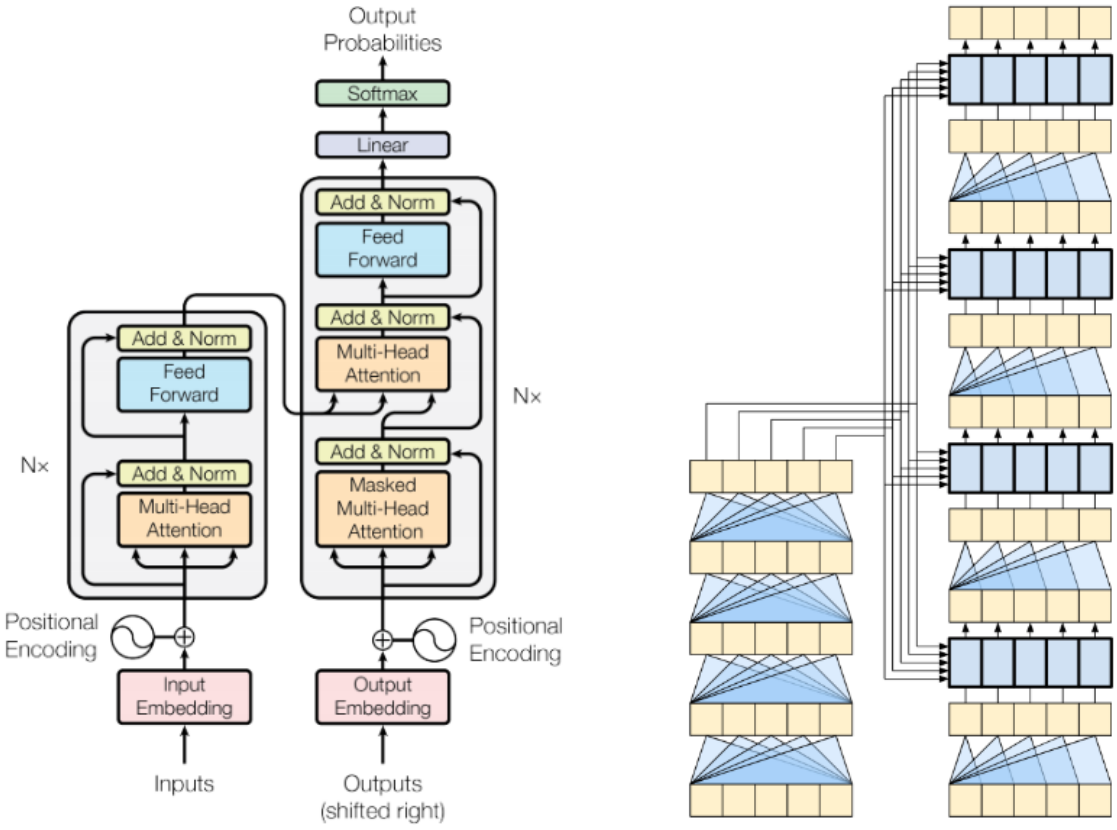


Fig.6.

BERT training involves two stages: pre-training and/or fine-tuning for a specific task. The pre-training phase performs dual functions namely Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM captures the bidirectional context of tokens within the sentence. The core idea of masking is to allow the model to learn from the textual sequence data. NSP captures the context of the sentence among other sentences. NSP shows connection between two sentences that MLM fails to produce.

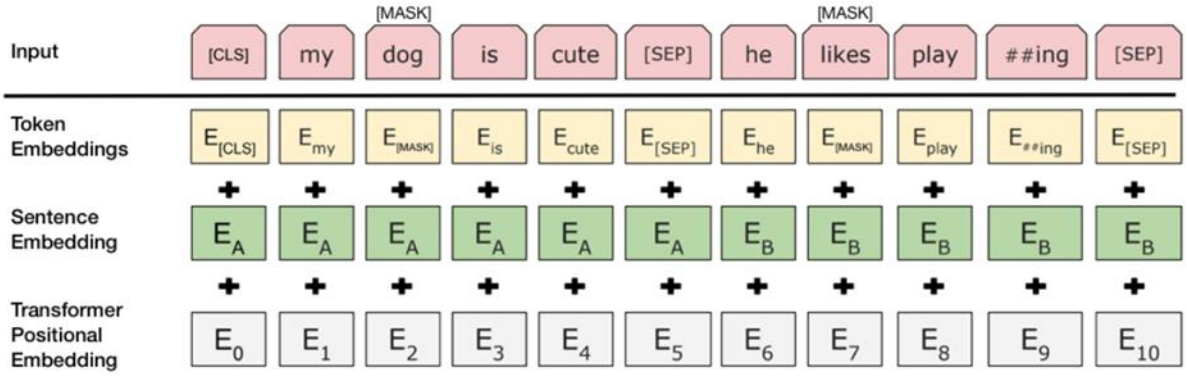


Fig.7

As shown in Fig.7, the input vector Word Embeddings are represented by the combination of three embeddings: Token, Segment, and Positional. The Token embedding is the pre-trained vector embedding for the token in the vocabulary of the textual data. The Segment embedding is for adding information about the specific sentence in the sequence that the token belongs to. The Positional embedding captures token's location in the sentence. Segment and Position embedding maintains

information about the sequence of clauses and the words' place in the sentence. The produced Embeddings are passed to the BERT stacked Encoder blocks. The outputs of the Encoder blocks are all the same size and are generated simultaneously as captured in Fig. 6.

3.2.3 Experimental Set Up

This work utilised 80:20 split for training and validation objectives. We used a publicly available dataset of tweets containing both sarcastic and non-sarcastic statements, created by Joshi et al. (2017). The important libraries were loaded. Tokenization was carried out by splitting the sentences into words with vocab size of 20900. This was followed by padding in encoding format. The transformer block was initiated in layers using a transformer-encoder function built on multi head units. This is implemented in a feed forward propagation linearly with a normalization across the layers were done at epsilon(1e-6) and a dropout rate of 0.3 to avoid overfitting. The embedding layers were implemented using the token and position embeddings which was built using a function within the build-up. The classifier model was initiated using the already developed transformer layer which outputs each vector across the input sequence. Here the mean across the time sequence is taken and a feed forward being applied on it for text classification. The input is designed using global average pooling with a dropout of 0.35 and activation function (sigmoid). The model is optimized using adam optimizer with a learning rate reduction factor within the scheduler.

3.2.4 Evaluation Metrics for Sarcasm Detection

For the purpose of analysis and comparison on this work for sarcasm detection, further study on the results were done as stated below in Fig.8 (Kundu R., 2013). Accuracy is not the most informative metric for imbalanced classes and probabilistic prediction respectively as it might lead to misclassification problems. Thus, there is need to weigh in the following performance metrics for more comprehensive result.

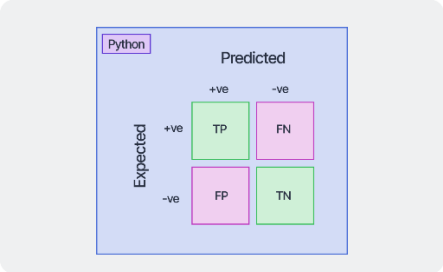


Fig.8

- **Accuracy** – gives the percentage of correctly classified instances among all instances within the text.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$ ..... Equation 1.

- **Recall** – gives actual positive instances that are correctly identified by the classifier.

$$Recall = \frac{TP}{TP + FN}$$

……. Equation 2.

- **Precision –** measures the portion of instances accurately classified as positive by the classifier among all instances that the classifier predicted as positive. It should be noted that high precision is prioritized even if it comes at the cost of lower recall or accuracy.

$$Precision = \frac{TP}{TP + FP}$$

……. Equation 3.

- **F-measure** – harmonic average of precision and recall mostly useful for imbalanced categorization.

$$F\text{ - }measure = \frac{2*Recall*Precision}{Recall + Precision}$$

……. Equation 4.

3.2.4.1 Result Evaluations

We evaluated the performance of the above model in conjunction with Bi-RNN with embeddings on the grounds of accuracy, precision, recall, F1-score, ROC-AUC score respectively and taking into consideration error function during the models' training. Our results show that BERT did better than others, achieving an accuracy of 0.94, F1-score of 0.94 with a ROC-AUC score of 0.9737. The simple embedding on global average pooling achieved an accuracy of 0.85, F1-score of 0.87, ROC-AUC of 0.9452 while the pretrained BI-RNN (GRU) achieved an accuracy of 0.94, F1-score of 0.94 with ROC-AUC of 0.9721.

Though during the training, we noticed that the BERT performs better due to its negligible error function of 0.33 as against 0.76 of BI-RNN (GRU). Findings are even with previous works that have shown the effectiveness of BERT in various NLP tasks (Devlin et al., 2019). Table 2 below shows summarized results.

S/N	MODEL	ACCURACY	LOSS	F1-SCORE	PRECISION	RECALL
1	**Glove Embedding**	**0.89**	**0.46**	**0.86**	**0.78**	**0.96**
2	**Bi-RNN(GRU)**	**0.94**	**0.76**	**0.94**	**0.92**	**0.96**
3	**BERT1**	**0.94**	**0.33**	**0.94**	**0.92**	**0.96**
4	**BERT2**	**0.94**	**0.34**	**0.93**	**0.90**	**0.97**

Table. 2

We were able to draw a conclusion that BERT is the best model for this classification task considering its high-performance metrics as indicated above and ROC-AUC score (Narkhede S., 2018) as shown in table 3 below.

S/N	MODELS	AUC SCORES
1	Simple Baseline Embedding	0.9452
2	Pre-trained Bi-RNN(GRU)	0.9721
3	BERT	0.9737
4	BERT2	0.9724

Table 3.

3.2.4.2 PLOT ANALYSIS – Below are some plots for more details**.**

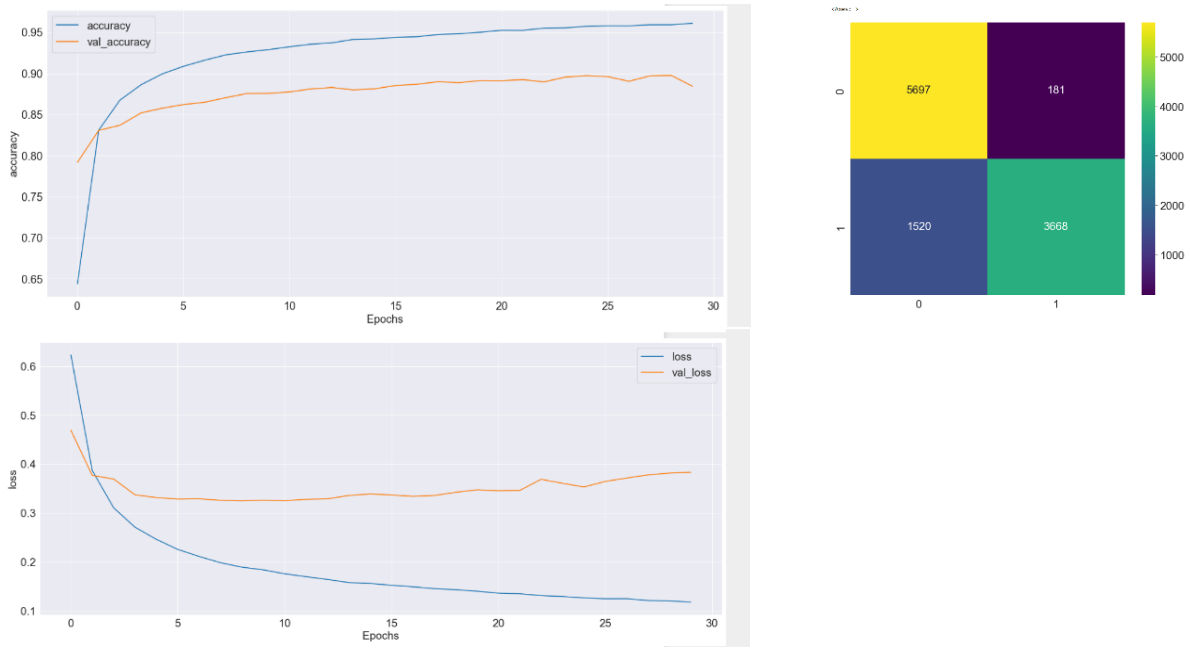


Fig.9 Simple model with embedding.

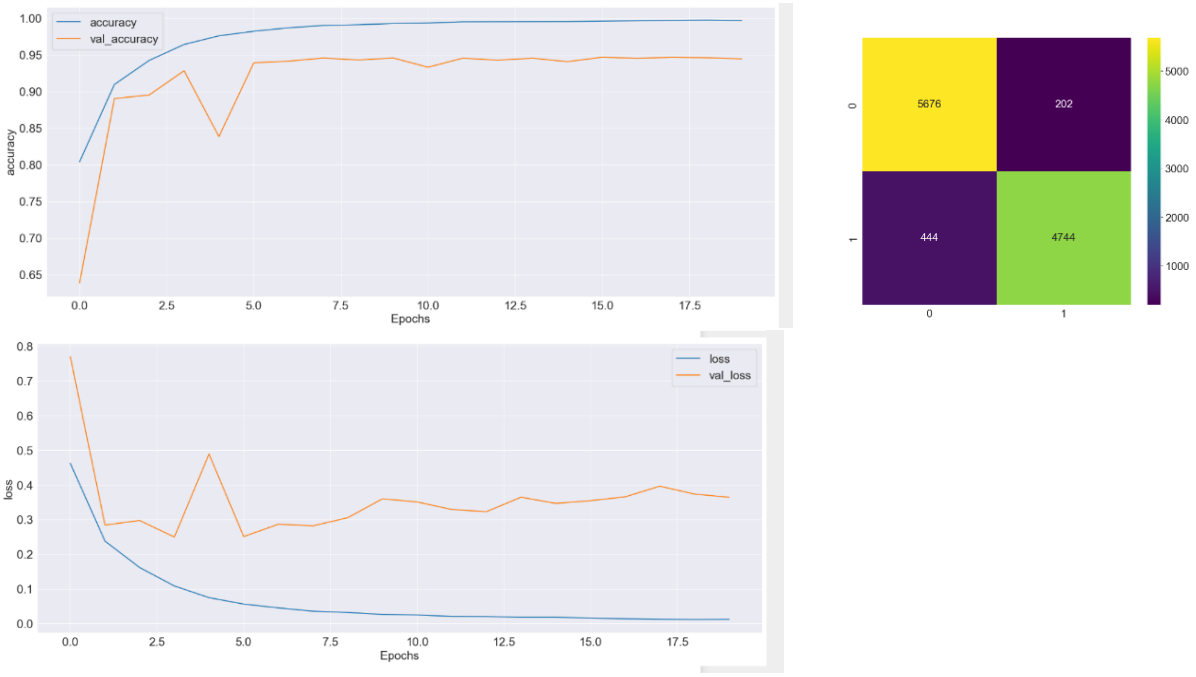


Fig. 10 Pre-trained Bi-RNN(GRU)

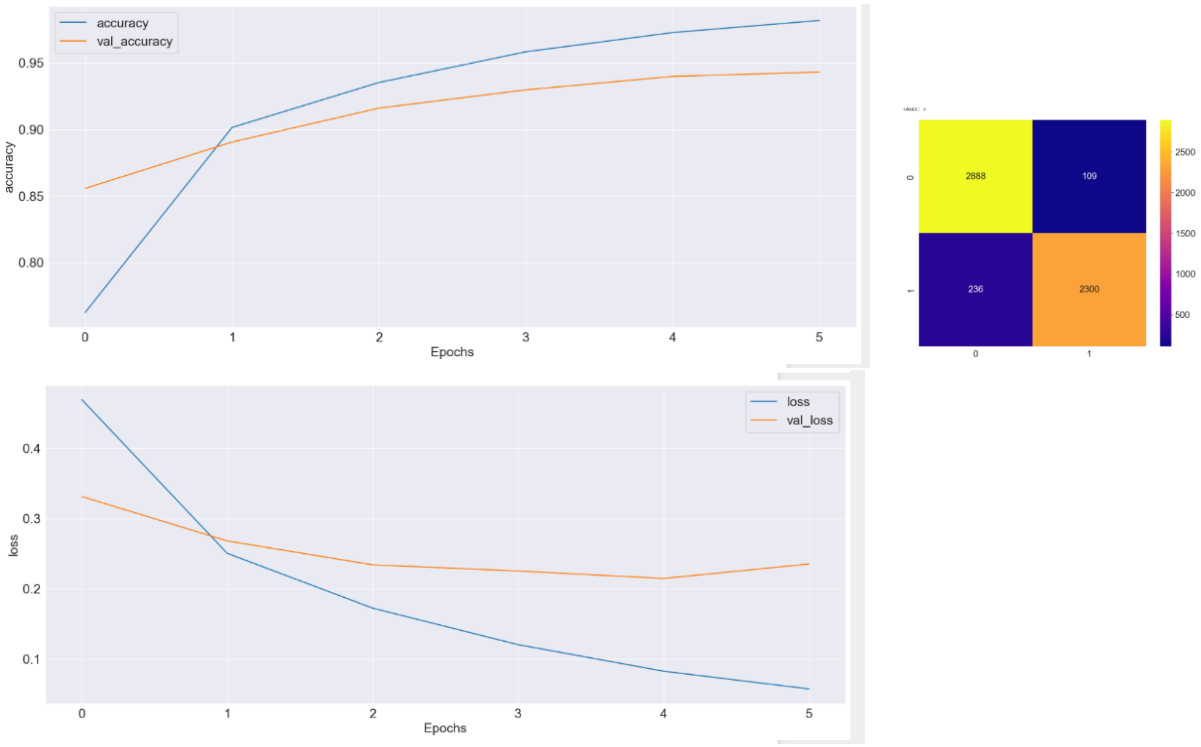
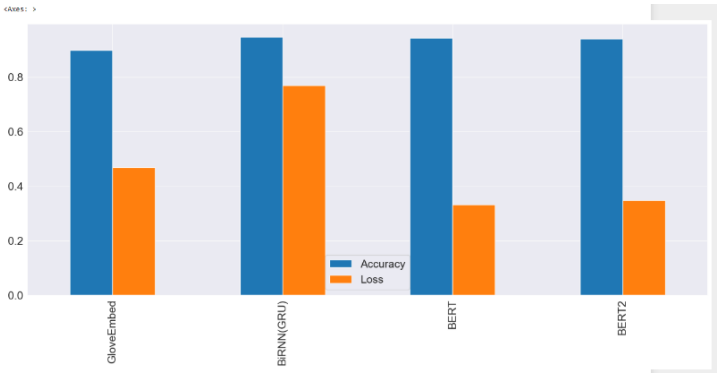


Fig. 11 BERT (Multi-head)



	Accuracy	Loss
GloveEmbed	0.897524	0.469188
BiRNN(GRU)	0.947045	0.769742
BERT	0.943069	0.331191
BERT2	0.940539	0.347400

Fig. 12 Bar chart representation

4.1 CONCLUSION

Our result analysis demonstrates the effectiveness of BERT as a deep learning model for sarcasm identification in an imbalanced textual dataset. BERT outperformed pretrained (GRU) model with embedding and simple model with embedding, achieving a significantly higher Accuracy, F1-score, precision and AUC-score respectively. Its trivial error function during training proves its significance in NLP tasks.

4.2 FUTURE WORK

This project provides some useful insights into opinion mining especially when it comes to contextual tweets. Several approaches and techniques are then used to set up a framework that is effective for sarcasm recognition. This work reveals the majority of BERT architecture for detection of sarcastic comments. However, there is still room for improvement for further calibrations and feature extractions. Fine tuning the BERT will be a good improvement on the model. Future work could involve incorporating multi-lingual contexts, voiced-based information and domain-specific knowledge to improve the performance of the models.

REFERENCES

Abulaish M., A. Kamal, and M. J. Zaki, 2020 - A survey of figurative language and its computational detection in online social networks,'' ACM Trans. Web, vol. 14, no. 1. 2018. In Proceeding IEEE/WIC/ACM International Conference on Web Intelligence (WI), IEEE.

Aquino J., 2012 ``Transforming social media data into predictive analytics,'' CRM Mag., vol. 16, no. 11, pp. 3842.

Bamman D. and N. A. Smith, 2015 - Contextualized sarcasm detection on twitter,'' in Proc. Int. AAAI Conf. Web social media. Vol. 9, Ninth International AAAI Conference on Web and Social Media.

BBC. “Hana Kimura: Netflix star and Japanese wrestler dies at 22.” In: 2020.

url: https://www.bbc.com/news/world-asia-52782235.

Barbieri F., H. Saggion, and F. Ronzano, 2014 pp. 50-58 - Modelling sarcasm in twitter, a novel approach,'' in Proc. 5th Workshop Comput. Approaches Subjectivity, Sentiment social media Anal.

Baruah A., K. Das, F. Barbhuiya, and K. Dey, 2020 - Context-aware sarcasm detection using bert,'' in Proc. 2nd Workshop Figurative Lang. Process.

Bharti S. K., B. Vachha, R.K. Pradhan, K.S. Babu and S.K.Jena, Sarcastic Sentiment Detection in Tweets Streamed in Real time: A BigData Approach, Digital Communications and Network S2352-8648(16)30027-X

Bouazizi M. and T. Otsuki Ohtsuki, 2016 - A pattern-based approach for sarcasm detection on Twitter,'' IEEE Access, vol. 4, pp. 5477-5488.

Davidov D., O. Tsur, and A. Rappoport, 2010 Semi-supervised recognition of sarcasm in twitter and amazon,'' in Proc. 14th Conf. Comput. Natural Lang. Learn., pp. 107116.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

Ebrahimi M., A. H. Yazdavar, and A. Sheth, 2017 Challenges of sentiment analysis for dynamic events,'' IEEE Intell. Syst., vol. 32, no. 5.

Fellbaum C., 2010 Wordnet,'' in Theory and Applications of Ontology: Computer Applications. Dordrecht, The Netherlands: Springer.

Hazarika D., S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, ``CASCADE: Contextual sarcasm detection in online discussion forums,'' 2018, arXiv:1805.06413. [Online]. Available: http://arxiv.org/abs/1805.06413.

Ili¢ S., E. Marrese-Taylor, J. A. Balazs, and Y. Matsuo, ``Deep contextualized word representations for detecting sarcasm and irony,'' 2018, arXiv:1809.09795. [Online]. Avalable: http://arxiv.org/abs/1809.09795

Joshi, A., Sharma, A., & Bhattacharyya, P. (2017). "Oh... So You Think I'm Dumb?" A Dataset and Analysis of Idiotism in Online Conversations. Proceedings of the 11th International Conference on Natural Language Processing (ICON-2017), 130-136.

Kumar Avinash, Vishnu Teja Narapareddy, Veerubhotla Aditya Srikanth, Aruna Malapati, and Lalita Bhanu Murthy Neti - Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM Birla Institute of Technology and Science at Pilani, Hyderabad 500078, India Corresponding author: (p20150507@hyderabad.bits-pilani.ac.in)

Margarita Zargaryan, Violeta Iskandaryan 2021 - Transformer Neural Networks for Natural Language Processing.

Mehndiratta P, Sachdeva S, Soni D (2017) Detection of sarcasm in text data using deep convolutional neural networks. Scalable Computing: Practice and Experience 18(3):219–228

Murk Asad, Fairouz Sharif 2018 - https://www.kaggle.com/code/murkasad31/sarcasm-detection-glove-word2vec-lstm-gru-murk?cellIds=95&kernelSessionId=119000995

Norman A. A., C. I. Eke, and L. Shuib, "Context-Based Feature Technique for Sarcasm Identification in Benchmark Datasets Using Deep Learning and BERT Model," in IEEE Access, vol. 9, pp. 48501-48518, 2021, doi: 10.1109/ACCESS.2021.3068323.

Pak A. and P. Paroubek, 2010 Twitter as a corpus for sentiment analysis and opinion mining,'' in Proc. LREc, vol. 10.

Poria S., H. Peng, Majumder N., N. Chhaya, E. Cambria, and A. Gelbukh, 2019 Sentiment and sarcasm classication with multitask learning,'' IEEE Intell. Syst., vol. 34, no. 3.

Poria S., E. Cambria, D. Hazarika, and P. Vij, ``A deeper look into sarcastic tweets using deep convolutional neural networks,'' 2016, arXiv:1610.08815. [Online]. Available: http://arxiv.org/abs/1610.08815

Priya Goel & Rachna Jain & Anand Nayyar & Shruti Singhal & Muskan Srivastava 2022 - Sarcasm detection using deep learning and ensemble learning.

Rajadesingan A., R. Zafarani, and H. Liu, 2015 Sarcasm detection on Twitter: A behavioral modeling approach,'' in Proc. 8th ACM Int. Conf.Web Search Data Mining.

Razali Saifullah MD, Alfian Abdul Halin, Lei Ye, Shyamala Doraisamy, (Member, IEEE), AND Noris Mohd Norowi 2021 Sarcasm Detection Using Deep Learning with Contextual Features Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Seri Kembangan 43400 Australia Corresponding author: Alan Abdul Halin (alan@upm.edu.my).

Reyes A., P. Rosso, and D. Buscaldi, 2012 From humor recognition to irony detection: The gurative language of social media,'' Data Knowl. Eng., vol. 74, pp. 112.

Riloff E., A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, 2013 - Sarcasm as contrast between a positive sentiment and negative situation,'' in Proc. Conf. Empirical Methods Natural Lang. Process., pp. 704714

Rohit Kundu 2013 https://www.v7labs.com/authors/rohit-kundu

Sarang Narkhede 2018 -Understanding Confusion Matrix https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Shmueli B., L.-W. Ku, and S. Ray, ``Reactive supervision: A new method for collecting sarcasm data,'' 2020, arXiv:2009.13080. [Online]. Available: http://arxiv.org/abs/2009.13080

Zhang Y., M. J. Er, N. Wang, M. Pratama, and R. Venkatesan, 2016 Sentiment Classification Using Comprehensive Attention Recurrent Models, pp. 15621569 DO - 10.1109/IJCNN.2016.7727384