EGR-518

A Multi-Model Approach for detecting and combating Fake News

Abstract

Internet plays a vital role in our daily lives, we use it for various purposes and benefit from advancements in technology and social media. However, the same platforms which make global information exchange also promote spread of fake news,raising a significant threat. To resist this issue, fact checking has become important, leading to extensive research to identify fake news and deal problems arising with them. Our project's mission is to find effective model for fake news detection. We explore different approaches and models, like BERT, Decision Trees, Logistic Regression, and Ada Boost classification and evaluate their performance. We aim to provide valuable insights on this critical fake news issue and show the best performing model among the pool of models.

Introduction

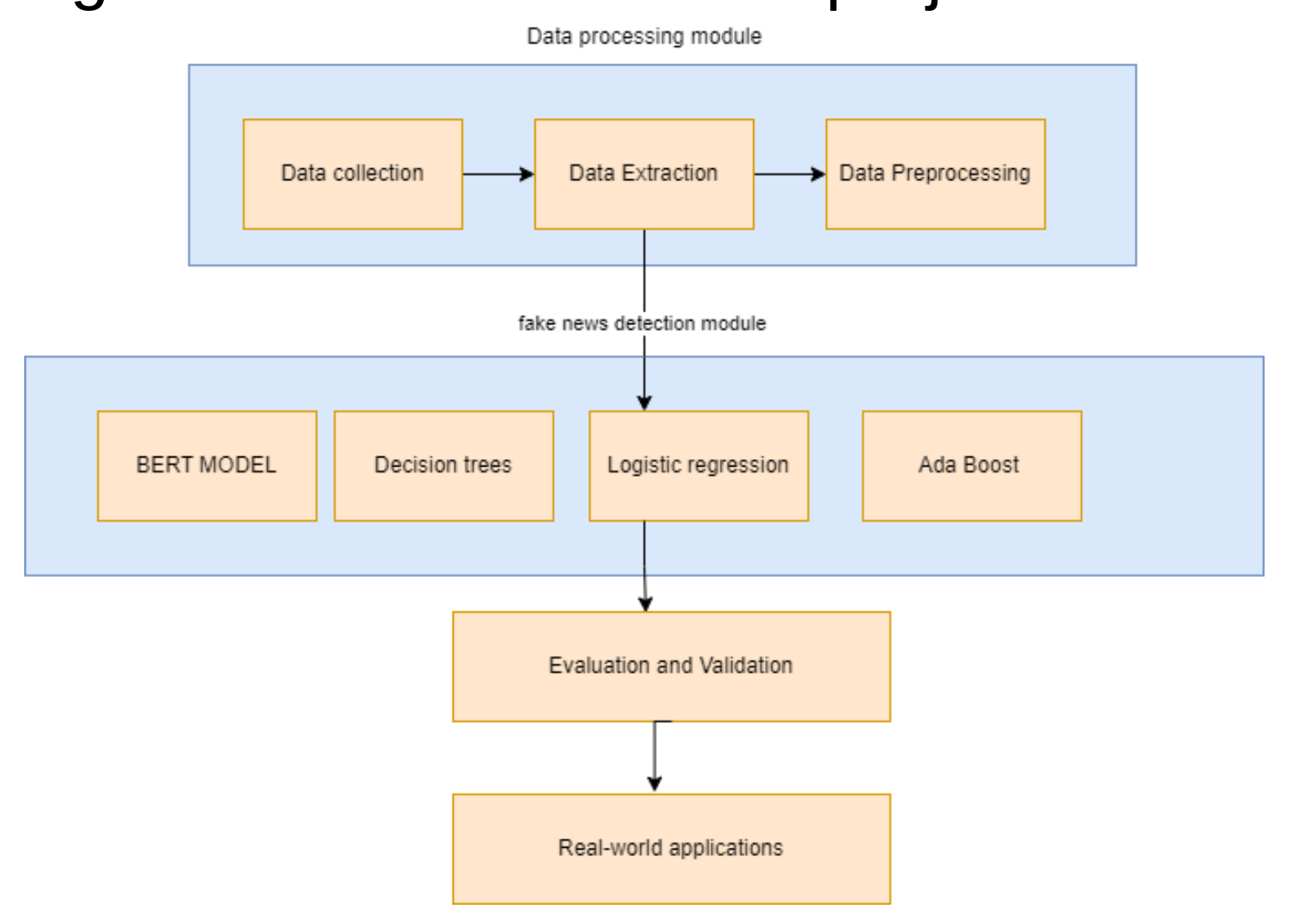
In today's world internet is everywhere, this is due to constant progression of technology. This digital revolution has enabled instant global information sharing through platforms like WhatsApp and Facebook, keeping people informed about events worldwide. It is an integral part of our lives starting with gaining knowledge, communicating with people who are far, entertainment etc and with this impact everyone is transforming themselves.

However, Along with this advancement, this connectivity also brings significant threat and negative impact– the widespread of False information, whether intentional or in- advertent, it can harm society by eroding trust in credible sources. Fact-checking is the remedy, empowering individuals to distinguish between real and deceptive news. Extensive re- search, experiments, and algorithms aim to combat fake news, improving the accuracy and authenticity of information shared online. News organizations embrace advanced technologies for verification.

Materials and Methods

Dataset Description: The dataset employed in this project is sourced from publicly dataset available on kaggle website
We preprocessed the data to clean without noise. Once we processed the data it will be easy to train the model and get accurate results. As text, our data must be converted into numerical data and vectorized. Natural language processing will be used to help us accomplish this.
As a result, we must first clean up the data before we can really translate the words into numbers. Basic preprocessing steps used on data are:
1.Stopwords: Every instance of a stop word that was found in the nltk stopwords corpus was identified and eliminated. The most frequent words in any text that are commonly used words like "a, the, is, and, my etc.". We removed these words during the preprocessing steps.
2. Numerics and special characters: Numerics such as integers, float and special characters such as punctuation marks are removed from the dataset using the regex which only retains the alphabet characters.
3. Lemmatization: We have lemmatized the text afterwards which means to switch any kind of word to its root node, basically grouping all forms of the words which have the same meaning. This is done by cutting down the suffices. WordNetLemmatizer module in NLTK is used to perform Lemmatization.
4.Convert to lowercase: We have lower cased all the texts in order to decrease the distinct words in the corpus, this helped us in making the dataset unique and clean.
5.Tokenization: We have splited the sentences into different tokens using the NLTK word tokenizer. 6. Padding: To make all the input sentences of same length, all the sentences were padded. 7. Removing irrelevant texts: We scanned the dataset and removed the texts which are of length 0, 1 and 2.

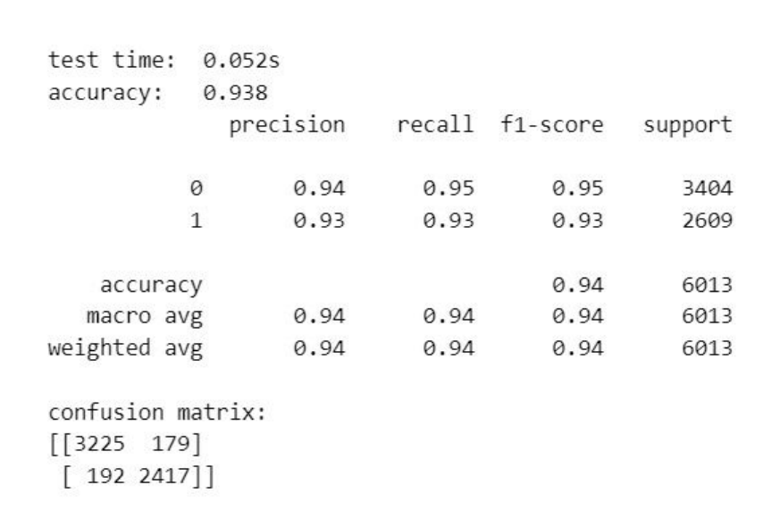
FEATURE EXTRACTION: Vectorization techniques: We are using various vectorization techniques to convert plain text to machine illustrable vectors:
1.OneHot Encoding: Categorical features are frequently preprocessed for machine learning models using One Hot encoding. We employed TensorFlow's one-hot technique, which accepts text input, the vocabulary size, and produces the integer encoding of each word in the text. To ensure consistency in the resulting vectors, all sentences were transformed into one-hot vectors and padded with zero vectors.
2.GloVe Embedding: GloVe is an unsupervised learning approach for producing vector representations of words. We used the glove model, one of the pretrained glove models. Specifically, 6B.300d.txt contains 300d vectors, 400K uncased words, and 6B tokens.
3. Word2Vec Embedding: Word2Vec is an embedding method that turns words into vectors while preserving the relationships between related words. The word2Vec embedding has the shape (300,)
4.Count Vectorizer: We used the count vectorizer to convert a set of sentences which are in text form into a matrix of token counts.
5.Hashing-Vectorizer: The hashing method is used by the hashing vectorizer to identify the token string to feature integer index mapping. It creates a matrix of token occurrences from a set of text sentences
6.TF-IDF and Bag Of Words: Next, we make use of the NLP algorithms TF-IDF (term frequency inverse document frequency) and BOW (bag of words). In essence, these methods aid in data vectorization, which turns text into numeric data. While using BOW, each dictionary key is set to a word, and the value is set to the number of times the word appears. However, while using TF-IDF, the relevance of the string representation (lemma, phrases, words) in a document is quantified among a group of documents. Below fig. is the workflow of our project



Results

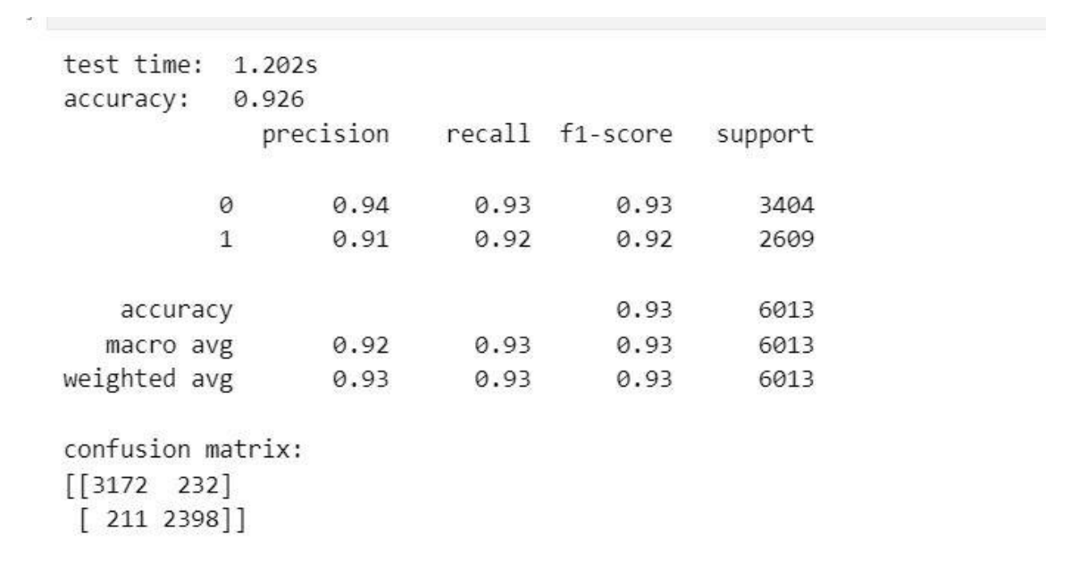
We implemented BERT model, Logistic regression, Decision trees and Ada Boost classification using the methods defined and obtained results 98%,93%,89% and 92% respectively. BERT Model stood out to be best algorithm.

Logistic regression



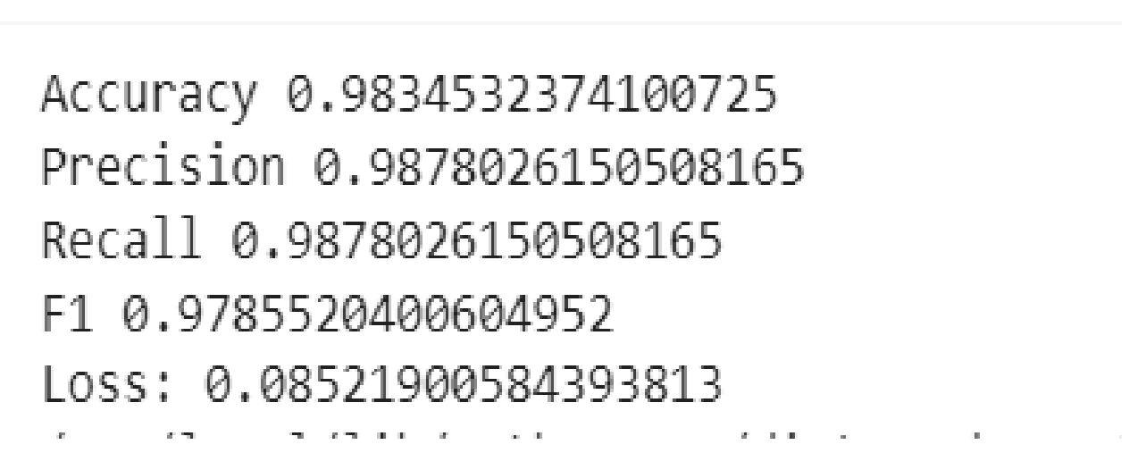
```
test time:  0.052s
accuracy:   0.938
              precision    recall  f1-score   support

           0       0.94      0.95      0.95      3404
           1       0.93      0.93      0.93      2609

    accuracy                           0.94      6013
   macro avg       0.94      0.94      0.94      6013
weighted avg       0.94      0.94      0.94      6013

confusion matrix:
[[3225  179]
 [ 192 2417]]
```

Ada Boost classification



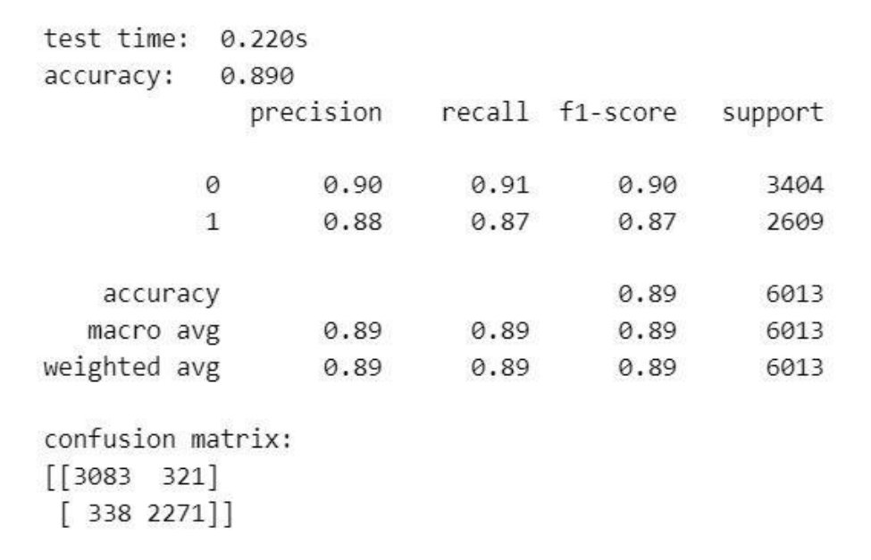
```
test time:  1.202s
accuracy:   0.926
              precision    recall  f1-score   support

           0       0.94      0.93      0.93      3404
           1       0.91      0.92      0.92      2609

    accuracy                           0.93      6013
   macro avg       0.92      0.93      0.93      6013
weighted avg       0.93      0.93      0.93      6013

confusion matrix:
[[3172  232]
 [ 211 2398]]
```

Bert Model :



```
Accuracy 0.9834532374100725
Precision 0.9878026150508165
Recall 0.9878026150508165
F1 0.9785520400604952
Loss: 0.08521900584393813
```

Decision tree:



```
test time:  0.220s
accuracy:   0.890
              precision    recall  f1-score   support

           0       0.90      0.91      0.90      3404
           1       0.88      0.87      0.87      2609

    accuracy                           0.89      6013
   macro avg       0.89      0.89      0.89      6013
weighted avg       0.89      0.89      0.89      6013

confusion matrix:
[[3083  321]
 [ 338 2271]]
```

Conclusions

We evaluated machine learning and deep learning models for fake news detection, including Decision Tree, BERT, Logistic Regression, and AdaBoost Classification. Initial signs favor the BERT-based model over Decision Tree.
Furthermore, we implemented Logistic Regression and AdaBoost Classification algorithms as part of our comprehensive analysis. In the hierarchy of detection accuracy, BERT emerged as the most effective model in discerning fake news among all the algorithms under consideration. This underscores the significance of leveraging advanced natural language processing techniques, such as BERT, in achieving robust and reliable fake news detection outcomes.

Acknowledgments

we extend our sincere appreciation to Professor Md Abdullah Al Hafiz Khan for their invaluable support and guidance throughout our project on fake news detection.

Contact Information

Sujisha Devineni – sdevine9@students.kennesaw.edu

Thrisandhya Bodakunta- tbodakun@students.Kennesaw.edu

References

[1] M. Umer, "Fake news stance detection using deep learning architecture (CNNLSTM)," IEEE Access, vol. 8, pp. 156695-156706, 2020.
[2] R. Barna, R. Maity, D. Minj, T. Barna and A. K. Layek, "F-NAD: An Application for Fake News Article Detection using Machine Learning Techniques", 2019 IEEE Bombay Section Signature Conference (IBSSC), pp. 1-6, 2019.
[3] S. Helmstetter and H. Paulheim, "Weakly Supervised Learning for Fake News Detection on Twitter", 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 274-277, 2018.
[4] Xu K, Wang F, Wang H, Yang B (2020) Detecting fake news over online social media via domain reputations and content understanding. Tsinghua Sci Technol 25(1):20–27. https://doi.org/10.26599/tst.2018.9010139.
[5] Kaliyar RK, Goswami A, Narang P, Sinha S (2020) FNDNet-a deep convolutional neural network for fake news detection. Cogn Syst Res 61:32–44
[6] A. Jain and A. Kasbe, "Fake News Detection", 2018 IEEE International Students' Conference on Electrical Electronics and Computer Science (SCEECS), pp. 1-5, 2018.
[7] R. R. Mandical, N. Mamatha, N. Shivakumar, R. Monica and A. N. Krishna, "Identification of Fake News Using Machine Learning", 2020 IEEE International Conference on Electronics Computing and Communication Technologies (CONECCT), pp. 1-6, 2020.
[8] A. Jain, A. Shakya, H. Khatter and A. K. Gupta, "A smart System for Fake News Detection Using Machine Learning", 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), pp. 1-4, 2019.
[9] A. Kuriakose, D. Sebastian, E. M. Mathew, H. Mathew and G. Er. Gokulnath, "ALIKAH- A Clickbait and Fake News Detection System using Natural Language Processing", 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 1203-1206, 2019.

KENNESAW STATE UNIVERSITY
DIVISION OF GLOBAL AFFAIRS
International Programs in Tuscany

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

Author(s): Sujisha Devineni, Thrisandhya Bodakunta
Advisors(s): Md Abdullah Al Hafiz Khan