A Deep Learning Approach for Sarcasm Detection in User generated Content

E. Deepak Chowdary[1], B. Naga Sudheer[2], K. Santhi Sri[3], P. Radha Madhavi[4]

[1,2,3,4]School of Computing and Informatics, Vignan's Foundation for Science, Technology & Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

Abstract: Using sarcasm in social media is a common way to express negative opinions using positive language, making identifying sarcasm an essential part of the sentimental analysis. Identifying sarcasm is approached as a two-class classification problem (Binary). Both deep learning models and traditional models have been developed using features such as lexical, semantic, and pragmatic elements. However, sarcasm can be challenging to detect in natural language processing as it involves language usage that is not always straightforward. Despite this, detecting sarcasm can be valuable in many contexts , which includes social media tracking or monitoring, sentimental analysis, and customer support. This research paper proposes a novel approach, BILSTM-GRU architecture, which uses text representations to learn difficult patterns and semantic structures in the text for identifying the sarcastic data. The approach which is going to propose has the ability to improve the accuracy of detecting sarcasm which contributing towards sentiment analysis on social media platform.

Keywords: Deep Learning, BILSTM, GRU, Sarcasm, tokenization, word embeddings, optimization, binary classification

Introduction

Detecting sarcasm in a text has several practical applications, particularly in analysing social media content, customer related feedback analysis, and e-commerce reputation management. For instance, identifying sarcastic comments in customer feedback can help businesses to improve their products or services and address customer grievances more effectively. Using sarcasm detection in sentiment analysis can develop the precision of sentiment classification, rendering it more adaptable for a diverse array of applications. The latest developments in deep learning [31] has demonstrated excellent outcomes in the field of sarcasm detection. There are different types of deep learning architectures have been developed. Some of these architectures include Convolutional Neural Networks (CNNs), which are good at analysing images, and Recurrent Neural Networks (RNNs), which help predict data sequences. Some architectures combine elements of both CNNs and RNNs. These models trained on large-scale datasets of labelled sarcastic and non-sarcastic expressions to learn the patterns and features distinguishing a sarcastic language from a literal language. In addition, sarcasm detection has also attracted attention from the computational linguistics community, leading to the development of specialised resources such as annotated datasets, lexicons, and rule-based systems. The above provides a solid base for developing and evaluating sarcasm detection methods and can help researchers better understand the linguistic properties and mechanisms of sarcasm. For example, "Zombies eat brains. You are safe" this contradiction between the actual condition of "Zombies eat brains" and the statement's content "safe" is clearly shown. Sarcasm is a type of sentiment analysis involving expressing a sentiment that contrasts with or varies from its regular expression, usually with an ironic or mocking tone. The ability to detect sarcasm in the text has become increasingly crucial in sentiment analysis, particularly for analysing vast social web data.

The problem of sarcasm detection involves classifying text as either sarcastic or not and is achieved through various feature extraction techniques and deep learning architectures. Deep learning architectures like BILSTM-GRU [31] prove notably efficient for natural language processing tasks, thanks to their capacity for learning from examples and managing extensive data sets without requiring manual feature engineering. The BILSTM with a multi-headed attention layer and GRU is one such model that has shown promising results in sarcasm detection. This model combines different types of recurrent neural networks to capture different aspects of the input text. The BILSTM layer, for example, is used to analyse the sequential dependencies of the text, and the multi-headed attention layer enhances the model's concentration on the most pertinent segments of the text. Meanwhile, the GRU layer captures contextual information from the text by taking into account the history of preceding words and phrases. The results from both the BILSTM and GRU layers are combined. The integration of recurrent neural networks delivers a comprehensive depiction of the input text. Here, the authors evaluate this model on two public datasets and compare the outcomes to other cutting-edge models regarding F1-score, recall, precision and accuracy. Using this technique, the model can significantly improve sentiment analysis performance. GLOVE [32] proposed generating word embeddings for sarcasm detection, a count-based model that generates a table of word vectors. These allocated vectors to each tokenised term sentence and correctly padded them to build a matrix. Then, passed the matrix through a BILSTM layer and GRU network to generate feature representations. These representations were combined and were fed through a convolution layer equipped with filters to produce an activation map. The RELU layer introduced nonlinearity to this map, creating a rectified feature map. K-max pooling generates a pooled feature map using the best k-features from hidden layers. Finally, a fully connected sigmoid layer determines each output phrase's likelihood and identifies the tweet as sarcastic or not. Developing effective sarcasm detection methods requires using various feature extraction techniques, deep learning architectures, and large-scale datasets of labelled sarcastic and non-sarcastic expressions. The ultimate objective is to improve the understanding of sarcasm's linguistic properties and mechanisms and develop methods to accurately identify the sarcastic language in various contexts.

Related Work

Ren, Y. et al. [1] evaluated a CNN model on the Twitter dataset and extracted contextual-augmented features using Glove. The model captures semantic features well in a conversation-based context and achieves a macro F-score of 62% on Wang Tweets dataset. Kumar A. et al. [2] developed the combined model based on attention-based BILSTM and CNN by extracting various levels of inputs from the Sem-Eval dataset resulting in an accuracy of 72%. Majumder N. et al. [3] proposed a hybrid based model employing BILSTM and Convolutional Neural network for multitask learning. They evaluated it on two independent datasets, the Stanford Sentiment Treebank and the Sarcasm Corpus dataset. This model produced better classification accuracy of 71% and 82% compared to other state-of-art-models. Di Gangi et al. [4] designed a hybrid model with SVM, Naïve bias and Logistic regression by extracting distributed representation of words in high-dimensional space and is evaluated on the Twitter dataset. Sonawane S. et al. [5] have developed Decision Tree for identifying sarcasm from the Twitter dataset resulting in an accuracy of 72.45%. Ren L. et al. [6] designed a multi-level memory network layer to store long-term dependencies with sentiment semantics. The architecture is based on LSTM and evaluated on Sem-Eval and Corpus data sets, which obtained an accuracy of 75.56% and 82.39%. Del Pilar et al. [7] provided a comprehensive review of the existing literature on sarcasm, highlighted the potential benefits of using figurative languages in social media, such as increased engagement, emotional impact, and improved communication, and discussed the challenges and limitations. A. Zhao et al. [8] proposed a model using knowledge-enabled BERT (KE-BERT), which incorporates external knowledge sources for increasing the precision related to sentiment analysis. The model tested on several benchmark datasets and achieved an accuracy of 87%. Bagate R. et al. [9] focused on various features and algorithms used in existing models and provided an overview of their strengths and limitations. Vithyatheri Govindan et al. [10] calculated multiple Machine Learning models and identified Random Forest model outperformed others with 77% accuracy from a negative sentiment tweet dataset containing hyperboles. Wen et al. [11] developed a hybrid based model which consists of LSTM and Convolutional Neural network, with Sememe and also the Auxiliary features taken out to improve performance, tested on the SARC and NLPCC data sets. The models' accuracy percentages were 76.5% and 77.34%, respectively.

Z. Keivanlou-Shahrestanaki et al. [12] developed an attention-based neural network evaluated on Corpus and Sem-Eval data sets to extract semantic features. Frenda S. et al. [13] studied the effects of sarcasm on emotional state and inventiveness without using machine learning or statistical models. Cruz et al. [14] proposed extracting complementary features like character n-grams and sentiment analysis scores combined with SVM and Random-Forest and then compared performance measures. Sharma et al. [15] proposed a MOH (Multi-Objective-Hybrid) technique to integrate SVM and Deep Neural Networks (DNN) with lexicon-based methods to detect hate speech and linguistic features extracted from a Twitter dataset. Low et al. [16] proposed a BERT transformer model with 12 transformers and 110 million parameters on a corpus dataset to differentiate between fake news and satire and achieved an accuracy of 94.8%. Kumar, P. et al. [17] proposed a text emotion recognition system that uses BERT using a contextual channel and a lexical channel on Emo-Bank and Sem-Eval data sets and achieved high F1 scores compared to baseline methods.

Chen W. et al. [18] proposed a joint learning approach to extract sentimental clues and context incongruity from a Twitter corpus and news headlines datasets. Jeyakarthic et al. [29] proposed a BILSTM model. It extracts the embeddings from inputs. The model is assessed on Sentiment140 and also Sem-Eval datasets. The model achieved a precision of 86% and 68.32%. Goel et al. [20] developed a model with the combination of CNN, LSTM, GRU by using an attention mechanism to assign weights to different words. The model is evaluated on news corpus datasets, and it noted that embeddings and attention played a crucial role, reporting an accuracy of 89.4%. Barhoom et al. [21] evaluated a model which is a combination of CNN and LSTM; studies about the impact of dataset size and training iterations by training a model on news headings dataset resulted in an accuracy of 71%. Nayak et al. [22] proposed a model with the combination of CNN and RNN by identifying salient words and capturing long-term dependencies. The model has evaluated on news headlines dataset, producing better results than baseline models. Bhardwaj et al. [23] developed a hybrid model that is a combination of BERT and SVM by extracting contextualised features with BERT, and SVM did classification. The model is trained on sarcasm corpus and Sem-Eval data sets and achieved 73% and 78% accuracy. Jayaraman A. et al. [24] developed a model for supervised learning by extracting lexical and semantic features. The model was assessed on news headings dataset with an accuracy of 88%.

Khan S. et al. [25] developed a model on BILSTM and CNN to detect hate speeches and evaluated them on two balanced data sets and one unbalanced class dataset. The model worked well with the data which is unbalanced with an accurate mark of 97% and reported an accuracy of 76% on balanced data. Misra, R et al. [26] proposed an Attention-based neural network architecture on news Headings dataset and the performance of several algorithms are compared accordingly, finding Random Forest algorithm stood best with an accuracy of 84%. Mohan et al. [27] developed a hybrid model by combining BERT and GCN on sarcasm corpus and news headlines data sets, extracting contextualised features, with an outstanding accuracy of 86.4% and 85.82%, respectively. Zhao, G. et al. [28] proposed Multitask learning with aspect extraction and sentiment classification on Sem-Eval datasets and achieved well-performing accuracy of 85.2%. Zhang, Y. et al. [29] proposed the model against baselines using a dataset of conversations containing text, audio, and visual cues, finding that their model outperforms the baselines with an accuracy of 84% for sarcasm detection, 78% for sentiment analysis, 76% for emotion recognition and presents a promising approach to multi-modal sarcasm, and recognition of emotions. Thakur S. et al. [30] provided a literature review about aspect-based sentiment analysis, encompassing both approaches implicitly as well as explicitly by examining the methods' merits and drawbacks.

Proposed Methodology

The research aims to develop a hybrid based model with the combination of BILSTM (BILSTM stands for Bidirectional Long Short-Term Memory) and GRU (Gated Recurrent Unit) for detecting sarcasm from News headings and IMDB reviews.

Dataset Collection: Prior research on detecting sarcasm has predominantly relied on datasets of Twitter with hashtag-based labels, which are often plagued by noise in terms of labels and language. Moreover, detection of sarcasm in tweet replies need access to contextual related tweets. We collected a dataset related to news headings for detecting sarcasm from news websites for evaluating the proposed model. The Onion, which produces satirical versions of current events, provided headlines for the sarcastic category, specifically News in Brief and News in Photos. Meanwhile, HuffPost provided real, non-sarcastic news headlines. We also collected the IMDB reviews dataset containing user reviews on various movies. Both datasets have two categories of classes: sarcastic and non-sarcastic.

Dataset	Sentences	Sarcastic	Non Sarcastic
News headlines	28619	14334	14285
IMDB Spoiler	573913	150924	422989

Table 1: Data Presentation in Datasets

Tokenization: Tokenisation is a fundamental step in NLP that helps standardise text data to be analysed and processed by machine learning algorithms. Tokenisation breaks down a sentence into multiple words or tokens. These tokens are converted into lowercase, and then punctuation marks are removed. The resulting text splits into individual words or tokens based on whitespace or other delimiters like commas or periods. In some cases, there is a need of extra pre-processing phases may be required to normalise text, such as removing common words such as “and” , "the" which may not convey much significance or reducing words to their root form. Each token is assigned a unique identifier (e.g. an integer or a one-hot vector) that represents its position in the vocabulary. These identifiers were used to create the final representation of the vectors of the sentence.

Input: what a groundbreaking movie. I especially loved how they used every cliche in the book.

Output: “what”, “a”, “groundbreaking”, “movie”, “I”, “especially”, “loved”, “how”, “they”, “used”, “every”, “cliché”, “in”, “the”, “book”

Word Embeddings: Each and every word in the sentence is mapped towards dense vector in a vector space continuously , so that the similarly semantic words are closly together. The sentence is represented as the sum or average of word vectors. This approach signifies a sentence as vector with many dimensions, as each one related to a specific word in a sentence, resulting in a sparse vector.. The value within each dimension indicates the frequency of that particular word within a sentence. Once the sentences are represented as vectors, they can be fed into a machine learning model for sarcasm detection. The model will learn to identify patterns in the vector representations of sentences and use this knowledge to make predictions on new, unseen (or) test data.

Input: “what”, “a”, “groundbreaking”, “movie”, “I”, “especially”, “loved”, “how”, “they”, “used”, “every”, “cliché”, “in”, “the”, “book”

Output: [-0.302, 0.503, 0.468, -0.710, ...],[0.112, -0.103, 0.778, 0.046, ...],[0.404, 0.567, -0.910, 0.134, ...],[-0.102, 0.234, -0.761, -0.882, ...],[0.008, -0.212, 0.336, 0.262, ...],[0.405, -0.816, 0.104, -0.601, ...],[0.085, -0.876, -0.234, 0.984, ...],[-0.296, 0.213, -0.878, 0.129, ...],[-0.527, 0.105, -0.196, 0.903, ...]

These vectors feed into the different models for feature extraction and extracted features are classified using classifier.

Bidirectional Long-Short Term Memory (BiLSTM) Network

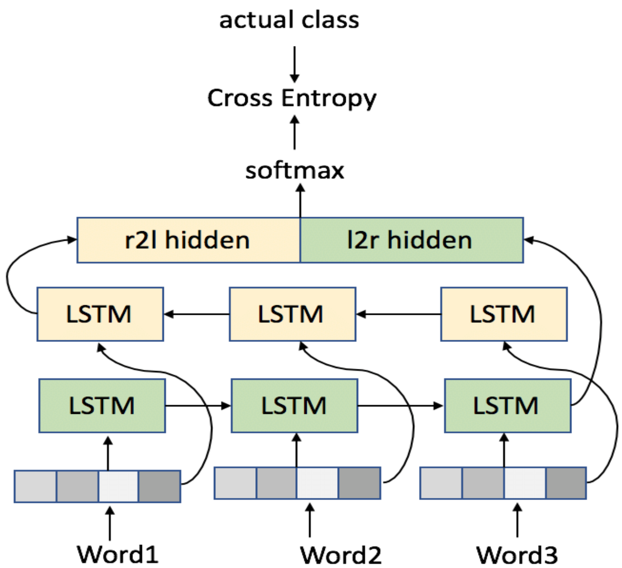

Fig. 1: BiLSTM Architecture

A BiLSTM is designed in a way for efficiently handling sequential data, which utilises a pair of LSTM models. BILSTMs analyse input sequence in forward as well as backward directions which provides more information for the network and improves its ability to understand the sequence context. GLOVE (Global Vectors for Word Representation), used to generate word embeddings which capture the meaning and syntax of words. GLOVE embeddings are commonly used as input to BILSTM networks in natural language processing tasks, like sarcasm detection. The GLOVE embeddings are typically fed into the first LSTM layer to process the sequence of word embeddings in a foreword direction and later in the backward direction.

Equations: Forgot-gate: $f_t = \sigma_g(w_f \times x_t + U_f \times h_{t-1} + \times)$

Input-gate: $i_t = \sigma_g(w_i \times x_t + U_i \times h_{t-1} + b_i)$

Cell-state: $c_t = f_t \times c_{t-1} + i_t \times tanh\ (w_c \times x_t + U_c \times h_{t-1} + b_c)$

Output-gate: $o_t = \sigma_g(w_o \times x_t + U_o \times h_{t-1} + b_o)$

Hidden-state: $h_t = o_t \times tanh\ (c_t)$

Where $'\sigma_g'$ is a sigmoid activation function, w stands for weights at different gates, b stands for bias, x stands for input at a particular time step, and $'U'$ stands for weights at the recurrent matrix.

Gated Recurrent Unit (GRU)

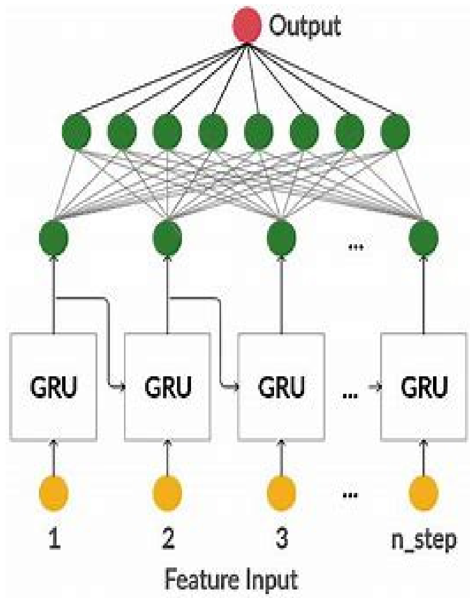

Fig. 2: GRU Architecture

GRU is mainly used for analysing data sequences. It was designed to solve a problem where the network would forget important information from earlier time steps due to vanishing gradients. The GRU has two gates that help the network remember or forget information selectively. The reset gate regulates the quantity of former information that will be ignored, while the update gate dictates the quantity of new data to be included in present state. The GRU can effectively analyse long data sequences and extract meaningful information from them using these gates. Similar to other RNNs, GRUs, possess a hidden state that undergoes updates at each timestep, influenced by both the input and the preceding hidden state. The hidden state is then used to make predictions or generate output

Equations:

Update gate: $u_t = \sigma(x_t \times U_u + H_{t-1} \times w_u)$

Reset-gate: $r_t = \sigma(x_t \times U_r + H_{t-1} \times w_r)$

Candidate Hidden State: $H_t' = tanh\ (x_t \times U_g + r_t \oplus h_{t-1} \times w_g)$

Hidden-State: $H_t = u_t \oplus H_{t-1} + (1 - u_t) \oplus H_t'$

Here U and w are weight matrices, σ is a sigmoid activation function and x stands for input at a particular time step.

Proposed Approach

The proposed Hybrid model has a combination of two customised designed models, with number of layers which has the property to detect sarcasm emotion correctly. The proposed model is trained individually, and their outputs are concatenated to capture both contextual and semantic features. BiLSTM uses GLOVE to get word embeddings and then train them individually.

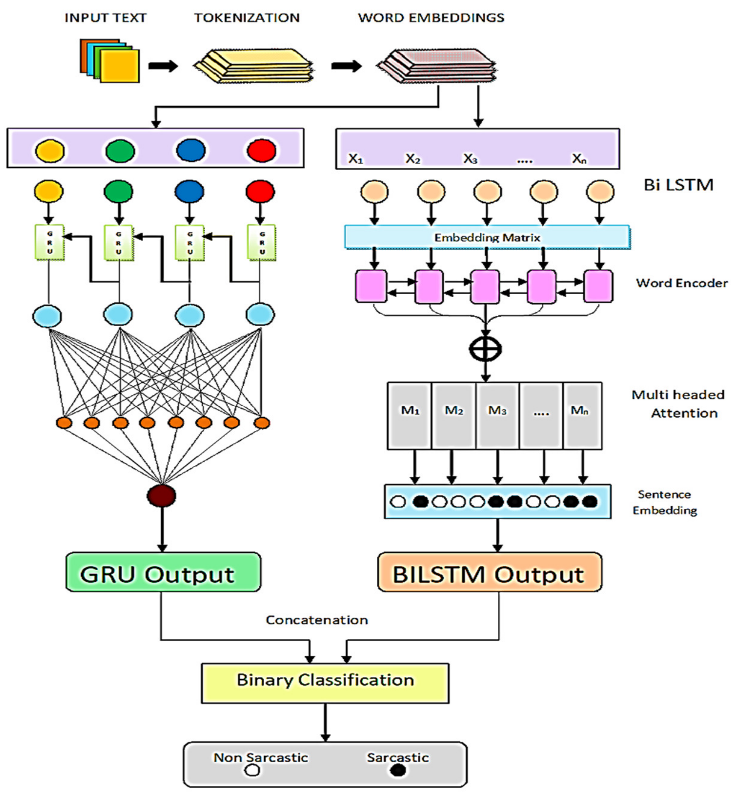

Fig. 3 : BILSTM+GRU Architecture

Feature Extraction

Both the GRU and BILSTM models can be used to extract features from sequences of text data. BILSTM is used to extract contextualised features from the sequences and finally applies a dense layer. In contrast, GRU can extract subtle and contextual information in the input sequence, such as the tone of voice and implied meanings in sarcastic statements. These extracted contextualised and subtle features are used for the sentimental analysis and classification of data. Here, BILSTM effectively captures contextual information from past and future contexts as it has a memory unit, and GRU is good at solving the vanishing gradient problem. Both models can handle inputs of variable length.

Multi-headed attention layer

A sarcastic comment may contain words with multiple meanings or use a particular tone of voice that can be difficult to interpret. Multiple heads of attention can be used in natural language processing models to address these challenges. Each head of attention focuses on a different aspect of the comment and gives appropriate importance to each word based on these factors. It allows the model to represent the overall semantics of the comment more nuancedly and helps to capture the different elements of sarcasm that may be present. These two models combine by concatenating their individual outputs along the last dimension, the feature dimension. The resulting tensor has a shape related to batch size and the number of features. A sigmoid activation function is applied to perform binary classification on this concatenated output to transform the output values into a range between 0 and 1. The output is then classified as sarcastic or non-sarcastic based on a threshold value. By combining the BILSTM and GRU models, the resulting model can capture a wider range of linguistic features and context than either model alone. It has the potential to enhance the task performances such as sarcasm detection in text.

Algorithm: Sarcasm detection using BiLSTM & GRU

Input: Pre-processed data

Output: Given sentence is sarcastic or Non-Sarcastic

Begin

1. For each sentence s1 do

   Tokens(t_1) ← Tokenization (s_1)

2. For each token(t_1) do

   Embedding vector(e_1) ← embedding(t_1)

3. BILSTM(bl1) ← embedding vector of each sentence
4. Feature extraction(F_{e1}) ← Multi Headed Attention (BILSTM(bl1))
5. GRU(g_1) ← embedding vector of each sentence
6. Feature extraction(F_{e2}) ← GRU(g_1)
7. Concatenation of features (c_f) ← F_{e1}+ F_{e2}
8. Dense(Sigmoid) ← Concatenated features(c_f)
9. Binary classifier(b_1) ← Dense(Sigmoid)
10. If $b_1 > 0.5$ then

    return Non-sarcastic sentence

11. Else

    return sarcastic sentence

End

Binary Classification

The second proposed model incorporates pre-processing and optimal feature extraction techniques. The acquired datasets (News headlines datasets and IMDB spoiler) are pre-processed datasets. Even though there are some concerns about the stopwords, duplication, and lemmatisation, which may cause an impact on model accuracy. In the proposed model, the extra noise will also be removed manually, and this optimised data is converted into tokens and then to word embeddings.

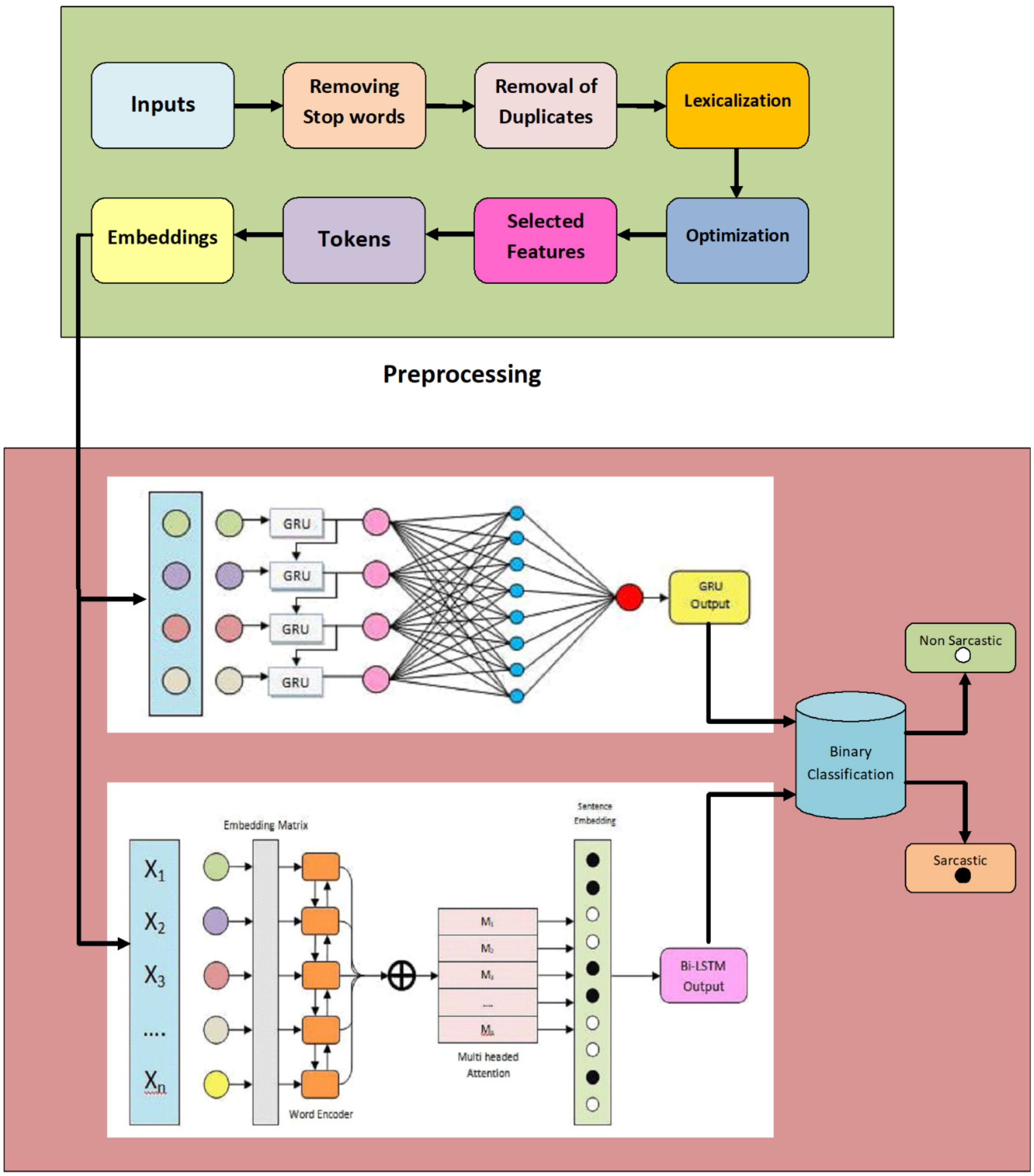

Fig. 4: Architecture for Binary Classification performed by proposed method

Stopwords: Stop Words are considered to have little or no significance in carrying the actual/original meaning of a sentence. These words are function words like prepositions, conjunctions, articles, and pronouns. The reason for removing stopwords mainly aim at reducing dimensionality of text data being analysed, which can support in improving the accuracy of specific NLP tasks like Text categorisation, sentimental evaluation, and modelling of topics are some of the techniques used. By eliminating these frequently occurring terms, the emphasis can be placed on more significant and informative words that can provide insights and comprehension about the text's substance. Example: "the", "a", "an", "and", "or", "of", "to", and "in".

Duplicates: Duplicates can help mitigate the impact of noise and increase the quality of data analysis in some circumstances, but they may additionally exacerbate noise and lead to incorrect outcomes in others. If the noise in the data is systematic and connected with the values being duplicated, then the duplicates can amplify the noise and lead to erroneous results. Duplicates may introduce bias into data processing in some situations.

Lemmatisation: Lemmatisation is a method in natural language processing that transforms the words into fundamental or root form, referred to as the lemma. The technique involves analysing a word's structure and context inside a sentence in order to discover the most basic and meaningful form of the word. This can help increase analysis accuracy and minimise processing complexity by decreasing the number of phrases in a document. Overall, lemmatisation is a useful technique for making sense of unstructured textual data in several applications, including information retrieval, sentiment analysis, and machine translation.
The pre-processed data is tokenised and turned to embeddings produced from the tokens, and it then proceeds in the same manner as the proposed approach 1.

Experimental Results

We conducted a series of experiments with deep learning models to evaluate their performance. F1-score, precision, recall and confusion matrix served as metrics for measuring their accuracy. These evaluation metrics are explained in detail in a section of the report. For a structured presentation of our experimental outcomes, we segmented this section into two parts, each dedicated to one of the two datasets used. Presented below are the performance outcomes of the deep learning models for each dataset.

Table. 2: Previous model accuracy for News headline & IMDB spoiler dataset

Models	News Headline Dataset			Imdb Spoiler Dataset		
	P	R	F	P	R	F
CNN+LSTM	0.96	0.94	0.95	-	-	-
BERT+GCN	0.89	0.88	0.87	-	-	-
CNN	0.82	0.85	0.83	-	-	-
BILSTM+CNN	0.95	0.92	0.93	-	-	-
BILSTM	0.82	0.81	0.81	0.75	0.74	0.75
BERT	0.91	0.90	0.91	0.81	0.83	0.82
BILSTM+GRU	0.96	0.96	0.96	0.93	0.93	0.93
Pre-processed BILSTM+GRU	0.93	0.94	0.89	0.85	0.87	0.84

Table. 3: comparison between different models

Approach	Algorithm	News Headline	IMDB spoiler
Jayarama et al.	Supervised Learning	88%	-
Barhoom et al.	CNN+LSTM	71%	-
Misra et al.	BILSTM+CNN	89.7%	-
Mohan et al.	BERT+GCN	85.2%	-
Misra et al.	CNN	87%	-
Proposed-1	BILSTM+GRU	84%	74%
Proposed-3	BILSTM	81%	77%
Proposed-4	BERT	93%	72%

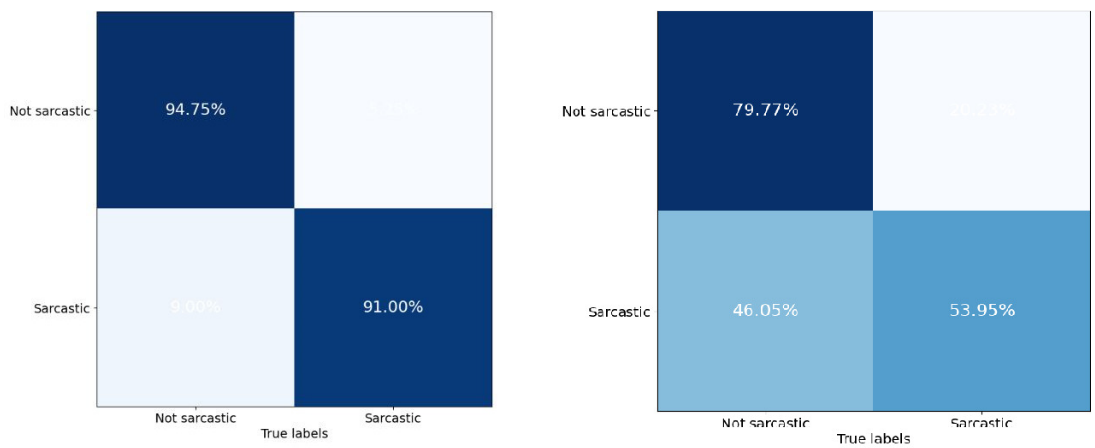

Fig. 5: Confusion matrix of News headings & IMDB Spoiler datasets

Experiments with news headline dataset

The conducted experiments using different deep learning models for classifying datasets into sarcastic or non-sarcastic. The outcomes of these experiements are represented in Table 2. Among all the classifiers tested, the BILSTM+GRU achieved the highest performance on the news headline dataset with 0.96 F1-score which indicates this model could precisely identify sarcastic and non-sarcastic statements. Alternatively, BILSTM classifier attained the lowest performance on news headings dataset with 0.81 F1-score, which suggests that this model might not be the optimal selection for classifying sarcasm within text. The proposed BILSTM, BERT, BILSTM+BERT, and pre-processed BILSTM+GRU models achieved F1 scores of 0.81, 0.91, 0.96, and 0.89. These models achieved accuracies of 81%, 93%, 84%, and 85.6%, respectively. The results suggest that BERT and BILSTM+GRU models exhibit greater efficacy in detecting sarcasm within text. The confusion matrix provides further insights into the models' performance. The authors observed that the BILSTM+GRU model correctly identified 91% of sarcastic and 94.75% of non-sarcastic.

Experiments with IMDB spoiler dataset

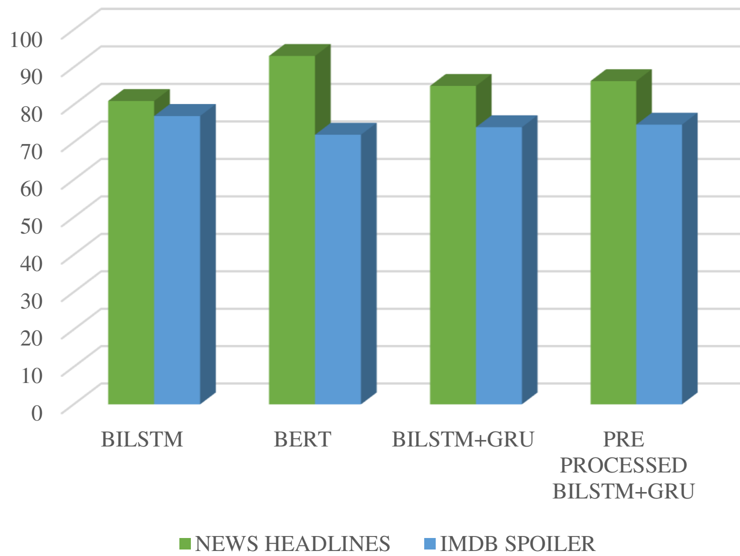

Fig. 6: Representation of accuracies

The BILSTM+GRU achieved the highest performance on the IMDB Spoiler dataset with 0.93 F1-score, where as BILSTM model achieved the lowermost performance on the IMDB Spoiler dataset with 0.74 F1-score. The accuracies attained by the models were 77%, 72%, 74%, and 74.7% for the BILSTM, BERT, BILSTM+GRU, and pre-processed BILSTM+GRU models, respectively. The F1 scores achieved by these models were 0.75, 0.82, 0.93, and 0.84, respectively. The results suggest that BILSTM+GRU model is the most effective in identifying sarcasm in text data. Additional insights into the models' performance are offered by the confusion matrix. The report shows that their proposed BILSTM+GRU models can successfully identify 54% of sarcastic instances and 80% of non-sarcastic instances. These results suggest that BILSTM+GRU model is most effective in sarcasm detection within text data. Utilizing deep learning models holds promise as an effective approach for sarcasm identification in textual content.

Conclusion

The two proposed models for detecting sarcasm use a combination of BILSTM and GRU algorithms. These techniques are used in both models for feature extraction, which gathers contextual and nuanced information from input sequences. The models differ in data pre-processing, with the second model removing stopwords, duplication, and lemmatisation to optimise the data before tokenisation and embedding. The outputs of BILSTM and GRU are concatenated in both models, which are then input into a dense layer and binary classifier for sarcasm classification. The models are predicted to detect sarcasm in text well. However, the performance may be influenced by the quality of the pre-processed data and the threshold value used for classification. However, the employment of BILSTM and GRU for feature extraction and the addition of multi-headed attention may lead to improved performance on detecting sarcasm in text. Overall, the presented models are an excellent starting point for further research.

References

[1] Ren, Y., Ji, D., & Ren, H. (2018). Context-augmented convolutional neural networks for twitter sarcasm detection. Neurocomputing, 308, 1-7.

[2] Kumar A, Sangwan SR, 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 7:23319–23328

[3] Majumder N, Poria S, Peng H, Chhaya N, Cambria E, Gelbukh A (2019) Sentiment and sarcasm classification with multitask learning. IEEE Intell Syst 34(3):38–43

[4] Di Gangi, M. A., Bosco, G. L., & Pilato, G. (2019). Effectiveness of data-driven induction of semantic spaces and traditional classifiers for sarcasm detection. Natural Language Engineering, 25(2), 257-285.

[5] Sonawane, S. S., & Kolhe, S. R. (2020). TCSD: Term co-occurrence based sarcasm detection from twitter trends. Procedia Computer Science, 167, 830-839.

[6] Ren, L., Xu, B., Lin, H., Liu, X., & Yang, L. (2020). Sarcasm detection with sentiment semantics enhanced multi-level memory network. Neurocomputing, 401, 320-326.

[7] del Pilar Salas-Zárate, M., Alor-Hernández, G., Sánchez-Cervantes, J. L., Paredes-Valverde, M. A., García-Alcaraz, J. L., & Valencia-García, R. (2020). Review of English literature on figurative language applied to social networks. Knowledge and Information Systems, 62(6), 2105-2137

[8] Zhao, A., & Yu, Y. (2021). Knowledge-enabled BERT for aspect-based sentiment analysis. Knowledge-Based Systems, 227, 107220.

[9] Bagate, R., Saini, A., Sethi, K., Tomar, H., & Singh, A. (2021, June). Sarcasm Detection and Explainable AI: A Survey. In Proceedings of the 3rd International Conference on Communication & Information Processing (ICCIP).

[10] Govindan, Vithyatheri and Vimala Balakrishnan. "A machine learning approach in analysing the effect of hyperboles using negative sentiment tweets for sarcasm detection." J. King Saud Univ. Comput. Inf. Sci. 34 (2022): 5110-5120.

[11] Wen, Z., Gui, L., Wang, Q., Guo, M., Yu, X., Du, J., & Xu, R. (2022). Sememe knowledge and auxiliary information enhanced approach for sarcasm detection. Information Processing & Management, 59(3), 102883

[12] Keivanlou-Shahrestanaki, Z., Kahani, M., & Zarrinkalam, F. (2022). Interpreting sarcasm on social media using attention-based neural networks. Knowledge-Based Systems, 258, 109977.

[13] Frenda, S., Cignarella, A. T., Basile, V., Bosco, C., Patti, V., & Rosso, P. (2022). The unbearable hurtfulness of sarcasm. Expert Systems with Applications, 193, 116398.

[14] Cruz, R. M., de Sousa, W. V., & Cavalcanti, G. D. (2022). Selecting and combining complementary feature representations and classifiers for hate speech detection. Online Social Networks and Media, 28, 100194.

[15] Sharma, A., Kabra, A., & Jain, M. (2022). Ceasing hate with moh: Hate speech detection in hindi–english code-switched language. Information Processing & Management, 59(1), 102760.

[16] Low, J. F., Fung, B. C., Iqbal, F., & Huang, S. C. (2022). Distinguishing between fake news and satire with transformers. Expert Systems with Applications, 187, 115824.

[17] Kumar, P., & Raman, B. (2022). A BERT based dual-channel explainable text emotion recognition system. Neural Networks, 150, 392-407.

[18] Chen, W., Lin, F., Zhang, X., Li, G., & Liu, B. (2022). Jointly learning sentimental clues and context incongruity for sarcasm detection. IEEE Access, 10, 48292-48300.

[19] Jeyakarthic, M., & Senthilkumar, J. (2022, October). Optimal Bidirectional Long Short Term Memory based Sentiment Analysis with Sarcasm Detection and Classification on Twitter Data. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon) (pp. 1-6). IEEE.

[20] 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.

[21] Barhoom, A., Abu-Nasser, B. S., & Abu-Naser, S. S. (2022). Sarcasm Detection in Headline News using Machine and Deep Learning Algorithms.

[22] Nayak, D. K., & Bolla, B. K. (2022). Efficient deep learning methods for sarcasm detection of news headlines. In Machine Learning and Autonomous Systems: Proceedings of ICMLAS 2021 (pp. 371-382). Singapore: Springer Nature Singapore.

[23] Bhardwaj, S., & Prusty, M. R. (2022). BERT pre-processed deep learning model for sarcasm detection. National Academy Science Letters, 45(2), 203-208.

[24] Jayaraman, A. K., Trueman, T. E., Ananthakrishnan, G., Mitra, S., Liu, Q., & Cambria, E. (2022, December). Sarcasm Detection in News Headlines using Supervised Learning. In 2022 International Conference on Artificial Intelligence and Data Engineering (AIDE) (pp. 288-294). IEEE.

[25] Khan, S., Fazil, M., Sejwal, V. K., Alshara, M. A., Alotaibi, R. M., Kamal, A., & Baig, A. R. (2022). BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection. Journal of King Saud University-Computer and Information Sciences, 34(7), 4335-4344.

[26] Misra, R., & Arora, P. (2023). Sarcasm detection using news headlines dataset. AI Open, 4, 13-18.

[27] 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.

[28] Zhao, G., Luo, Y., Chen, Q., & Qian, X. (2023). Aspect-based sentiment analysis via multitask learning for online reviews. Knowledge-Based Systems, 110326.

[29] Zhang, Y., Wang, J., Liu, Y., Rong, L., Zheng, Q., Song, D., ... & Qin, J. (2023). A Multitask learning model for multimodal sarcasm, sentiment and emotion recognition in conversations. Information Fusion.

[30] Thakur, S. Y., Walse, K. H., & Thakare, V. M. (2023, February). A Systematic Review on Explicit and Implicit Aspect Based Sentiment Analysis. In Proceedings of International Conference on Data Science and Applications: ICDSA 2022, Volume 1 (pp. 329-356). Singapore: Springer Nature Singapore.

[31] Chen, S., Wang, P., Fang, W., Deng, X., & Zhang, F. (2019). Learning to predict charges for judgment with legal graph. In Artificial Neural Networks and Machine Learning–ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part IV 28 (pp. 240-252). Springer International Publishing.

[32] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors forward representation," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1532–1543.

[33] Barhoom, A., Abu-Nasser, B. S., & Abu-Naser, S. S. (2022). Sarcasm Detection in Headline News using Machine and Deep Learning Algorithms.

[34] 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.

[35] Misra, R., & Arora, P. (2023). Sarcasm detection using news headlines dataset. AI Open, 4, 13-18.

[36] Jayaraman, A. K., Trueman, T. E., Ananthakrishnan, G., Mitra, S., Liu, Q., & Cambria, E. (2022, December). Sarcasm Detection in News Headlines using Supervised Learning. In *2022 International Conference on Artificial Intelligence and Data Engineering (AIDE)* (pp. 288-294). IEEE.

[37] Misra, R., & Arora, P. (2019). Sarcasm detection using hybrid neural network. *arXiv preprint arXiv:1908.07414*.