2020 IEEE Pune Section International Conference (PuneCon)
Vishwakarma Institute of Technology, Pune India. Dec 16-18, 2020

Fake News Detection System using Web-Extension

Yash Khivasara
School of Computer Engineering and Technology
MIT World Peace University
Pune, India
khivasra.yash@gmail.com

Yash Khare
School of Computer Engineering and Technology
MIT World Peace University
Pune, India
yashkharess10@gmail.com

Tejas Bhadane
School of Computer Engineering and Technology
MIT World Peace University
Pune, India
tejassb1999@gmail.com

2020 IEEE Pune Section International Conference (PuneCon) | 978-1-7281-9600-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/PuneCon50868.2020.9362384

Abstract—**Internet is a supreme one-stop source of information that enables the sharing of news and curated user-content at a rapid, effortless, and in a routine manner. News is a global medium of daily events worldwide, offering absorption of quick information. With ample availability of news content online, these news articles has by-products in information generation in both ways - real and fake news. Considering the context and volume of information shared online, it is challenging to establish authenticity of news. This leads to the immense growth of fake news on various websites, which can lead to serious concerns in society, fading away the correct news content to reach the users creating misconceptions and deceived views of the readers. To ensure the readers have the credibility of the content, we propose a web-based extension enabling them to distinguish from the fake and real news content. The proposed web extension in the paper uses multiple deep learning models. The first is based on our model trained on LSTM, and the other uses OPEN AI's well-developed AI-generated text classifier GPT-2. The devised web-extension displays both probabilities of news being either AI-generated or written by an individual.**

Keywords—**AI (Artificial Intelligence), Web Extension, LSTM (Long Short-Term Memory), GPT-2 (Generative Pre-trained Transformer)**

I. Introduction

Fake news is intentionally misleading content brought forward under the cover of responsible journalism, which spreads like wildfire. It is either generated to manipulate people's perceptions of reality or to attract viewers for collecting advertising revenue. The credibility of news is one thing, but being able to differentiate between genuine and fake news has been a concern over the past decade. For example, a study shows that ”45% of the British public believe they encounter fake news online every single day” [1].

Fake news affects us not only mentally but also socially and economically. Mistaking fake news as real-life events leads to drastic impacts on society. For example, in 2013, a fake tweet claiming that Barack Obama, was injured in an explosion [2] created a socio-economic impact by wiping out the stock value of $136 billion and leading to chaos in society. Thus, to mitigate the problem of fake news for the readers, we present a web-extension that enables users to check the score of the news being fake without the hassle of changing the webpages.

A. Related Work

There are several approaches for detecting fake news by using machine learning. Previously experimented approaches include classification models such as Naïve Bayes, SVM, Random Forest, Logistic Regression, and Recurrent Neural Network.

Many works involved experimenting with several variants of Recurrent Neural Networks, such as Vanilla LSTM, unidirectional and Bi-directional LSTM. These methodologies significantly improved accuracy over other classification models. For instance, K.Greff et al. [3] described that forget and output activation functions are the most critical components of the LSTM block. They also mentioned that tuning only one of the hyper-parameters, such as learning rate or network size rather than both, saves a lot of experimental time.

Prannay S Reddy et al. [4] used Doc2Vec feature extraction in data pre-processing. The pre-processed data was used with various classifier models such as Naïve Bayes, SVM, and LSTM. The proposed model that used LSTM had the highest accuracy of 94% among all the other models.

Damian Mrowca et al. [5] used the headline-article pair of the news. They used bidirectional LSTM with global features to predict the stance of the news as agree, disagree, discuss, or not related.

Terry Traylor et al. [6] suggested using an attribution-based fake news detection tool that used only one feature extraction to classify an article with an accuracy of 69.4% on the test dataset. Monther Aldwairi and Ali Alwahedi [7] proposed a tool that can identify and remove fake websites based on user feedback.

II. Overview

According to a survey [8], most of the users are accustomed to reading news articles on websites and sharing them on social media. The objective is to create a system, which is user-friendly, cross-platform, and accurate for detecting all types of fake textual content. For this, a web extension can be used as an assistive tool by the users. This extension can classify two types of news or articles:

1) Fake content generated by a human.
2) AI-generated content

A user would download our web-extension from the official web extension store of the browser. The functionality of the extension would come into play while surfing the web. Whenever the user finds any suspicious article or news, the user clicks on the icon of the extension. A call is made to both the

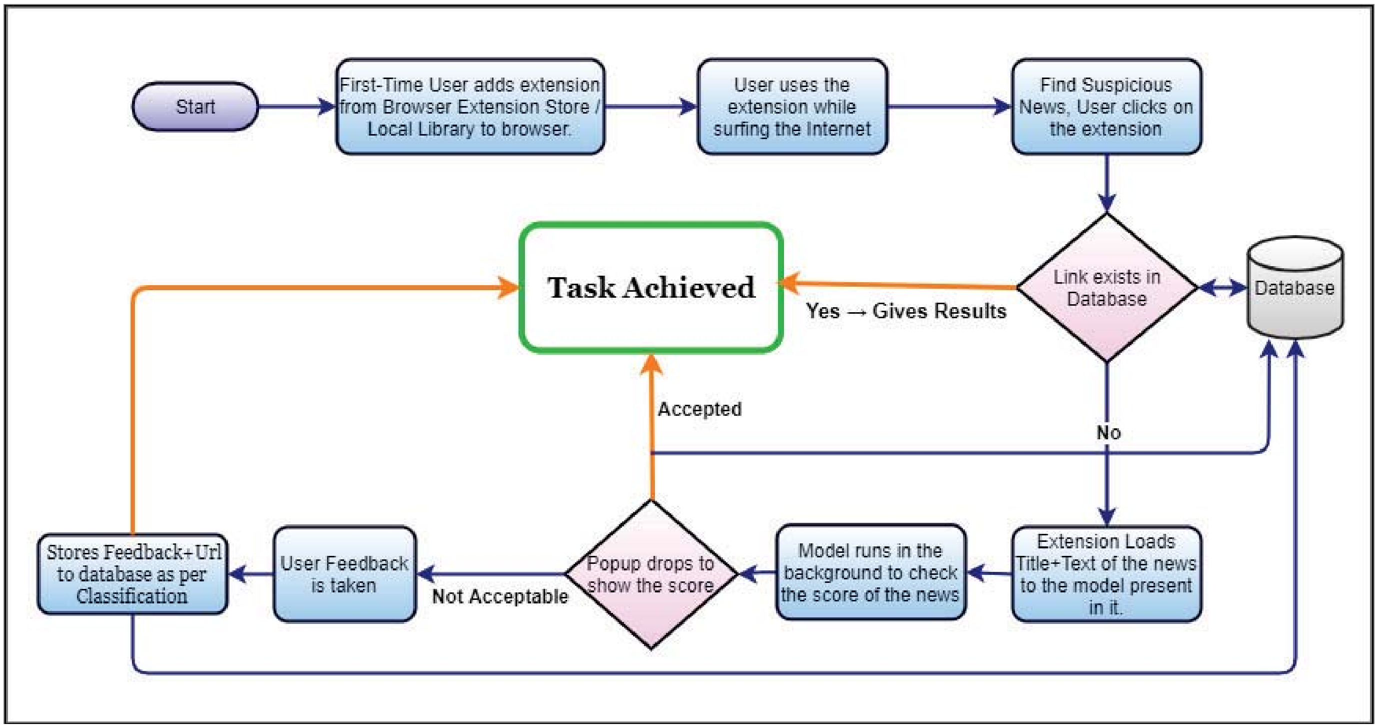

Start
First-Time User adds extension from Browser Extension Store / Local Library to browser.
User uses the extension while surfing the Internet
Find Suspicious News, User clicks on the extension
Link exists in Database
Database
Task Achieved
Yes → Gives Results
Accepted
No
Extension Loads Title+Text of the news to the model present in it.
Model runs in the background to check the score of the news
Popup drops to show the score
Not Acceptable
User Feedback is taken
Stores Feedback+Url to database as per Classification

Fig. 1. Overall Flow Diagram of Fake News Detection System

'Algorithmia' API [9] of our deployed model and the GPT-2 AI Detector [10] to produce the respective outcome. The results are displayed immediately. In case of unsatisfactory results, users can dislike the outcome, and in turn, flag the content to its category accordingly. Users can also give an authentic source for the content being flagged.

III. Dataset

The primary dataset used in the project is a collection of news articles obtained from open Machine Learning Repository 'Kaggle' [11] [12].

TABLE I
Dataset Specification

Sr. No.	Attribute	Type	Description
1.	ID	Int	Unique identifier for each row in the dataset
2.	URL	Text	Specifies the URL from which the content was extracted
3.	Title	Text	Specifies title of the News or Article
4.	Body Text	Text	Specifies content of the News or Article
5.	Label	Text	Specifies content label as either REAL or FAKE of the News or Article based on Title + Body Text

This collection consists of news articles from authentic news sources such as NY TIMES [13], BBC [14], and Beforeitsnews [15]. After removing duplicate records, the final dataset consists of 45836 records.

A. Feature Description

Each entry in the data frame consists of features: ID, URL, Title, (Body Text) Content, and text label as REAL or FAKE. To ensure there is no bias present in the data, the number of real and fake labels are approximately balanced.

- Real:24345
- Fake:21055

IV. Approach and Implementation

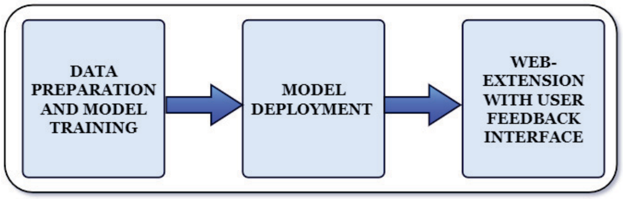

DATA PREPARATION AND MODEL TRAINING
MODEL DEPLOYMENT
WEB-EXTENSION WITH USER FEEDBACK INTERFACE

Fig. 2. Technological Flow

A. Data Preprocessing

The data that we accumulated was quite inconsistent for taking it directly for training our model. To mitigate this, 'Title + Text' was pre-processed using NLTK [16]. We removed URLs, double spaces, punctuations, special characters, and stopwords. For converting words to vector matrices, we tested our dataset on the best-known word embedding models such as FastText [17], Word2Vec [18], and GLoVe [19]. FastText improves on Word2Vec by also considering word parts. The limitation of FastText is that it works better on smaller datasets.

However, for a larger dataset, the additional benefits of using GloVe over word2vec and FastText are that it is easier to parallelize the implementation, which means it is easier to train over more data. GloVe captures both global statistics and local statistics of a corpus, in order to come up with word vectors. The final step in pre-processing is to create a dictionary mapping word to integers. The title and text were concatenated to be used as a single feature. The dataset was sliced into the [80:10:10] ratio for training, testing and validation, respectively. The data entries were taken as [36668], [4584], and [4584].

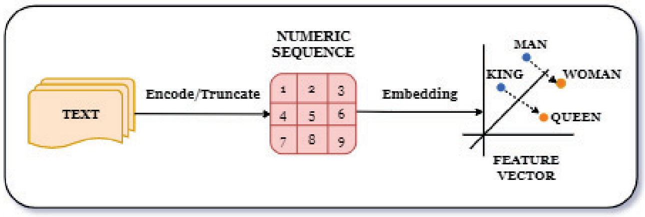


Fig. 3. Data Pre-processing

B. Model Training and Architecture

a) Word Embedding: A word embedding is simply representing texts as numbers. To convert words to vectors, each word is one-hot encoded. For small numbers, the vectors are randomly initialized. For example, if we know the vectors of Man, King, and Woman, subtracting ”Man-ness” from the King and adding ”Woman-ness” would result in a Queen vector.

$$\text{(King - Man + Women)} \approx \text{(Queen)}$$

The size of the vector space in GloVe is defined as part of the model in various dimensions such as 50, 100, or 300. We considered GLoVe’s pre-trained size of dimensions 50 for the best results. We used an in-built Keras layer for word embedding of input nodes. Parameters for the input and output size were set at 500 and 50, respectively.

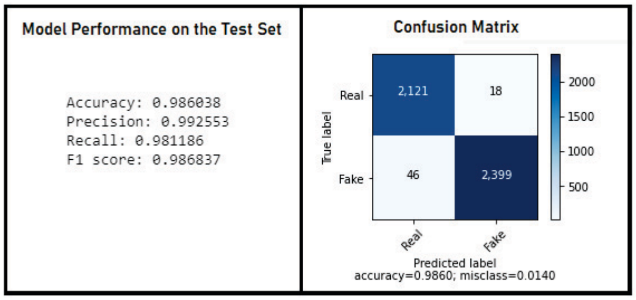
b) Long Short Term Memory: The problem with Recurrent Neural Networks (RNN) is it’s short-term memory. If a sequence is long enough, they will find it difficult to carry information from earlier steps to later ones. So, if we try to process a text paragraph to make predictions, RNN’s might leave important information out from the start. This is where LSTM comes into the picture to tackle the problem of short-term memory. LSTM has a control flow, which is similar to a recurrent neural network. It processes information that is passed on as it propagates. In addition to RNN structure, there are operations inside the cells of LSTM. These operations are used to help keep or forget information from the LSTM cell.

The architecture of our model involved one input word embedding layer, three hidden layers, and an output layer. The first input layer consisted of 50 neurons as features; the three hidden layers consisted of 128, 64, and 32 neurons, respectively. The activation function for hidden layers is ’ReLU’ (Rectified linear activation function). It overcomes the problem of the vanishing gradient, allowing the models to learn more quickly and perform better. To implement regularization, we used dropout in each hidden layer. For the output layer, the ’SIGMOID’ activation function was used. The model is then compiled using optimizer as ’Adam’ and loss function as ’binary_crossentropy’. Adam amalgamates the features of both algorithms ’AdaGrad’ and ’RMSProp’, to provide an optimization algorithm for sparse data that contains noise.

c) Results & Comparison of Model: Two different models are trained on the same dataset mentioned above. Even after training both models for several iterations and tuning hyperparameters, LSTM + GloVe (Fig. 4) proved to perform better than Naïve Bayes.

TABLE II
COMPARISON OF MODELS

Sr. No.	Model	Accuracy
1.	LSTM + GloVe (Proposed Model)	98.6%
2.	Naïve Bayes	88.91%



Model Performance on the Test Set

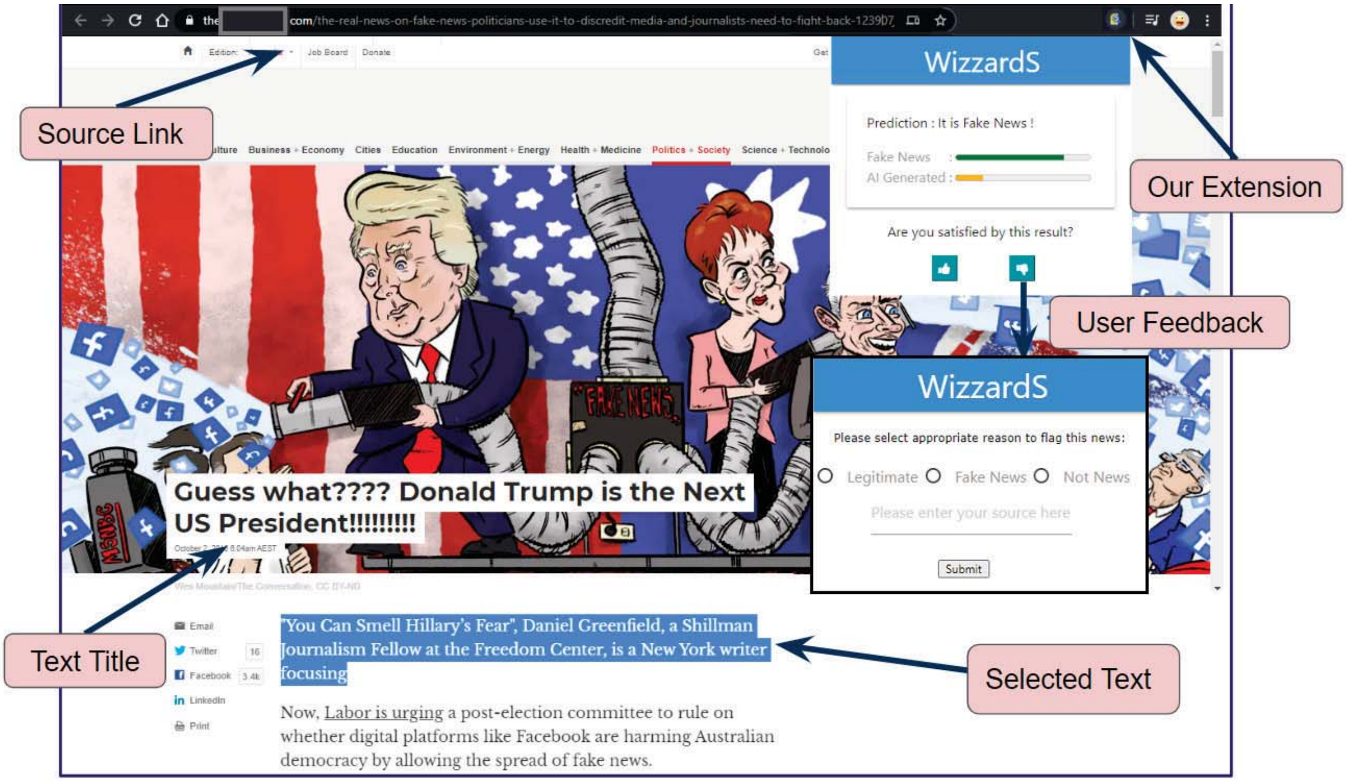
```
Accuracy: 0.986038
Precision: 0.992553
Recall: 0.981186
F1 score: 0.986837
```

Fig. 4. Result of our Model

C. Extension

a) Frontend: For every instance, when a user selects text and click on the extension, JavaScript is used to extract URL (Uniform Resource Locator) of the webpage. It checks if the URL is present in the blacklisted URLs saved in the database. If the URL is present, it shows the output as fake content; otherwise, it gets passed on to the models.

b) Backend: Two different API platforms, i.e, Algorithmia & GPT-2, are used to check whether the content is fake or not.



the[redacted]com/the-real-news-on-fake-news-politicians-use-it-to-discredit-media-and-journalists-need-to-fight-back-123907

WizzardS

Prediction : It is Fake News !

Fake News :

AI Generated :

Are you satisfied by this result?

WizzardS

Please select appropriate reason to flag this news:

○ Legitimate ○ Fake News ○ Not News

Please enter your source here

Submit

Guess what???? Donald Trump is the Next US President!!!!!!!!!

"You Can Smell Hillary's Fear", Daniel Greenfield, a Shillman Journalism Fellow at the Freedom Center, is a New York writer focusing

Now, Labor is urging a post-election committee to rule on whether digital platforms like Facebook are harming Australian democracy by allowing the spread of fake news.

Fig. 5. Overall Flow Diagram of Fake News Detection System [20]

- Model using LSTM and GloVe: The model is deployed on 'Algorithmia', a ML model deployment solution. Algorithmia generates a unique API key and uses it to interact with the extension. The user-selected content acts as an input to the deployed model. The result generated by it is fetched by the extension.
- GPT-2: Hugging Face GPT-2 Output Detector is used to detect AI-generated text. This detector is used in the extension to classify whether the news is AI-generated or not. The selected text is sent to the detector, and results are fetched by the extension.

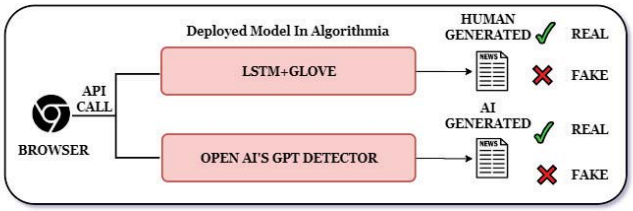


Fig. 6. Web Interface

c) Result Display: The results of both models are shown to the user in a popup notification generated by the extension.

- Fake Content: If the result is 'fake', the URL of the web-page is saved in the real-time database for future reference.
- User Feedback: In case, if the user is not satisfied with the result given by the model, he can flag the malicious content. After selecting the option of "not satisfied", the user is redirected to a popup for flagging. As shown in Fig. 5, users can classify and submit, news as "Legitimate news, Fake news, or Not a news" with an authentic source. This record is stored in real-time with its type in the database.

V. Conclusion and Future Work

In this paper, we propose a simple but effective system to detect fake news. We built a web extension that alerts the readers about the prevalence of fake news and AI-generated news on different media websites using a combination of two deep learning models - our model based on LSTM and the OPEN AI's GPT-2 Output Detector. LSTM is well suited for a long-range semantic dependency-based classification, while GPT-2 marks the AI-generated text. This current model was tested against the existing dataset and shows that it performs significantly well. The extension also offers readers the option to report content with a credible source if the user is not happy with the model performance.

For future work, news articles data can be considered related to recent incidents in the corpus of data. The next step then would be to train the model and analyze how the accuracies vary with the new data to improve it further. This model can also be implemented on chatbots or web applications.

ACKNOWLEDGMENT

We thank our mentor, Prof. Avadhoot Joshi, for his guidance and advice throughout the project.

REFERENCES

[1] Based on the survey publish, https://journolink.com/resource/319-fake-news-statistics-2019-uk-worldwide-data

[2] News Article, https://www.telegraph.co.uk/finance/markets/10013768/Bogus-APtweet-about-explosion-at-the-White-House-wipes-billions-off-USmarkets.html.

[3] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, ”LSTM: A Search Space Odyssey,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232, Oct. 2017.

[4] Reddy, Prannay & Roy, Diana & Manoj, P & Keerthana, M & Tijare, Poonam. (2019). A Study on Fake News Detection Using Naïve Bayes, SVM, Neural Networks and LSTM. Journal of Advanced Research in Dynamical and Control Systems. 1. 942-947.

[5] Mrowca D, Wang E, Kosson A. Stance detection for fake news identification. Stanford University, California, US, rep.. 2017.

[6] T. Traylor, J. Straub, Gurmeet and N. Snell, ”Classifying Fake News Articles Using Natural Language Processing to Identify In-Article Attribution as a Supervised Learning Estimator,” 2019 IEEE 13th International Conference on Semantic Computing (ICSC), Newport Beach, CA, USA, 2019, pp. 445-449.

[7] Aldwairi M, Alwahedi A. Detecting fake news in social media networks. Procedia Computer Science. 2018 Jan 1;141:215-22.

[8] Based on the survey publish, https://timesofindia.indiatimes.com/india/huge-jump-in-no-of-peoplereading-newspapers-for-over-an-hoursurvey/articleshow/75336131.cms

[9] Algorithmia, https://algorithmia.com/

[10] GPT detector, https://huggingface.co/openai-detector/

[11] Fake and real news dataset, https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[12] Fake News detection, https://www.kaggle.com/jruvika/fake-newsdetection

[13] NY Times, https://www.nytimes.com/

[14] BBC, https://www.bbc.com/news

[15] Beforeitnews, https://beforeitsnews.com/

[16] NLTK: Natural Language Toolkit, https://www.nltk.org/nltk_data/

[17] FastText, https://fasttext.cc/

[18] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013 Jan 16., https://code.google.com/archive/p/word2vec/

[19] Pennington, Jeffrey & Socher, Richard & Manning, Christoper. (2014). Glove: Global Vectors for Word Representation. EMNLP. 14. 1532-1543. 10.3115/v1/D14-1162., https://nlp.stanford.edu/projects/glove/

[20] Demonstration, http://bit.ly/WizzardS