Fake News Detection Using State of The Art Model: BERT & Future Advancements, Scope in the Field

Team Trailblazers

Institutional Affiliation(s): NITTE Meenakshi Institute of Technology

Intel Unnati Industrial Training

**Introduction:**

Due to the dynamic nature of news, annotated samples may become outdated quickly and cannot represent the news articles on newly emerged events. Therefore, how to obtain fresh and high-quality labelled samples is the major challenge in employing deep learning models for fake news detection. Fake News is the spread of disinformation and hoaxes through any news platform. The imminent threat of such widespread misinformation is obvious; hence we have looked into ways such Fake News can be identified with the help of Artificial Intelligence. Fake News Detection and analysis is an open challenge in AI!

Annotated samples may quickly become outdated and are unable to accurately depict news items on recently developed events due to the dynamic nature of the news. Therefore, the main obstacle to using deep learning models for false news detection is how to get new, high-quality, labelled examples. Disinformation and hoaxes are propagated through any news outlet as fake news. Given the clear danger posed by such pervasive false information, we have researched methods artificial intelligence might be used to detect fake news. AI is still working on the detection and analysis of fake news.

• The main question is how can this challenge of Fake News Detection be simplified?

A: Various methodologies could include using many other technical stacks such as:

* using APIs such as Google API, newsapi.org, and also some of the Web Scraping techniques could help us keep up with the day-to-day real news by providing us with a real-time check.
* Using Natural Language Semantics and creating models where our model tries to classify and learn from our dataset and its features provided.
* The above two methods can be combined together to create a sure shot classifier model which could predict with the maximum generalization.
* What we’ve done differently***: Instead of only human-generated fake news classification we’ve also added another parameter in our Labels feature and that is AI-generated fake news as nowadays many sources do use AI-generated text for creating Fake News.***

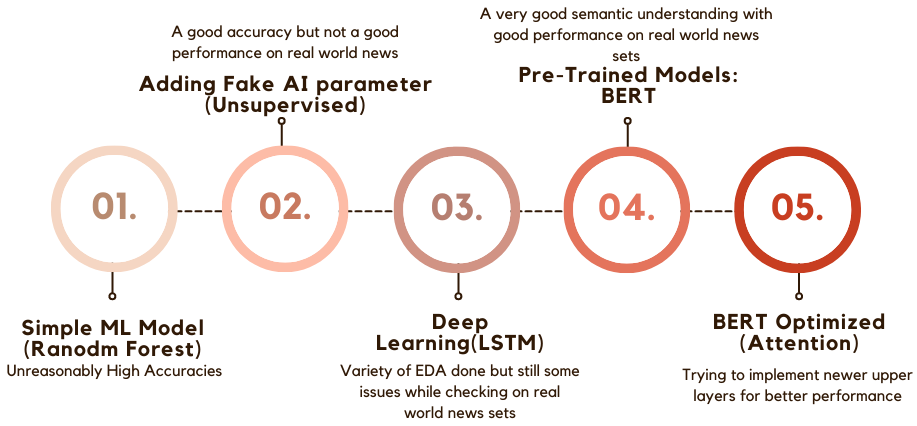
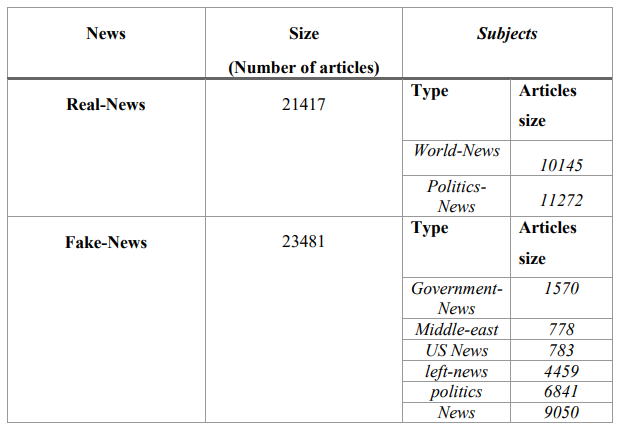
***And a lot more scope for improvement is here.***

**Literature Survey and Related Work:**

After going through multiple research papers and GitHub repositories these were the insights, we obtained about the current best techniques used:

1. FakeBananas: One of the best-documented repositories out there and used a unique methodology based on the concept of stance detection and it also presents you the links of the related articles directly available from different sources.
2. CSI: A hybrid deep model for fake news detection which is Temporal event modelling based on LSTM
3. LIAR: Fake News Detection by Learning Convolution Filters through Contextualized Attention
4. IIT – Guwahati: They created a fake news detection system including the parameter of Clickbait which is an interesting work as most of the fake news today is in the form of clickbait.
5. Grover: is a model for Neural Fake News -- both generation and detection (one of the perfect examples for our use-case)
6. GNN-based Fake News Detection: Pytorch-Geometric implementation of a series of Graph Neural Network (GNN) based fake news detection models.

These were some of the best architectures we came through and these could’ve been improved by adding a few things on top of them, one of the best methods was to use BERT pre-trained model having attention layers added upon them on such datasets which have many features to work upon for getting trusted sources and publishers.

**OUR WORKFLOW**

**Dataset Selection:** We had been given the ISOT Dataset as our base dataset and no restrictions were placed on adding more on top of it, so we tried to add more datasets due to the reason that our news dataset covers only a limited number of domains such as Politics and World-News focused on US & Middle-East governments. This couldn’t be effective for news from different domains, so we combined it with other resources, but it took different pre-processing techniques for each of them respectively before concatenating them and would be attempted later on while pursuing to perfectly complete our project as a whole and being deployed.

**Original Dataset**

**Exploratory Data Analysis:**

The most important step in the ML development cycle is the data preprocessing or the EDA part as it prepares our dataset in such a way that the machine could take in the most important features for classification. We had begun building by using the most Basic ML algorithms such as Logistic Regression, Support Vector Classifiers, XGBoost, Passive Aggressive Classifiers, and Random Forests. We applied preprocessing techniques like Regex and lowering cases to remove unnecessary symbols and link present our data.

* In the beginning, when we ran our models, we noticed that we had very high accuracies nearing 99%. This made us realize that a lot of our data was not pre-processed fully since our model was obtaining near 100% accuracy while only being trained on the data. Then we went back to the EDA, cleaning our data and noticed some features were very distinguishable between the different classes (avg. words per sentence, article lengths, news content). We normalized these features and were able to successfully remove those factors.

Even after getting reasonably good accuracies, the model was not able to predict the real-world recent scenarios as fake or true, so we figured out that either our model was not strong enough to learn semantics or we were required to train or more data, thus we decide to move onto Deep Learning models as the Neural Networks lay a very strong learning foundation which learns a lot of features and is proven more powerful than basic ML models. The link for the complete EDA

*EDA Deep Learning:* There were multiple techniques we applied here for creating word Embeddings like 🡪

* GloVe (Global Vectors for Word Representation): Based on matrix factorization techniques on the word-context matrix.
* Word2Vec: Word2vec is not a single algorithm but a combination of two techniques – CBOW(Continuous bag of words) and Skip-gram model.
* One Hot Representation & Tokenizers: Used in NLP to encode categorical factors as binary vectors, such as words or part-of-speech identifiers.

The best results were obtained by using GloVe embeddings which were about 98.11% on the test set and validation accuracy of about 98.06% using nearly the same model for all the different techniques.

*EDA on Pre-Trained Model BERT (SOTA):* Even though the accuracy did seem very good but there was not enough semantics according to us for the model to learn the way of human interpretations, so we decided to move for one of the state of the art models in Natural Language which is BERT that is trained on Wikipedia’s dataset but didn’t use the large model having 340M parameters due to the increase in computation times.

The transformer model BERT uses sub-word tokenization, here the BERT tokenizer splits the string into multiple substrings and the tokenization we’d done was using BertTokenizerFast for tokenizing the texts of the words on the ‘Title’ of the dataset rather than the ‘Text’ of the dataset. BertTokenizerFast is a tokenizer class that inherits from PreTrainedTokenizerFast which contains most of the main methods. More info here

**Metric & Model Selection:**

After rigorous use of Machine Learning and Deep Learning models, the model we selected was a transformer model ‘bert-base-uncased’ having 12 encoders with 12 bidirectional self-attention heads totaling 110 million parameters. By sacrificing a bit of accuracy, we saved a lot of computation time here as the ‘bert-large-uncased’ is a heavy model pre-trained on the Toronto BookCorpus (800M words) and English Wikipedia(2,500M words) with 24 encoders with 16 bidirectional self-attention heads totaling 340 million parameters. BERT was trained previously trained on Wikipedia’s dataset and thus has a very comprehensive understanding of Human Language Semantics and thus is used extensively in the NLP domain, thus we gave it priority over other trained models. Another option could be the RoBERTa model which has the same base architecture as BERT but has different tokenization and other techniques explained forward.

***A math equations with black text

Description automatically generatedMetrics*** used here was the **Confusion Matrix** for BERT’s initial training as it is considered one of the best to be used in binary classification tasks. It takes into consideration all four values in the confusion matrix and can be defined by the following:

Other metrics that were chosen are the **ROC-AUC curve and Confusion Matrix elements** for evaluating our *ML models* and for *Deep Learning models.*

All the encoding methods used in our project were introduced briefly inside the EDA section and a more in-depth version of the file could be seen on the following link:

**Model Evaluation:**

For our ML-based work, we evaluate our models on the accuracy score and also other measures included in the Confusion Matrix. The average accuracy we’d gained on binary classification was found to be 92.17% and AUC score of 0.8

The model evaluation is done using our metrics such as accuracy and loss. We received the best accuracy by using Glove representations as 97.86% on training data and 97.81% on testing data. Also the ROC-AUC scores were

By using SOTA models such as BERT got the best accuracy to be 89% on BERT but it worked too well while using current news sources as compared to our DL models. Using Tf-idf embeddings with the BERT Architecture got us better accuracy of 93.59% only on 1 epoch, thus have a very high chance of going even higher if trained on 5-10 epochs.

We also tried training on other transformer-based models such as DistilBERT & RoBERTa but were not able to wrap it up completely.

Future Aspects: Instead of going for a different dataset as this dataset doesn't have much info about the authenticity of the sources and we can actually create better models using the attention mechanism and give more weightage to other features rather than only focusing on the label and the context. Also, a method which would be able to support our decisions by displaying links related to the context done with the help of APIs & web scraping.

Clickbait can also be a classification criterion with the inclusion of True and Fake

References:

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A Hybrid Deep Model for Fake News Detection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17). Association for Computing Machinery, New York, NY, USA, 797–806. <https://doi.org/10.1145/3132847.3132877>

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending Against Neural Fake News. [*ArXiv*. /abs/1905.12616](https://arxiv.org/abs/1905.12616)

Pytorch GitHub source for Custom BERT Pipelining -[Pytorch BERT usage example](https://github.com/sugi-chan/custom_bert_pipeline)

Pytorch GitHub Tutorials - [Pytorch Deep Learning framework](https://github.com/pytorch/pytorch)

Clickbait - [Ideas on ClickBait](https://github.com/IITGuwahati-AI/Fake-News-Detection/tree/master)

An application combining ML with APIs - [FakeBananas](https://github.com/likeaj6/FakeBananas/tree/master)