News Headlines for Sarcasm Detection

Daniel Ferreira Mec. Number: 102442
DETI
Universidade de Aveiro
danielmartinsferreira@ua.pt

Guilherme Antunes Mec. Number: 103600
DETI
Universidade de Aveiro
guilherme.antunes@ua.pt

***Abstract*—This paper presents a comprehensive study on sarcasm detection in news headlines, an emerging field in Natural Language Processing (NLP) and Sentiment Analysis. We utilize the News Headlines Dataset for Sarcasm Detection from Kaggle, which contains numerous examples of sarcastic and non-sarcastic news headlines, allowing for the training and testing of machine learning algorithms. We propose and apply various machine learning models to this dataset, conducting a thorough analysis of their performance. Through this work, we aim to contribute to the ongoing efforts in sarcasm detection, a critical aspect of understanding sentiment and context in human language. The paper further discusses the data preprocessing steps taken, the selected features, and presents the results and insights obtained from the applied models. Finally, it compares the performance of our model with existing solutions and discusses future research directions in sarcasm detection.**

***Index Terms*—sarcasm, Bayes Networks, RNN, CNN, BERT**

I. Introduction

This paper presents an extensive exploration of the methods, approaches, and algorithms used in our project for the Topics in Automated Learning (TAA) course at Universidade de Aveiro, taught by Professor Petia Georgieva. The course required the application of machine learning techniques, which we either learned throughout the semester or independently, to address one of several suggested problems.

Our group decided to tackle the challenge of sarcasm detection in news headlines, a problem of increasing significance in the field of natural language processing. This issue required us to develop a machine learning model capable of accurately identifying sarcasm in text data, specifically in news headlines. The complexity of the task and the potential contribution to the understanding of human communication and its algorithmic modeling, particularly the linguistic nuances of sarcasm, drew our attention to this problem.

For this project, we utilized the News Headlines Dataset for Sarcasm Detection [2] [1]. This dataset, collected from TheOnion and HuffPost, contains approximately 28,000 headlines, about 13,000 of which are sarcastic. TheOnion focuses on creating sarcastic versions of current events, while HuffPost publishes genuine news.

This study aligns with previous research in the fields of natural language processing, text analysis, and machine learning. It underscores the significance of research in language understanding, particularly in the detection and understanding of sarcasm, and the potential impact it may have on various applications, such as sentiment analysis, social media monitoring against fake news, and the development of more nuanced dialogue systems.

II. State of the Art

The application of machine learning and artificial intelligence for detecting sarcasm and clickbait has garnered significant attention in academic research, with various studies developing and fine-tuning sophisticated models for these specific tasks.

A. Sarcasm Detection

A notable research in this field is titled ”Sarcasm Discernment on Social Media Platform” by Namasani Sagarika *et al.* [4]. The study made use of the News Headlines Dataset, which includes around 28K headlines, of which approximately 13K are sarcastic [1]. The advantages of using professionally written news headlines, which have fewer spelling mistakes and a more formal language usage, were emphasized in the study. These characteristics reduce the dataset's sparsity and enhance the potential for finding pre-trained embeddings. Unfortunately, due to the limited depth of the full paper, the detailed implementation is not sufficient for comparison purposes.

A model developed by Siddharth Agarwal, titled ”Simple BERT Sarcasm Detection," has been identified as a superior benchmark [5]. This model, hosted on Kaggle, employs the BERT (Bidirectional Encoder Representations from Transformers) model, which has shown impressive results in various natural language processing tasks. The model's simplicity, combined with the power of BERT, makes it an ideal benchmark for sarcasm detection. The required computational power to simulate these results made its use as a benchmark cumbersome.

B. Clickbait Detection

In the domain of clickbait detection, a study by Diyah Utami Kusumaning Putri and Dinar Nugroho Pratomo [3] has made significant strides. The researchers used a fine-tuned version of the Bidirectional Encoder Representations from Transformers (BERT) model, specifically a model called IndoBERT tailored for the Indonesian language. The research used the CLICK-ID dataset, which contains both clickbait and non-clickbait Indonesian news headlines. The fine-tuned IndoBERT classifiers outperformed all word-vectors-based machine learning classifiers in classifying these headlines. The highest accuracy achieved was 0.8247, which was 6% better than the accuracy of the SVM classifier with the bag-of-words model.

C. Benchmark Selection

For clickbait detection, the model ”LSTM, CNN with Tensorflow + LDA (topic modelling)” by Jinghui Wong stands out [6]. This model, also hosted on Kaggle, uses a combination of Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), and Latent Dirichlet Allocation (LDA) for topic modelling. The integration of these diverse techniques allows the model to capture both the sequential nature of the text and the underlying topics, making it a robust tool for clickbait detection.

This model, due to its innovative use of advanced machine learning techniques and its demonstrated effectiveness, will be used a a benchmark for this study. The goal is to develop models that can match or surpass its performance in detecting sarcasm and clickbait in text data.

III. Dataset and Preprocessing

A. Dataset Description

The dataset under consideration is the ”News Headlines Dataset For Sarcasm Detection” collected by Rishabh Misra and available on Kaggle. This dataset is specifically designed for the task of sarcasm and fake news detection, and it represents a high-quality resource for such tasks.

The dataset is collected from two news websites: TheOnion and HuffPost. TheOnion is known for producing sarcastic versions of current events, and all the headlines from ”News in Brief” and ”News in Photos” categories, which are sarcastic, are included in the dataset. On the other hand, HuffPost provides real and non-sarcastic news headlines.

The dataset has several advantages over existing Twitter datasets used for sarcasm detection. News headlines are written by professionals in a formal manner, eliminating spelling mistakes and informal usage. This reduces sparsity and increases the chance of finding pre-trained embeddings. Furthermore, since TheOnion's sole purpose is to publish sarcastic news, the dataset provides high-quality labels with much less noise compared to Twitter datasets. Unlike tweets, which are often replies to other tweets, the news headlines in this dataset are self-contained, which aids in identifying the real sarcastic elements.

The dataset consists of three attributes for each record:

- **is_sarcastic**: This is a binary attribute where 1 indicates that the record is sarcastic, and 0 indicates otherwise.
- **headline**: This attribute contains the headline of the news article.
- **article_link**: This attribute provides the link to the original news article, which can be useful for collecting supplementary data.

B. Dataset Analysis

Containing around 28k entries, this dataset is fairly distributed, both absolutely and relatively, given the fact that there are only two classes (sarcastic and non-sarcastic) as seen in Figures 1 and 2.

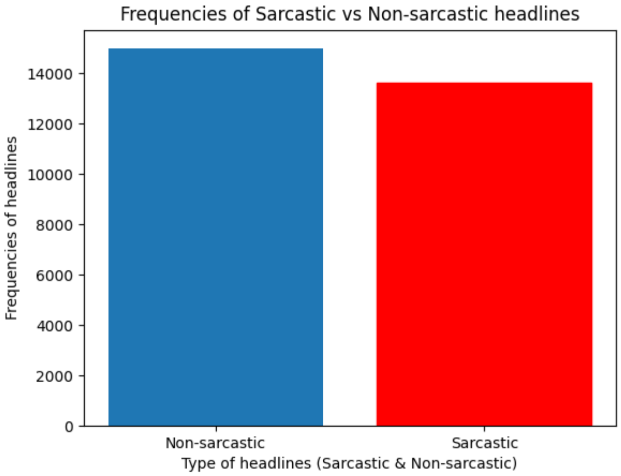


Fig. 1. Absolute Distribution of Classes in the Dataset

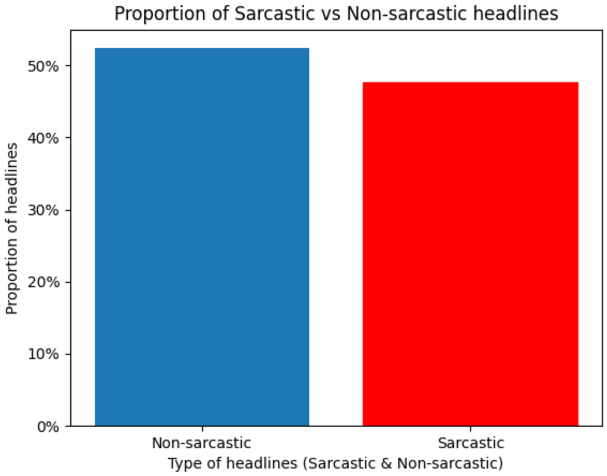


Fig. 2. Relative Distribution of Classes in the Dataset

An analysis of the wordcloud of this dataset reveals that it is heavily skewed towards political news, with references to political figures being very present (”Donald Trump”, ”Obama”, ”senate”, ”white house” etc.), as well as terms generally associated with news articles in general (”new”, ”local” etc.) as seen in Figure 3.

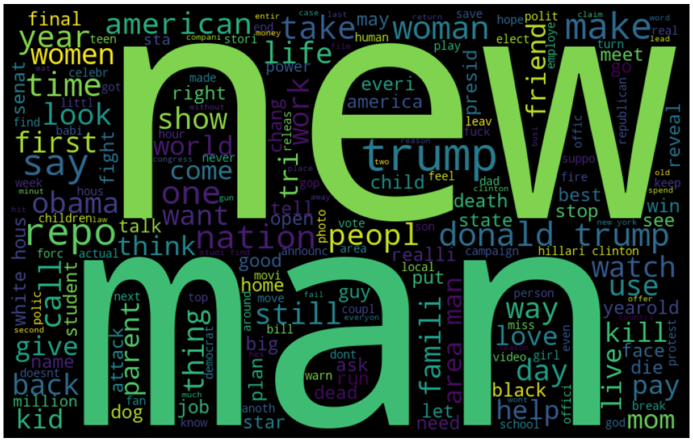


Fig. 3. Dataset Word Cloud

Another potentially relevant aspect for analysis is the distribution of the length of each headline, since longer headlines could provide more context and, therefore, be easier to categorize. In Figure 4, as expected, there is a normal distribution of the headline length, with some anomalies that extend the range of considered results. Distributions are similar for both the sarcastic and non-sarcastic subsets, noting that the sarcastic set (Figure 5) contains the aforementioned anomalies that result in the horizontal distension of the graph, while the non-sarcastic subset lacks these higher sparse values and is more concentrated in general (Figure 6).

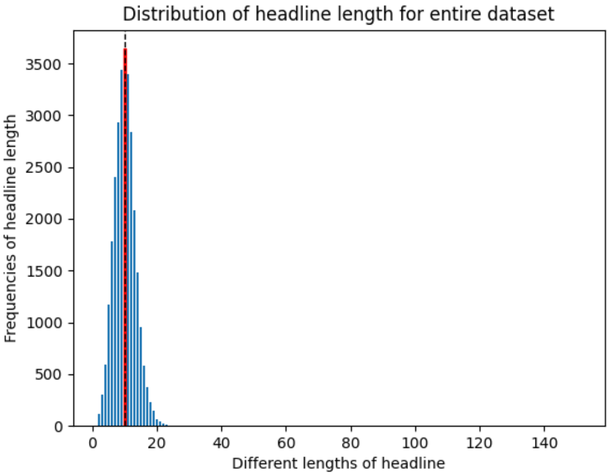


Fig. 4. Headline Length Distribution

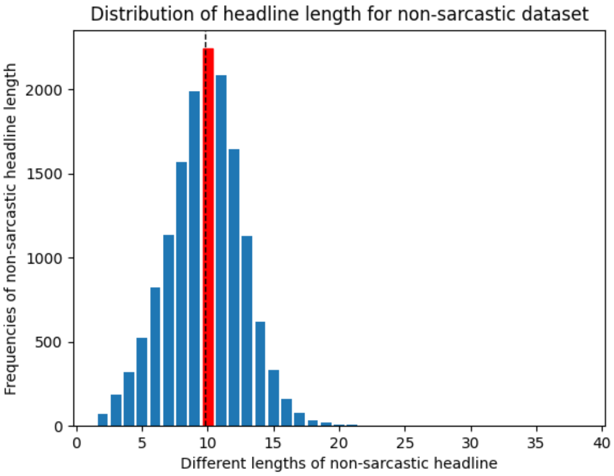


Fig. 5. Headline Length Distribution (non-sarcastic)

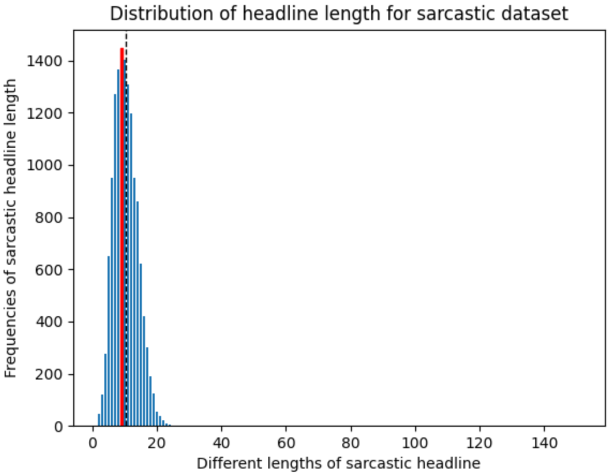


Fig. 6. Headline Length Distribution (sarcastic)

C. Preprocessing

The data cleansing process was straightforward yet crucial for preparing the data for the analysis. All the text in the headlines was converted to lowercase to ensure consistency. Subsequently, all characters in the text data that were neither alphabetic nor whitespace were eliminated. This was done to simplify the text and eliminate any potential noise in the data. Additionally, any accents on letters were also removed to further simplify the text and standardize the words.

The cleaning process also involved the removal of certain strings, namely ’rt’ and ’http’, which are often found in social media posts and do not carry meaningful information for the task at hand. Finally, words with less than three characters were discarded. These short words often include prepositions and articles which might not provide significant value for the text classification task.

These preprocessing steps ensured the cleanliness and standardization of the text data, thereby setting the stage for further analysis and modeling.

Another of the essential steps in preprocessing textual data is the removal of *stop words*. Stop words in natural language processing (NLP) are extremely common words that carry little meaningful content and that were not already discarded in the initial data cleaning. They include articles, pronouns, conjunctions, and prepositions such as ”and”, ”or”, ”the”, ”is”, ”at”, ”which”, and ”on”.

The set of stop words used in this project includes common words as well as contractions, such as ”didn't”, ”haven't”, and ”shouldn't”. An example of the stop words used is shown below:

```
{'your', 'been', 'after', 'once',
'between', 'most', "didn't", "haven't",
'with', 'she', 'off', ...}
```

These words are filtered out during the preprocessing phase because their high frequency can add noise to the analysis if included. They often do not provide meaningful information on their own and can potentially hinder the performance of machine learning algorithms. By removing these stop words, we can significantly reduce the dimensionality of the data and allow the algorithm to focus on the important words that carry more informational weight.

However, it is important to note that the use of stop words is not always advisable. In some NLP tasks, such as language modeling, machine translation, and text summarization, stop words can contain important grammatical information, and removing them may lead to undesirable results. Hence, the decision to use stop words is permissible in this situation because it does not fall into the aforementioned categories.

Beyond this change, applied to the **headline** column, no further changes were applied or required.

D. Labelling and Feature Selection

In terms of labels, the **is_sarcastic** column is the only relevant option and is already clearly identified in each row. In terms of feature selection, word analysis of the **headline** atribute is, again, the only plausible option, since the **article_link** does not carry any relevant information for the task at hand.

IV. Approach

A. Overview

In this project, we make use of the BernoulliNB classifier from the Python library scikit-learn. BernoulliNB is a type of Naive Bayes classifier that is particularly suited for binary or boolean features, i.e., features that can assume only two values, such as True/False.

The Naive Bayes method is based on Bayes' theorem, a principle in probability theory and statistics that calculates the probability of an event based on prior knowledge of conditions that might be related to the event. Naive Bayes classifiers are termed ”naive” because they assume that the features are independent of each other. Although this assumption is not always met in real-world data, Naive Bayes classifiers can still perform remarkably well.

The ”Bernoulli” in BernoulliNB refers to the Bernoulli distribution, a probability distribution of a random variable which takes binary outcomes: 1 (success, true, yes) with probability p, and 0 (failure, false, no) with probability $1 - p$.

In the context of BernoulliNB, each feature is assumed to be a binary variable following a Bernoulli distribution. Each feature is considered independently, meaning that the occurrence or non-occurrence of a feature does not influence the occurrence or non-occurrence of any other feature.

BernoulliNB estimates the parameters of the Bernoulli distribution for each feature and each class during the training process. These parameters are then used to predict the probabilities of each class for a new instance, and the class with the highest probability is output.

This model is particularly useful for text classification tasks, where each instance can be described by the presence or absence of specific features, such as words in a document. In this project, BernoulliNB was employed for the detection of sarcasm and clickbait in text, demonstrating its applicability in real-world NLP tasks.

B. Hyperparameter Selection

In terms of hyperparameters, there was one main decision to be made: the alpha smoothing parameter.

In the context of Naive Bayes classifiers, ”smoothing” refers to a technique used to handle the problem of zero probability. When we're estimating the probability of certain events based on frequency in our data, if a given feature-label combination does not occur in the data, its estimated probability will be zero. This is problematic, because when we then use these probabilities to make predictions on new data, the presence of any feature with a zero probability will result in a zero probability for the entire prediction.

To avoid this, smoothing assigns a small, non-zero base level of probability to all possible events. The alpha parameter in BernoulliNB controls the amount of smoothing: a larger alpha means more smoothing, and a smaller alpha means less smoothing. If alpha = 1, this is known as Laplace smoothing or add-one smoothing, where 1 is added to the count of each feature-label combination. If alpha ¡ 1, it's known as Lidstone smoothing, where a fraction is added instead.

Even though both labels (sarcastic and non-sarcastic) occur in your data, smoothing is still important for handling features (in this case, individual words or phrases) that may not appear with each label. For instance, there might be words that appear in sarcastic headlines in your training data, but never in non-sarcastic headlines. Without smoothing, if the model then sees these words in a non-sarcastic headline in the test data, it would assign a zero probability to that prediction, which is not desirable.

By setting alpha to 1, we're using Laplace smoothing, which is a common choice that generally works well in practice. This ensures that all possible feature-label combinations are assigned some base level of probability, which can help the model generalize better to unseen data reducing bias and overfitting.

C. Training, Testing and Validation

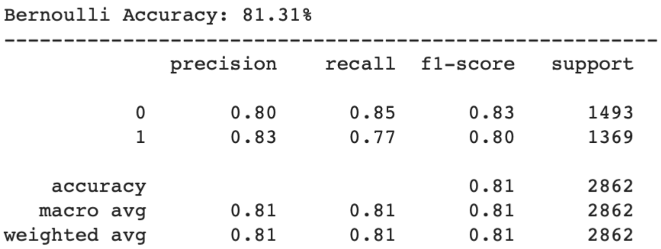
The data was split into a training set and a test set, with 90% of the data used for training and 10% used for testing. This allows the model to be trained on one set of data and then evaluated on a separate set that it has not seen before, which gives a more realistic assessment of its performance.

Holdout was used, so no validation set was defined since the use of Laplace smoothing and train-test split already sufficiently prevent overfitting.

V. Results and Discussion

A. BernoulliNB

Figure 7 details the results obtained with our model which are satisfactory and very impressive given the lack of validation and the fact that this model isn't based on training over various EPOCHs.



```
Bernoulli Accuracy: 81.31%
-----------------------------------------------------
              precision    recall  f1-score   support

           0       0.80      0.85      0.83      1493
           1       0.83      0.77      0.80      1369

    accuracy                           0.81      2862
   macro avg       0.81      0.81      0.81      2862
weighted avg       0.81      0.81      0.81      2862
```

Fig. 7. BernoulliNB Performance Metrics

As seen in the confusion matrix, it is reliable enough to be used in real world applications, but still lacks near-perfect accuracy required for more demanding tasks.

B. Reference Comparison

For reference, we used the models developed by Jing Hui Wong [6]: a BiRNN model with all combinations of LSTM or GRU and L1 or L2 regularzation, from which the best possible combinations were used a 1D CNN model, again, with either Lasso or Ridge regularization and a CNN-RNN hybrid model.

- **BiRNN Model with LSTM/GRU and L1/L2 Regularization:** BiRNN stands for Bidirectional Recurrent Neural Networks. It is a type of Recurrent Neural Network (RNN) that involves duplicating the first recurrent layer in the network so that there are now two layers side-by-side. The input sequence is then provided to the first layer as-is, and a reversed copy of the input sequence is provided to the second. LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) are different types of RNNs which are used in these models. Regularization is a technique used to prevent overfitting in a model by adding a penalty term to the loss function. L1 (Lasso) and L2 (Ridge) are two types of regularization techniques.
- **1D CNN Model with Lasso/Ridge Regularization:** CNN stands for Convolutional Neural Networks. They are mainly used for image processing but can also be used for any type of prediction tasks with grid-like topology. 1D CNN models are generally used for sequence data, like a string, which is the case. Lasso (L1) or Ridge (L2) regularization is applied to the model to prevent overfitting.

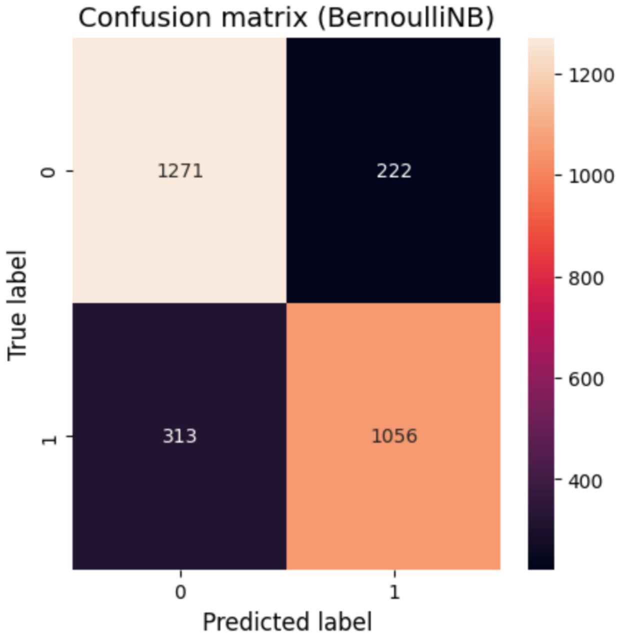


Fig. 8. Confusion Matrix for BernoulliNB

- **CNN-RNN Hybrid Model:** This model combines Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNN layers are used to extract a meaningful sequence of features from the input data. These sequences are then fed into RNN layers which capture the temporal dependencies of the features. This type of model can be particularly useful when you are working with data where both the temporal sequence and the local features in the sequence are important.

Out of all these combinations, the best results were obtained with BiRNN with LSTM and L1 (83,82% accuracy), CNN with L2 (84,31% accuracy) and CNN-RNN Hybrid (84,80% accuracy).

The corresponding confusion matrices, training validation over EPOCHs and training loss results are presented ahead.

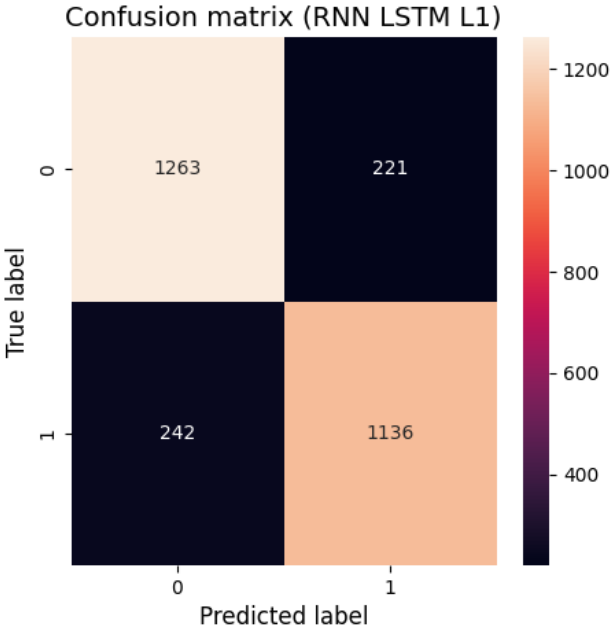


Fig. 9. Confusion Matrix for BiRNN (LTSM and L1)

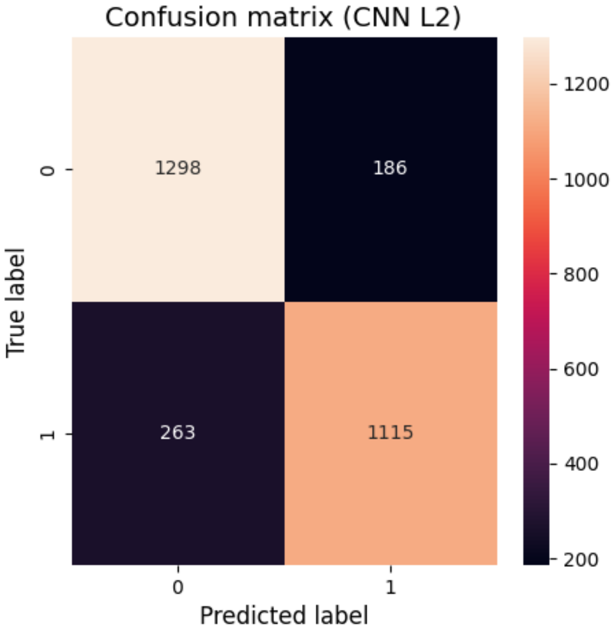


Fig. 10. Confusion Matrix for CNN (L2)

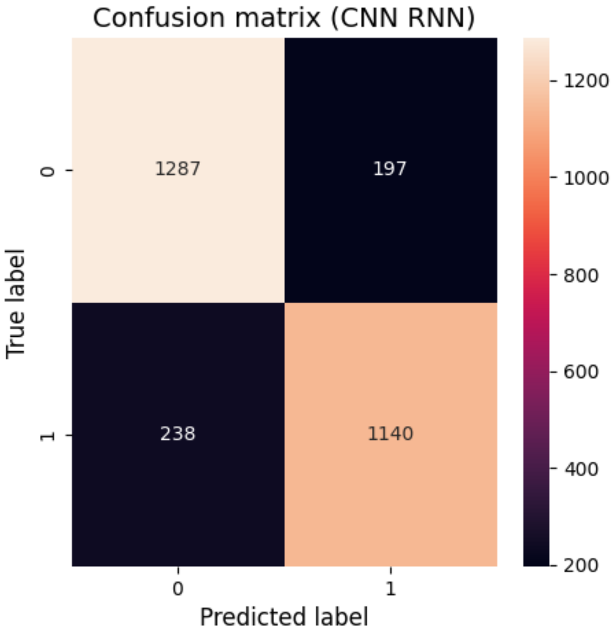


Fig. 11. Confusion Matrix for CNN-RNN Hybrid

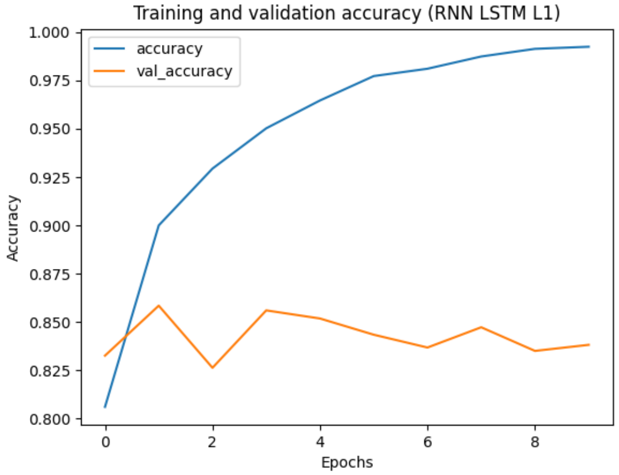


Fig. 12. Training Accuracy over EPOCHs for BiRNN (LTSM and L1)

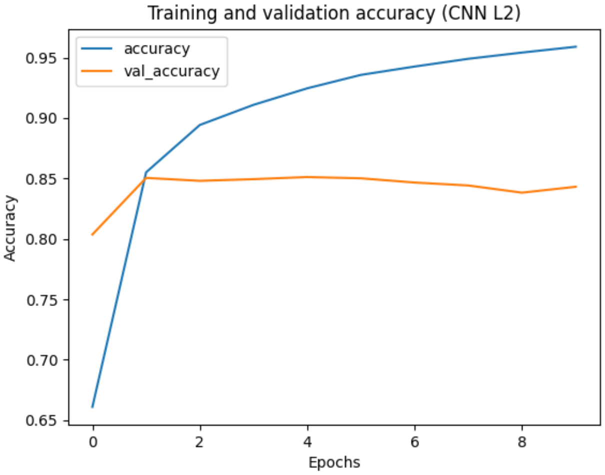


Fig. 13. Training Accuracy for CNN (L2)

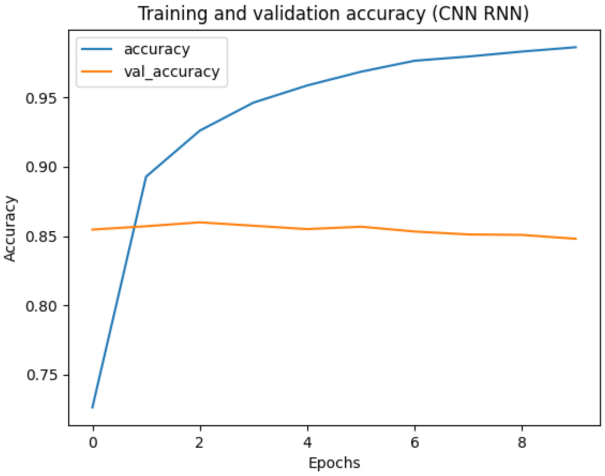


Fig. 14. Training Accuracy for CNN-RNN Hybrid

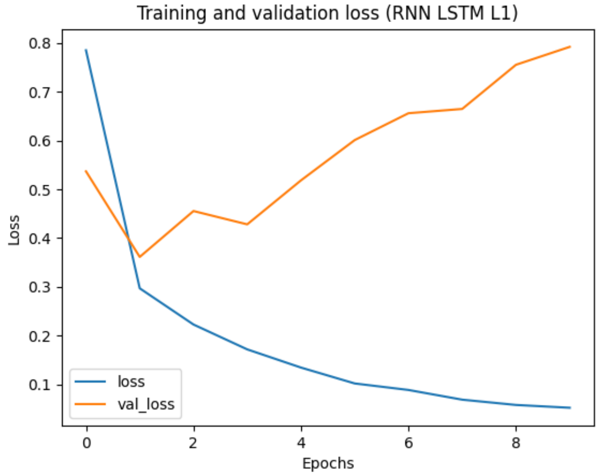


Fig. 15. Training Loss for BiRNN (LTSM and L1)

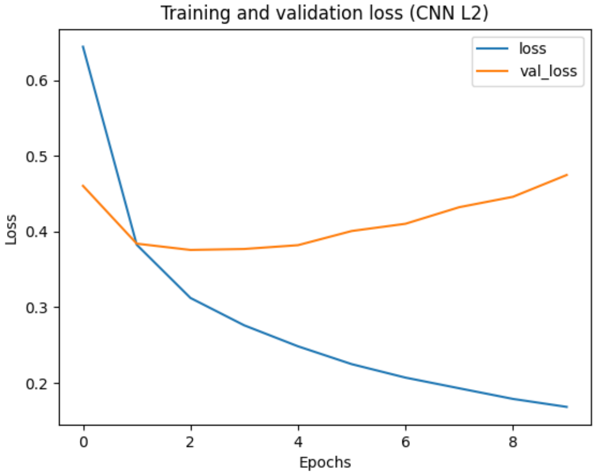


Fig. 16. Training Loss for CNN (L2)

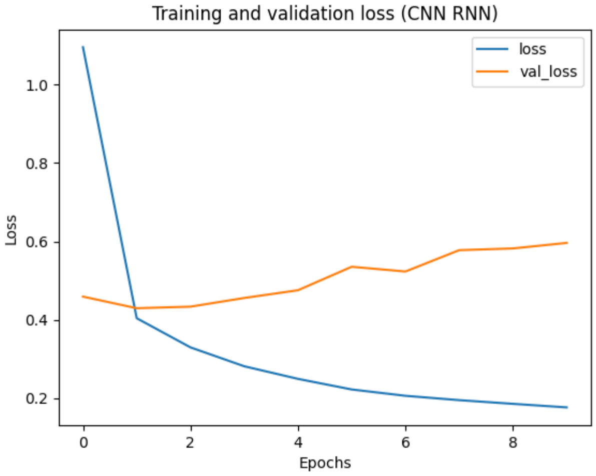


Fig. 17. Training Loss for CNN-RNN Hybrid

VI. Conclusion

While our model may not have matched the performance metrics of the benchmark references, it is important to consider the overall cost-benefit analysis of our solution. Training complex deep learning models often requires multiple EPOCH, which can be computationally expensive and time-consuming. Each EPOCH represents a full pass through the entire dataset, which can be a significant overhead especially for large datasets or complex models.

In contrast, our solution provides a more efficient alternative. The Naive Bayes classifier, while being a simple model, is fast to train even on large datasets. This is because it does not require multiple iterations over the dataset to converge to a solution, unlike deep learning models. Moreover, it has less computational requirements, making it a more feasible solution for devices with limited computational power.

Furthermore, the performance of our model is within the same order of magnitude as the more complex solutions. This implies that while there is a difference in performance, it is not exceedingly large. Therefore, when considering the balance between the resources used (in terms of computational power and time) and the performance obtained, our model presents a more cost-effective solution for sarcasm detection in news headlines.

Thus, the overall effectiveness of a machine learning model should not be solely judged by its performance metrics, but also by assessing the computational efficiency and resource requirements. This holistic evaluation approach ensures that the model is not only accurate but also scalable and practical for real-world applications.

VII. Contributions

Both members contributed equally to this project.

References

[1] Rishabh Misra and Prahal Arora. Sarcasm detection using news headlines dataset. *AI Open*, 4:13–18, 2023.

[2] Rishabh Misra and Jigyasa Grover. *Sculpting Data for ML: The first act of Machine Learning*. 01 2021.

[3] Diyah Utami Kusumaning Putri and Dinar Nugroho Pratomo. Clickbait detection of indonesian news headlines using fine-tune bidirectional encoder representations from transformers (bert). *Inform: Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 7(2):68–77, 2022.

[4] Namasani Sagarika, Bommadi Sreenija Reddy, Vanka Varshitha, Kodavati Geetanjali, N. V. Ganapathi Raju, and Latha Kunaparaju. Sarcasm discernment on social media platform. In *E3S Web of Conferences*, volume 309, page 01037. EDP Sciences, 2021.

[5] Siddharth. Simple bert sarcasm detection, Year. Accessed: 2023-06-20.

[6] Jing Hui Wong. Lstm, cnn with tensorflow + lda (topic modelling), Year. Accessed: 2023-06-20.