



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 6, June 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.118



9940 572 462



6381 907 438



ijirset@gmail.com



www.ijirset.com

Fake News Detection using Machine Learning Algorithms

Rishabh Sharma¹, Shivam Garg², Anmol Joshi³, Lokesh Kumar⁴, Soumya Sharma⁵

Students, Dept. of Computer Science and Engineering, HMR Institute of Technology and Management, Hamidpur,
New Delhi, India^{1,2,3,4}

Assistant Professor, Dept. of Computer Science and Engineering, HMR Institute of Technology and Management,
Hamidpur, New Delhi, India⁵

ABSTRACT: In recent years, campaigns of lying and disinformation have increasingly prevalent in the India and worldwide, as many social media platforms have acted as a compelling force in their operations, with unscrupulous individuals creating false information for distribution at the click of a button. Spread of false news has far-reaching effects as the creation of biased ideas in dynamic elections results for the benefit of certain people. Additionally, spam senders use attractive news headlines to monetize ads using clickbait's. In this paper, we aim to make binary classification of various available news topics online with the help of concepts related Artificial Intelligence, Natural Language Processing and Machine Learning. Introducing a new tool, a free social web application that provides a clear user interface to get news articles with unreliable information through natural language processing and machine learning methods.

I. INTRODUCTION

As international competition shifts to the 21st digital environment in the century, lying campaigns have become a powerful weapon to sow discord between their opponents. This is a new type of asymmetrical information warfare, in which a concerted effort is made to manage public opinion and exacerbate tensions between groups, thus weakening the intended nation, and ultimately profit in this character's geopolitical interests. As an increasing number of our lives are wasted to communicate online through social media, most people tend to hunt and eat stories from social media instead of traditional news organizations. Definitions of this transformation in consumption behavior is natural within nature of those forums: (i) is usually more timely and very expensive to use news on social media compared to traditional journalism, e.g., newspapers or television; and (ii) easy to proceed share, discuss, and discuss issues with friends or other students on social media. This is incorrect information, or "false news," is designed to target people and is widely distributed media to increase the chances of getting the virus. It goes without saying that sources of verification, critical analysis of information, and to verify the validity of all valid methods for the protection of inaccurate information. These methods work at the individual level, but are time consuming. In a digital culture designed to simplify, we suggest that on a larger scale it is most effective to perform this process automatically to meet the following conditions:

1. Allows the user to verify information with the highest level of confidence.
2. The user experience is friction-less (i.e., only one copy / paste function required to get a guess).
3. We are proposing a new tool, a free social web application allowing users to simply copy / paste into any article, and get a prediction "real news" or "false news" in seconds. This is supported by feed-forward a multi-layer-perceptron, trained in 30,000+ labeled models from the "Fake and Real News" data set.

II. PROBLEM STATEMENT

In our modern age where the internet is available everywhere, everyone relies on a variety of internet news resources. With the increase of use of social media platforms such as Facebook, Twitter, etc. The news spread quickly among millions of propaganda users in a very short time. Spread of false news has far-reaching effects as the creation of biased ideas in dynamic elections results for the benefit of certain people.

2.1 Aim

The objective of this project is to do binary classification of various available news topics online with the help of concepts related Artificial Intelligence, Natural Language Processing and Machine Learning. We aim to give the user the ability to sort issues as fake or real and check the authenticity of news website that publishes news. Introducing a new tool, a free social web application that provides a clear user interface to get news articles with unreliable information through natural language processing and machine learning methods.

2.2 Literature Survey

Trying to identify false news using natural language processing and machine learning is not a new concept. However, much research has been done focused on enhancing certain metrics and benchmarks through complexity structures, including textual metadata, and more.

Therefore, it is our argument that although the methods are interesting from a research point of view, they cannot be used by most people in this way that will solve the problem in the real world. Specifically, we argue that most people encounter issues with an article on Facebook and want to know the reliability of the source / information, they are not going to invest enough in finding the answer to spend time collecting the various metadata about it that these sophisticated models incorporate as features. Instead, we estimate that the machine learning method is more efficient for false news to be used in the real world only text models will be used in an application that can be used by the user to copy / paste only one.

Mykhailo Granik et. al. in their paper [2] exhibits an easy way to get fake news using naïve Bayes classifier. This method was used as a system software and tested against a data set of Facebook news posts. They are grouped into three Facebook pages are one from the right and from left, and three major political issues pages (Politico, CNN, ABC News). They succeeded section accuracy is approximately 74%. The accuracy of the fabrication of the false news is slightly worse. This may be due to data bias: only 4.9% of it is fake news.

Therefore, in our project we focus on our efforts to find a method that is both useful for the production area and that requires less user interaction and / or effort to produce results that increase usability once and for all user interaction. Thus, some of the barriers to our problem space are included.

1. Store all app and model within the memory limit of 500MB up to which is a free example of Heroku under it.
2. Creating a text-only model trained in article writing only. Limit the one copy / paste functionality the user needs to enter the text of the article.

III. TECHNOLOGY USED

We are using Python, NumPy, TensorFlow, Html, CSS and JavaScript.

The specifications of all the technologies used are enlisted one by one in detail in the following subsection.

3.1 HTML

HyperText Markup Language or HTML is a standard language for marking texts that are designed to be displayed in a web browser. It can be aided by technologies such as Cascading Style Sheets (CSS) and writing languages such as JavaScript.

3.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of text written in marking language such as HTML. CSS is the basic technology of the World Wide Web, compatible with HTML and JavaScript. CSS is designed to enable presentation of presentation and content, including layout, colors, and fonts. This segmentation can improve access to content; provide additional flexibility and control in the presentation of presentation elements; allow multiple web pages to share formatting by specifying the appropriate CSS in a separate .css file, which reduces the complexity and duplication of content content; and enable the .css file to be saved in order to improve page load speed between file-sharing pages and their formatting.

3.3 JAVASCRIPT

JavaScript, commonly abbreviated to JS, is a programming language that is one of the core technologies of the World Wide Web, next to HTML and CSS. More than 97% of websites that use JavaScript on the client side for web page behavior, typically include third-party libraries. All major web browsers have a JavaScript engine dedicated to code usage on user devices. JavaScript is advanced, usually the most recent language compliant with the ECMAScript

standard. It has flexible typing, prototype-based object shape, and first-level functions. Multi-paradigm, supports event-driven, functional, and essential planning styles. Contains application programming interfaces (APIs) for text, dates, common expressions, standard data structures, and the Document Object Model (DOM).

3.4 NumPy

NumPy is a Python programming language library, adding support for multiple, multi-layered and mathematical editing, as well as a large collection of high-level mathematical functions running on these arrays.

3.5 TensorFlow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used for a wide range of activities but is more focused on training and pursuing deep neural networks. TensorFlow is a symbolic mathematical library based on data flow and split systems.

3.6 Python

Python is a high-level editing language translated into high-quality translations. Its design philosophy isemphaticreadability of code by its use of critical accuracy. Its language is structured and focused on the object. The approach aims to assist editors in writing clear, logical code for small and large projects.

IV. PROPOSED METHODOLOGY

We took a two-prong approach during our research and development phase, using a feed-forward multi-layer perceptron and BERT fine-tuning, and evaluating the results to determine which model to deploy to production.

4.1 Multi-Layer Perceptron

Our first step was to build a multimedia perceptual training pipeline. We used the code published by Mehmet Lauda Tekman as a starting point and built on that to meet the needs of this project. The main libraries used for this method include NLTK, SKLearn and PyTorch. First, the data is pre-processed to replace non-alphabetic characters in a white space, converted to lower case letters, and to create a tokenizer word from the NLTK library. The tokens are then "fixed" or approved by WordNet lemmatizer from the NLTK library. Next, the feature removal of these graduated tokens is done by a calculation vectorizer from the SKLearn library, which returns the matrix counting matrix. Set the vectorizer to use only 4000 high-quality features / tokens, to keep the complexity of the calculation logical. Then, the model is defined as an input size of 4000 (corresponding to the size of the matrix retrieved for processing work), 3 hidden in-line layers with ReLU opening functions, and an output layer of size 2. Finally, Adam the optimizer is used for reading level very low of 0.001, and the error function used is the loss of entropy, as we do the binary split function.

4.2 BERT Fine-tuning

Our second approach was to use BERT as a feature key by freezing pre-trained parameters and adding a 3 layerlayer that is fully connected at the end of it, which we then train in the database. We used Clement Bissaillon's published code as a starting point and built on it to better fit the needs of the aforementioned project. The main libraries used in this approach were Hugging Face and PyTorch. First, the data is pre-processed into word classification and encoded 300 words at a time with BertTokenizer (using bert-base-uncased), which returns a list of 500-sized tensors. Next, a pre-model BertForSequenceClassification is loaded, its frames are frozen, and a small multi-layer perceptron is added at the end as a dual stage training. Finally, we use a stochastic gradient descent (SGD) with a square-error error (MSE) function with a slightly higher reading value of 0.01. Then we practice for 1 period, repeating in the database once.

4.3 Software Architecture

The software framework is straightforward, with a front end built using plain HTML, CSS, and JavaScript, as well as a background built with a Python Frame frame (see Figure 6: layout diagram). Raw data is copied / pasted into the user's inbox. When the "prediction" button is clicked, an incompatible HTTP request is sent to the Flask / predict' repository. Once the API has successfully acquired the text of the article, the background conclusion performs the same processing steps as it did during the model development, using the NLTK function. However, here we load in memory the calculation of the same vectorizer and the model parameters created during the training loop. In this way it generates a matrix dimension 4000 (representing the top 4000 tokens commonly found in training data) instead of creating a new one that will start counting tokens from scratch again. We chose to use the multi-layer perceptron model in the application due to its high performance (see the test section below), and the fact that using the BERT

model resulted in a total build size exceeding the 500MB Heroku event limit. We have considered the use of low-weight BERT, such as TinyBERT and plan to test this further, however, for the purposes of this project a multi-layered perceptron met our requirements.

Another important consideration in this construction is the scale. This is a proof of the immeasurable evidence of millions of users. However, since the scope of this web application is very limited, and it is actually a single main page that hits the API in the background, there are a few ways we can measure this: 1. Model and background background. API has no status (i.e., if there are multiple API + simultaneous model models sitting behind the load rating, it does not matter which model receives the prediction request from the end, the result is guaranteed to remain the same.). Therefore, this can be easily measured horizontally with the Dockerizing back-end Flask API and model and using the Kubernetes container orchestration to automatically measure the number of scenarios we need to capture current traffic. 2. With millions of people wanting to verify a trendy, popular topic, it may make sense to add a repository layer (such as Redis) between the front and back areas, to avoid re-calculating the same title predictions more than once. This is an easy way for you both to improve performance (delay) and to reduce the load on our back servers.

V. EVALUATION

The key metrics we selected to evaluate the performance of our models were accuracy; however, we also looked at accuracy, memory, and F1 scores. Using a multi-horizontal perceptron, we were able to achieve 98.6% accuracy in the test database, while the BERT precision correction method achieved 96.7% accuracy in the test database. In addition, the multi-layer perceptron performed better than the BERT method for fine tuning across measured metrics. This was somewhat surprising result, as we initially thought that encoding input using a calculator vectorizer would be just an absurd method, as this is actually a matrix training that contains counting the top 4000 tokens that appear frequently in training. data. We hypothesized that BERT coding, used as input in a fully integrated phase layer, would contain more information for the reader to learn from than vectorized tokens embedded in the multi-layer perceptron. Next, we did the matrices of confusion to analyze the speculation of the models very closely.

Interestingly, the multi-horizontal perceptron had a low level of both good and bad lies. In addition, memory requirements were lower in perceptron for more layers than BERT-tuning as well. In fact, the BERT fine tuning model did not meet the constructive obligation size of the HerMB 500MB model. Based on these findings, we decided that a multi-layered perceptron model would be better in this case and invested in production.

VI. EXPERIMENTAL RESULTS

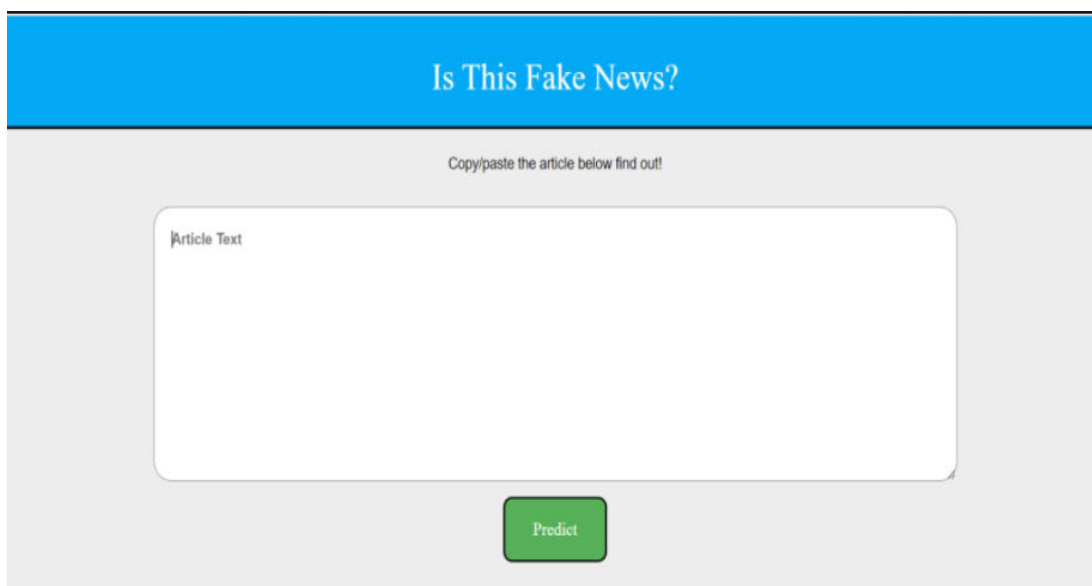
The image shows a web application interface for detecting fake news. It has a blue header with the text 'Is This Fake News?'. Below the header, there is a light gray box containing a text input field with the placeholder 'Article Text'. Above the input field, there is a small instruction 'Copy/paste the article below find out!'. Below the input field, there is a green button with the text 'Predict'.

Fig 1: User Interface Page

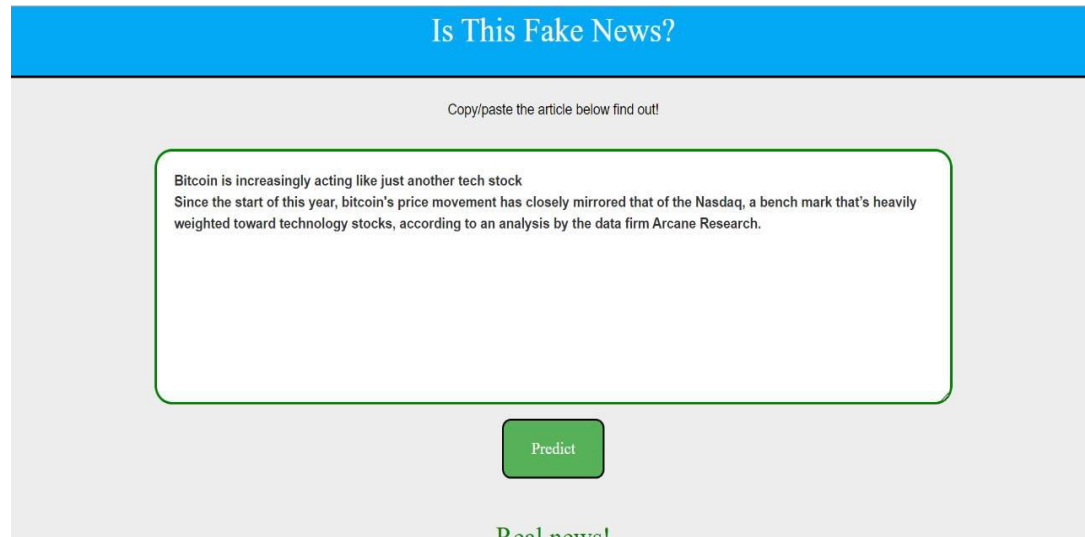


Fig 2: Final Output1

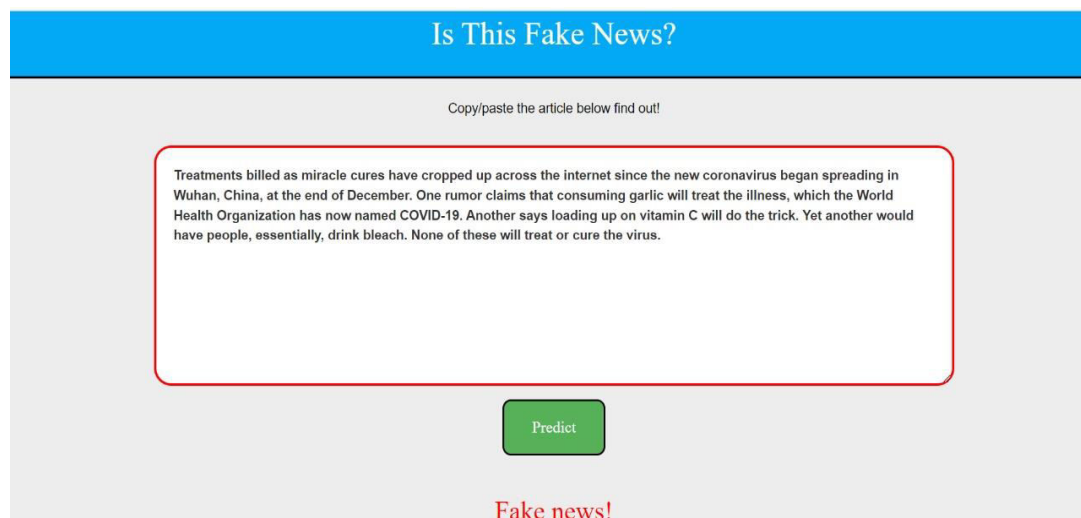


Fig 3: Final Output2

VII. CONCLUSION

Disinformation campaigns are a growing part of the 21st century cyber-threat landscape, and our effective defense will be to develop tools and technologies that reduce the effects of disinformation on a large scale, rather than leave it to individuals. For this project, we have introduced a new tool called *isthisfakenews.ai*, a free public web application that identifies incorrect information about the work of copying / pasting just one. During the research and development phase, we investigated two possible mechanisms: the horizontal perceptron and the BERT correction. After evaluating the performance of the models and the needs and constraints of the project, we chose to use a multi-horizontal perceptron model in the product as it achieved 98.6% accuracy while maintaining a total size of less than 500MB for Heroku. Example is used.

REFERENCES

- [1] U.S. Department of State. Weapons of mass distraction, 2019.
- [2] Wikipedia. Feedforward neural network (multilayer perceptron), 2021.
- [3] Clément Bisailon. Fake and real newss dataset, 2020.



- [4] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection, 2017.
- [5] Mehmet laudaTekman. Detailed fake news classification with pytorch, 2020.
- [6] Wikipedia. Cross entropy, 2021.
- [7] Wikipedia. Bert (language model), 2019.
- [8] ClémentBisaillon. Classifying fake news with bert, 2020.
- [9] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding, 2020.



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details