Leveraging GloVe Embeddings and LSTM Networks for Sarcasm Detection in News Headlines

1st Md. Tawsifur Rahman
CSE dept. of Brac University
Brac University
Dhaka, Bangladesh
md.tawsifur.rahman@g.bracu.ac.bd

2nd Md. Siam Sadman Azad
CSE dept. of Brac University
Brac University
Dhaka, Bangladesh
siam.sadman.azad@g.bracu.ac.bd

3rd Sayma Akter Lubna
CSE dept of Brac University(of Aff.)
Brac University
City, Country
sayma.akter.lubna@g.bracu.ac.bd

4th Md Mahbub Alam Prithibi
CSE dept of Brac University(of Aff.)
Brac University
Dhaka, Bangladesh
mahbub.alam.prithibi@g.bracu.ac.bd

5th Farah Binta Haque
CSE dept of Brac University
Brac University
Dhaka, Bangladesh
farah.binta.haque@g.bracu.ac.bd

6th Md Sabbir Hossain
CSE dept of Brac University
Brac University
Dhaka, Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

7thAnnajiat Alim Rasel
CSE dept. of Brac University)
Brac University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—Sarcasm detection in textual data remains a challenging yet pivotal task in natural language processing. This paper presents an approach utilizing GloVe word embeddings and LSTM (Long Short-Term Memory) networks to detect sarcasm in news headlines. Leveraging a dataset curated from news sources, we explore the effectiveness of pre-trained GloVe embeddings to represent the semantic and contextual information within headlines. The LSTM architecture, known for its ability to capture sequential dependencies, is employed to discern the intricate linguistic nuances associated with sarcastic expressions. The study involves preprocessing the headline dataset, integrating GloVe embeddings into an LSTM-based model, and evaluating its performance in distinguishing sarcastic from non-sarcastic headlines. Experimental results showcase the efficacy of the proposed methodology, demonstrating promising accuracy and providing insights into the robustness of employing deep learning techniques for sarcasm detection in news headlines. The findings underscore the potential of leveraging sophisticated neural architectures and pre-trained embeddings in deciphering the subtle linguistic cues embedded within textual data, particularly in the realm of news sarcasm detection.**

***Index Terms*—nlp, sarcasm, sarcasm detection**

I. Introduction

Sarcasm, that sly, playful friend of irony, has long eluded the grasp of machines. But with the rise of Natural Language Processing (NLP), a new dawn is breaking. We're now equipped with tools to delve into the depths of human communication and finally understand the subtle nuances of sarcasm.

Imagine a world where machines can decipher the "not really" behind a "great weather, huh?". A world where robots don't take your "loving" punch in the arm literally. This is the promise of sarcasm detection, and NLP is the key.

This paper delves into the realm of sarcasm detection specifically within news headlines, a domain replete with linguistic intricacies and diverse contextual cues. Leveraging advancements in deep learning methodologies, particularly GloVe (Global Vectors for Word Representation) embeddings and LSTM (Long Short-Term Memory) neural networks, we aim to unravel the subtleties of sarcastic language embedded within news headlines.

The use of pre-trained word embeddings, such as GloVe, enables the conversion of textual data into continuous vector representations that encapsulate semantic and contextual information. Complementing this, LSTM networks, renowned for their capability to capture sequential dependencies, offer a promising avenue to discern the implicit sarcasm embedded within headlines by comprehending the underlying linguistic structures.

This study seeks to address the scarcity of research dedicated to sarcasm detection within the news domain while exploring the efficacy of state-of-the-art deep learning techniques. Through an empirical investigation involving the curation and preprocessing of a dataset comprising news headlines, the integration of GloVe embeddings, and the design of an LSTM-based sarcasm detection model, we aim to elucidate the effectiveness of these methodologies in accurately identifying and distinguishing sarcastic from non-sarcastic headlines.

The ensuing sections delve into the methodologies employed, presenting the experimental setup, results, and discussions that underpin the effectiveness and implications of

Identify applicable funding agency here. If none, delete this.

leveraging GloVe embeddings and LSTM networks for the challenging task of sarcasm detection within news headlines.

II. Literature Review

The authors of Abaskohi et al. [2022] [1] investigated a variety of models, including those that were LSTM-based, BERT-based, SVM-based, and attention-based, in addition to various data augmentation techniques. By augmenting, they made an effort to increase the training set.

With an F1 score of 0.414, the authors discovered that optimizing RoBERTa for sentiment classification without data augmentation and additional fine-tuning solely on the iSarcasm dataset produced the best results. The RoBERTa model, which was improved for sarcasm identification by Hercog et al. [2022]b2, outperformed this, though, with a higher F1 score of 0.526 on iSarcasm thanks to its use of a training set that included both iSarcasm and a subset of SARC.

The iSarcasm dataset (Oprea and Magdy [2019]) is unique as it consists of tweets labeled by the authors themselves as sarcastic or non-sarcastic, focusing on intended sarcasm. This labeling approach differs from datasets that label perceived sarcasm, and models trained on such datasets performed poorly on iSarcasm. There are 777 sarcastic tweets and 3,707 non-sarcastic tweets in the sample.

The Self-Annotated Reddit Corpus (Khodak et al. [2017]) is a dataset of Reddit comments labeled for perceived sarcasm, using the ”/s” token convention. However, SARC is noisy due to false negatives (sarcastic comments without ”/s”) and false positives (non-sarcastic comments with ”/s”).

In order to reduce test error more effectively than scaling laws, Sorscher et al. [2022] investigated different approaches to training sample selection. They demonstrated that the amount of beginning data determines the best pruning procedures and that exponential scaling is conceivable with regard to the size of the pruned dataset by choosing an increasing Pareto optimal pruning fraction based on the initial dataset size. The findings hold true for both fine-tuning and scratch training, which is consistent with our configuration.

Cross Entropy Loss was employed by Abaskohi et al. [2022] to address class imbalance in the iSarcasm dataset, we consider Weighted Cross Entropy Loss. Squared Hinge Loss, known for good performance in classification tasks, is also explored. Janocha and Czarnecki [2017] compare various losses and recommend using squared hinge loss when operating in a data-scarce regime, as it converges faster and exhibits more robustness to noise in training set labeling and input space. This robustness is particularly beneficial when augmenting with SARC, a noisier dataset than iSarcasm labeled using distant supervision.

III. Methodology

A. Dataset Preparation:

In this research, we outline a comprehensive methodology for constructing a distinctively labeled dataset by exclusively utilizing news headline datasets. The process begins with the meticulous collection of data from relevant subreddits and reputable news sources, followed by thorough data cleaning to ensure the integrity of the information. Notably, the integration step involves merging the news headlines and Reddit comments datasets into a single data frame, ensuring compatibility between the columns or features of both datasets. Techniques such as fuzzy matching are employed to create a cohesive hybrid dataset encompassing comment text, headlines, timestamps, and sources. Subsequent labeling involves sentiment analysis and topic categorization, assigning labels like positive, negative, neutral, or topical categories. To address potential biases, class distribution is carefully balanced. The dataset is then split into training, validation, and test sets, and features are engineered through text vectorization. Model training, evaluation on a dedicated test set, and iterative improvements complete the methodology, providing a robust foundation for subsequent research. The entire process, including sources, cleaning steps, labeling methods, and model performance metrics, is meticulously documented.

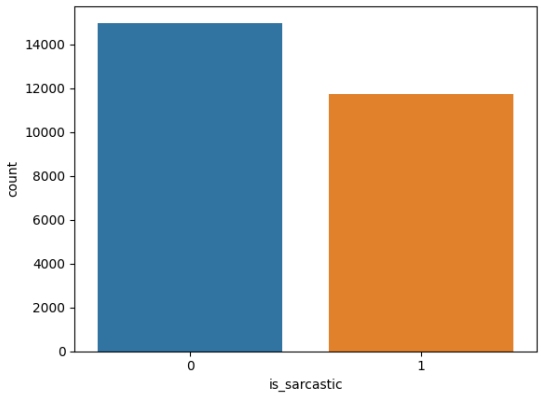

Fig. 1. dataset

B. Data Preprocessing:

In the research, we employed a succinct yet effective data preprocessing method for our hybrid labeled dataset. After loading the dataset, we separated textual data and labels, and subsequently removed brackets, punctuation, numbers before splitting the dataset into training and testing sets. Removing any stopwords seemed to decrease the overall accuracy of the model, therefore we refrained from removing any such english stopwords. To convert the textual data into a format suitable for machine learning, we utilized the TfidfVectorizer.

C. Feature Engineering:

To convert the textual data into numerical representations, we employed the TF-IDF (Term Frequency-Inverse Document Frequency) technique. This involved using the TfidfVectorizer, a method that transforms the raw text into a numerical format by considering both the frequency of each term within a document and its importance across the entire dataset. We configured the TfidfVectorizer to consider a maximum of 5000 features and to exclude common English stop words, optimiz-

ing the representation of the textual content for subsequent machine learning tasks.

D. Model Training:

1) Logistic Regression: Logistic Regression is a fundamental and widely used statistical technique for binary classification problems. Despite its name, it's used for classification rather than regression tasks.

2) Multinomial Naive Bayes: For text-based classification applications, the multinomial naive bayes algorithm is a straightforward but powerful method. In actual use, it frequently works well despite its feature independence assumption, particularly when working with high-dimensional and sparse data, such word frequencies in text. However, multinomial naive bayes has its weaknesses. The assumption of feature independence might not hold in real-world scenarios, potentially affecting the accuracy. If a feature doesn't occur in a class in the training data, the probability for that class will become zero, impacting predictions (Laplace smoothing is often used to mitigate this issue).

3) Bernoulli Naive Bayes: The Bernoulli Naive Bayes (BernoulliNB) model is a probabilistic classifier that belongs to the family of Naive Bayes classifiers. It's specifically designed for binary feature vectors, making it well-suited for datasets where features represent binary variables or occurrences (such as word presence/absence).

E. BiLSTM GloVe SarcasmDetector model Architecture

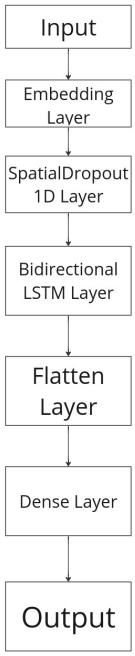

Fig. 2. Model architecture

a) Embedding Layer: The model starts with an Embedding layer. This layer converts input sequences into dense vectors of fixed size. It uses pre-trained word embeddings loaded into the embedding matrix. The parameter num words specifies the size of the vocabulary. trainable=False means the embeddings (weights) loaded from GloVe will remain fixed and not be updated during training.

b) Spatial Dropout: A particular kind of dropout called SpatialDropout1D works by randomly changing a portion of the input units to zero in order to avert overfitting in neural networks. The dropout rate, or 40% of the input units randomly dropped during training, is represented by the number 0.4.

Long Short-Term Memory (LSTM) bidirectional neural networks (RNNs) are a kind of recurrent neural network (RNN) used to identify long-term dependencies in data that is sequential. When an LSTM is bidirectional, it means that it analyzes the input sequence in both directions, gathering data from the sequence's past and future states.

c) Flatten Layer: This layer flattens the output from the Bidirectional LSTM into a 1D array. It prepares the output for the subsequent Dense layer.

d) Dense Layer: A Dense layer with just a single unit and an activation function that is sigmoid marks the model's conclusion. For a binary classification task (like sentiment analysis or binary categorization), it generates a binary categorization output of either 0 or 1. The output is compressed between 0 and 1 by the sigmoid activation function, which then interprets it as a probability.

e) Summary: The model architecture consists of an Embedding layer with pre-trained GloVe embeddings, followed by a Spatial Dropout layer to prevent overfitting. Then, a Bidirectional LSTM layer captures contextual information bidirectionally, and a Flatten layer reshapes the output for the final Dense layer, which produces a binary classification output using a sigmoid activation function. This type of architecture is commonly used in NLP tasks for sequence classification, especially when dealing with text data.

F. Model Comparison:

The logistic regression, BernoulliNB, and MultinomialNB models perform reasonably well, with accuracies ranging from 83% to 85%. However, the BiLSTM model utilizing GloVe embeddings stands out, achieving the highest accuracy of 87.68%. Its capacity to capture complex linguistic patterns and contextual information enables more accurate detection of sarcasm in news headlines. While the traditional machine learning models show decent performance, the BiLSTM model utilizing GloVe embeddings demonstrates superior accuracy, highlighting the efficacy of deep learning techniques in capturing nuanced linguistic features for sarcasm detection within news headlines.

IV. Analyze Results

A. MultinomialNB

1) Precision: Precision shows 0.81 for category 0 and 0.88 for category 1.

2) Recall: Represents the model's ability to correctly identify instances of each category. Category 0 has a higher recall (0.93) compared to category 1 (0.72). This implies that the model is better at capturing most of the category 0 instances but misses some of the category 1 instances.

category	precision	recall	f1-score	support
0	0.81	0.93	0.87	3017
1	0.88	0.72	0.79	2325
accuracy	0.84	5342		
macro avg	0.85	0.82	0.83	5342
weighted avg	0.84	0.84	0.83	5342

TABLE I
CLASSIFICATION REPORT

3) F1-Score: Harmonic mean of precision and recall, providing a balanced measure between the two. Category 0 has a higher F1-score (0.87) compared to category 1 (0.79). It highlights the overall effectiveness of the model in classifying both categories.

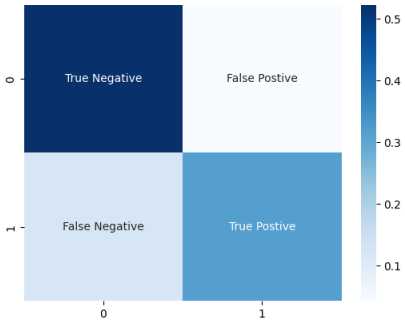

Fig. 3. Confusion matrix

4) Support: Shows the instances belonging to each category.

5) Accuracy: The overall accuracy of the model is 84%, which is the proportion of correctly classified instances across both categories. While accuracy is a valuable metric, the discrepancy in recall between categories indicates an imbalance in the model's performance between the two classes.

B. BernoulliNB

category	precision	recall	f1-score	support
0	0.84	0.91	0.87	3017
1	0.87	0.77	0.82	2325
accuracy	0.85	5342		
macro avg	0.85	0.84	0.84	5342
weighted avg	0.85	0.85	0.85	5342

TABLE II
CLASSIFICATION REPORT

1) Precision: For category 0, the precision is 0.84, and for category 1, it's 0.87. This suggests that the model's accuracy in predicting headlines belonging to category 0 or 1 is approximately 84% and 87%, respectively.

2) Recall: Category 0 exhibits a recall of 0.91, while category 1 has a recall of 0.77.

3) F1-Score: Category 0 has an F1-Score of 0.87, and category 1 has an F1-Score of 0.82.

4) Accuracy: The overall accuracy of the model is 85%.

C. Logistic Regression

1) Precision: For category 0, precision stands at 0.86, and for category 1, it's 0.82.

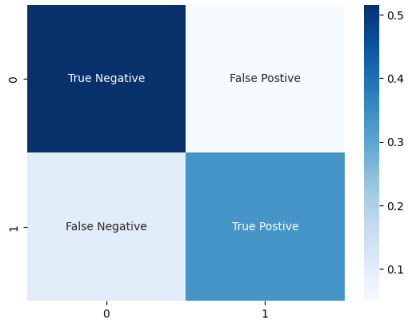

Fig. 4. Confusion matrix

category	precision	recall	f1-score	support
0	0.86	0.87	0.86	3017
1	0.82	0.81	0.82	2325
accuracy	0.84	5342		
macro avg	0.84	0.84	0.84	5342
weighted avg	0.84	0.84	0.84	5342

TABLE III
CLASSIFICATION REPORT

2) Recall: Category 0 exhibits a recall of 0.87, while category 1 has a recall of 0.81.

3) F1-Score: Category 0 has an F1-Score of 0.86, and category 1 has an F1-Score of 0.82.

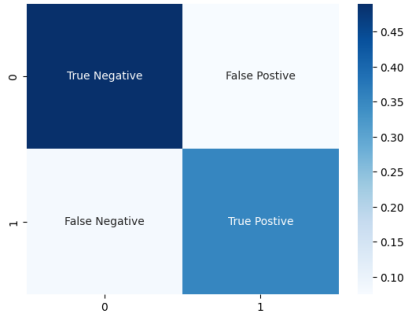

Fig. 5. Confusion matrix

4) Accuracy: The overall accuracy of the model is 84%.

D. BiLSTM GloVe SarcasmDetector

An accuracy of 87.68% and a test loss of 30.57%

E. Model Result Comaparison

model	f1-score
Logistic Regression	0.86
Bernoulli NB	0.87
Multinomial NB	0.87
BiLSTM GloVe SarcasmDetector	0.88

TABLE IV
MODEL COMPARISON

Logistic Regression, BernoulliNB, and MultinomialNB from the table IV demonstrate robust performance with F1-Scores ranging from 0.86 to 0.87. However, the BiLSTM

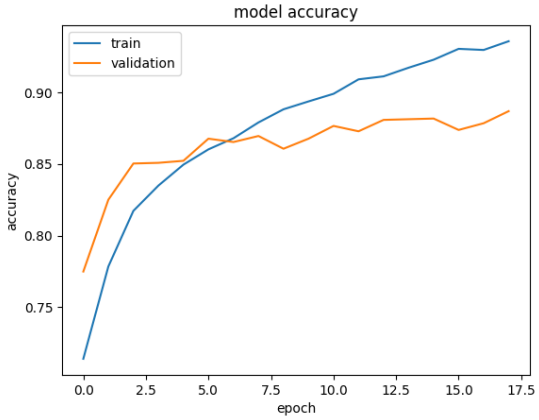

Fig. 6. accuracy graph

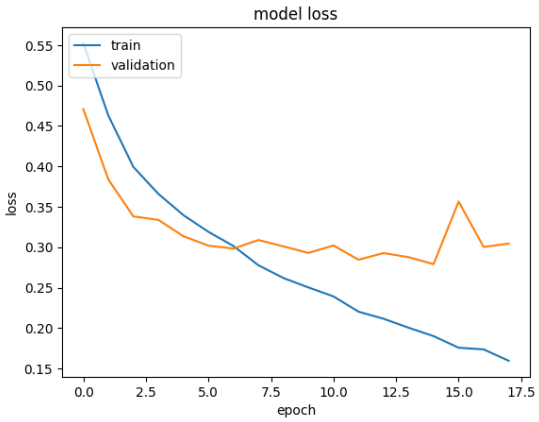

Fig. 7. loss graph

model utilizing GloVe embeddings stands out with the highest F1-Score of 0.88, showcasing its effectiveness in capturing complex linguistic patterns for sarcasm detection in news headlines. While traditional machine learning models perform well, the BiLSTM model leveraging deep learning techniques demonstrates superior accuracy, precision, and recall, making it a compelling choice for sarcasm detection tasks within news headlines.

REFERENCES

[1] Amirhossein Abaskohi, Arash Rasouli, Tanin Zeraati, and Behnam Bahrak. Utnlp at semeval-2022 task 6: A comparative analysis of sarcasm detection using generative-based and mutation-based data augmentation. arXiv preprint arXiv:2204.08198, 2022.

[2] Maciej Hercog, Piotr Jaroński, Jan Kolanowski, Paweł Mieczyński, Dawid Wiśniewski, and Jedrzej Potoniec. Sarcastic roberta: A roberta-based deep neural network detecting sarcasm on twit- ter. In International Conference on Big Data Analytics and Knowledge Discovery, pages 46–52. Springer, 2022.

[3] Silviu Oprea and Walid Magdy. isarcasm: A dataset of intended sarcasm. arXiv preprint arXiv:1911.03123, 2019.

[4] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A large self-annotated corpus for sarcasm. arXiv preprint arXiv:1704.05579, 2017.

[5] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. CS224N project report, Stanford, 1(12):2009, 2009.

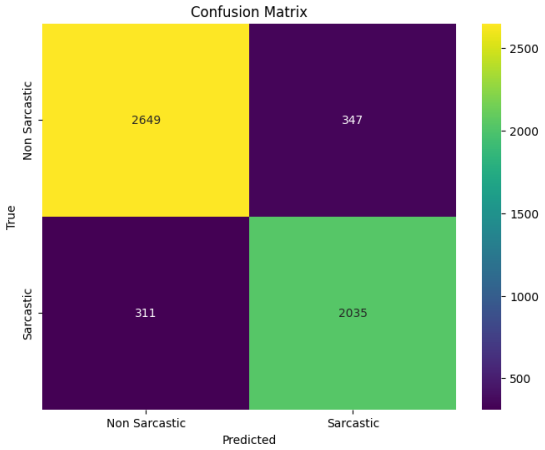

Fig. 8. confusion matrix

[6] Rishabh Misra and Prahal Arora. Sarcasm detection using hybrid neural network. arXiv preprint arXiv:1908.07414, 2019.

[7] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. arXiv preprint arXiv:2206.14486, 2022.

[8] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. arXiv preprint arXiv:1702.05659, 2017.