**Project Final Report**

**Detecting Fake News on Social media**

**Group Member and Contribution:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team member Name** | **Team member Email IDs** | **R# IDs** | **Contribution** |
| Vamsi Krishna Muppala | [vamuppal@ttu.edu](mailto:vamuppal@ttu.edu) | R11770853 | * Implemented the BERT technology * Done research on SVM and executed the method in the model. * Computed the F1 score and accuracy |
| Edwin Sahil Samuel Kathi | [ekathi@ttu.edu](mailto:ekathi@ttu.edu) | R11800320 |
| Yaswanth Kumar Chaganti | [ychagant@ttu.edu](mailto:ychagant@ttu.edu) | R11801037 | * Worked on the presentation. * Done research on Binary Feature representation and executed the method in the model. * Computed the confusion matrix * Done research on BERT technology |
| Jayakumar Kukkathotti | [jkukkath@ttu.edu](mailto:jkukkath@ttu.edu) | R11801622 |
| Spoorthy Gope Gowda | [sgopewgow@ttu.edu](mailto:sgopewgow@ttu.edu) | R11795421 | * Done research on future work. * Worked on the Report * Research work on the feature selections on Counts * Done the research on BERT technology |
| Manoj Kumar Reddy | [manoredd@ttu.edu](mailto:manoredd@ttu.edu) | R11804056 |
| Rahul Bantu | [Rbantu@ttu.edu](mailto:Rbantu@ttu.edu) | R11798527 | * Worked on the presentation. * Read papers for future work. * Research work on the feature selections on TD-IDF * Helped in Neural network model development. |
| Thivatharan Jeganathan | [tjeganat@ttu.edu](mailto:tjeganat@ttu.edu) | R11845830 |

**Problem Statement:**

There is a lot of false information on social media. To detect it, designing a news monitor system that concentrates to alert the public about fake news is important.

**Goals**:

Identifying fake news and guiding people to think about false information.

Build a model which helps in predicting if the news is fake or real.

The text discusses the challenges of detecting fake news and the limitations of statistical approaches due to the lack of labeled benchmark datasets.

**Dataset**:

The dataset used is the LIAR dataset, which can be used for fact-checking research and fake news detection. We clean the dataset to get the processed data which helps in detecting fake news.

Link:<https://github.com/kiranrawat/Detecting-Fake-News-On-Social-Media/tree/main/data>

**Existing Problem Solution:**

**Cleaning Dataset:**

* We clean the raw dataset by eliminating punctuation's, white spaces, lowercase's, stop words and lemmanize the tokens.
* After cleaning the data, we use the processed data to train the model.

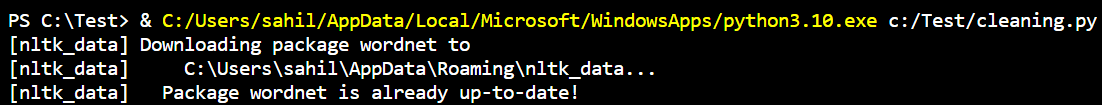
**Training Model:**

* After pre-processing the data, we extract the features using Binary, Count Vectorizer and TFIDF
* For training the model we use methodologies like SVM
* After selecting a methodology, we train the model using Scikit-Learn
* Now, we have calculated F1 score and confusion matrix for every methodology corresponding to every feature selection mentioned above.
* Based on the F1 score and confusion matrix we select the best model and corresponding feature selection to train the model.
* After we checked the best model to compare the accuracy, BERT model has been at its best to give higher accuracy when we compare it to SVM.

**Previous Experiment Results:**

Work done apart from paper: We tried TF-IDF, Count and Binary vectorisation for SVM, XG boost and Logistic Regression and got maximum accuracy for SVM-TF-IDF model and it is shown below.

**Cleaning**:

****

**Training:**

(i) **Feature selection:** TF-IDF (Models Accuracy and F1 score)

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

(ii)**Feature selection**: Counts (Models Accuracy and F1 score)

**Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated**

(iii)**Feature selection**: Binary (Models Accuracy and F1 score)

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

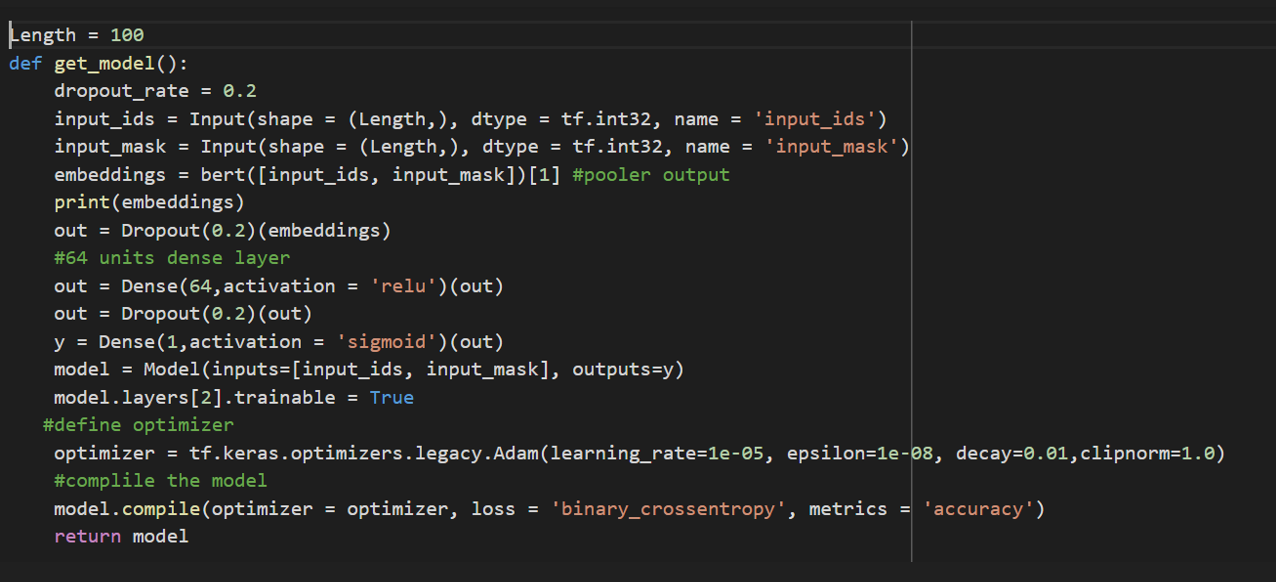
**Proposed Solution:**

We have created a neural network with BERT integration to get better results.BERT is a state-of-the-art pertained language model developed by google. It performs various natural language processing such as text classification, sentiment analysis, question answering and many more. It has greater accuracy because of the ability to capture the context the words in a sentence by considering both from left to right of each word. This is accomplished by employing a bidirectional transformer encoder, a type of neural network architecture.

**Implementation:**

* Once we cleaning the dataset have concatenated the test.csv and train.csv dataset and created a new dataset called as new\_test.csv and split the dataset as x\_train,x\_test,y\_train,y\_test.
* Then we tokenize x\_train and x\_test to convert a list of text documents (X) into tokenized sequences that can be used as input to a machine learning model. The tokenizer object is initialized with the 'Bert-base-uncased' model, which is a pre-trained version of the BERT model.
* Then we creating a neural network model using two dense layer with relu and sigmoid as activation function and compiled the model using Adam optimizer and binary cross-entropy as loss function. As neural network models have shown superior performance on many NLP tasks such as BERT
* Then we are loaded the BERT model.
* Train the Keras model using the fit method, with early stopping as a call back.
* Once the model is trained. We predict the model using test dataset, represented by the input data x\_test\_tokens.
* Then we calculated the evaluation metrics, classification reports.

**Neural Network Model:**



Diagram

Description automatically generated

**Model Fitting:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Classification Report:**

A screenshot of a computer

Description automatically generated with medium confidence

**Graphical results of BERT:**

**Model Accuracy: Model Loss:**

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

**Comparison:**

**Previous algorithm: SVM**

From the above results we can see a better F1 score, and accuracy has been observed from SVM - TF-IDF model and the values are 0.67 and 0.61 respectively.

**Feature selection: TF-IDF (Models F1 score and confusion matrix) using SVM**

A picture containing text

Description automatically generated

**New algorithm: BERT**

We can achieve an accuracy of 0.70 which is greater compared to SVM.

**Classification Report: Confusion Matrix:**

Text

Description automatically generated Chart, treemap chart

Description automatically generated

Chart, bar chart

Description automatically generated

**Conclusion:**

By looking at the above results we can conclude that by using the BERT algorithm we can get a better accuracy of 0.70 which is greater than accuracy with SVM. BERT algorithm due to its sophisticated deep learning architecture and extensive pre-training on enormous amounts of data, BERT performs better than SVM in a variety of NLP tasks since it is better able to understand the context meaning of words in a sentence.

**References:**

1. GitHub link: <https://github.com/kiranrawat/Detecting-Fake-News-On-Social-Media>
2. Wang, William Yang. " liar, liar pants on fire": A new benchmark dataset for fake news detection." *arXiv preprint arXiv:1705.00648*(2017).
3. https://kavita-ganesan.com/news-classifier-with-logistic-regression-in-python/#.X7XeFBNKhQK.
4. Xiang Zhang, Junbo Zhao, Yann LeCun, “Character-level Convolutional Networks for Text Classification,” *IEEE Conference on Neural Information Processing Systems*, pp. 649–657, 2015.
5. Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria, “Recent Trends in Deep Learning Based Natural Language Processing,” I*EEE Computational Intelligence Magazine, vol. 13*, pp. 55–75, 2018.
6. Zhihui Zhu, Jianhua Huang, Yueming Hu, and Xingquan Zhu, "SVM-Based Feature Selection for Multi-Label Classification," *IEEE Transactions on Knowledge and Data Engineering, vol. 25*, pp. 1240-1252, 2013.
7. Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, 2016.