RESEARCH ARTICLE

DOI: 10.34256/ijcci2324

International Journal of Computer Communication and Informatics

Deceptive Content Analysis using Deep Learning

Hritik Gupta [a,*], Divyam Pal [a], Palak Sharma [a], Krishna Raj [b], Deep Kumar [a], Sachin Kumar Tyagi [b]

[a] Department of CSIT, KIET Group of Institutions, Ghaziabad, Uttar Pradesh-201206, India

[b] Department of ECE, KIET Group of Institutions, Ghaziabad, Uttar Pradesh-201206, India

* Corresponding Author: hritik162001@gmail.com

Received: 16-08-2023, Revised: 29-11-2023, Accepted: 13-12-2023, Published: 21-12-2023

Abstract: Fake news is the deliberate spread of false or misleading information through traditional and social media for political or financial gain. The impact of fake news can be significant, causing harm to individuals and organizations and undermining trust in legitimate news sources. Detecting fake news is crucial to promote a well-informed society and protect against the harmful effects. Tools such as machine learning and natural language processing are being developed to help identify fake news automatically. Necessity of fake news detection is very important to maintain a trustworthy and responsible media environment. We have used Word2Vec model for word vectorization and represents words in a multi- dimensional space based on their semantic and syntactic relationships. The use of the LSTM with 256 units allows our model to capture the sequential nature of the data and make predictions based on past information. The proposed model uses Word2Vec and LSTM models to provide a powerful approach to fake news detection, combining the ability to capture the complexity of language and the sequential nature of the data. The model has the potential to accurately detect fake news and promote a well-informed society. The accuracy achieved by building the model was 97%.

Keywords: Fake news, LSTM (Long Short-Term Memory), Word vectorization, Word2Vec.

1. Introduction

Fake news has become a pervasive problem in our society, causing harm to individuals and communities by spreading misinformation and sowing seeds of distrust in the media and democratic institutions. It is crucial that we all take responsibility for verifying the accuracy of information before sharing it and actively seeking out reputable sources of news and information. The spread of false information has the potential to create mass confusion and incite anger, particularly when it comes to sensitive topics like politics, health, or national security. Moreover, it can also undermine trust in journalism and institutions responsible for verifying the authenticity of news. The situation has become even more complex with the rise of deepfake technology.

which allows for the manipulation of audio and video to create false content that appears to be real. Fake news has impacted various domains, including politics, sports, health, and science, and has led to many negative consequences, thus making it mandatory to check information before sharing it with others. As the spread of misinformation continues to be a pressing issue, there is an urgent need for effective and efficient fake news detection mechanisms. Recent advancements in machine learning and natural language processing have opened new possibilities for detecting fake news. The spread of fake news has become a major concern in recent years, particularly with the rise of social media and the ease of sharing information online.

Fake news can cause real harm by spreading misinformation and influencing people's beliefs and actions. The research by Common Sense Media showed that young people are no better than the rest of the society in identifying fake news and that they may unknowingly spread false information. This situation has raised concerns about digital literacy and the need for individuals to be able to navigate and critically assess the information they encounter online. Researchers have developed numerous machine learning models that utilize linguistic features. The commonly used models for training include decision trees, support vector machines (SVM), and logistic regression. Additional features such as user engagement and stance detection are often employed to improve the results of these models. Some studies have even incorporated social and psychological theories and data mining algorithms to construct these models. A study was conducted in the political domain to determine the effectiveness of combining text and speaker features, and the metadata elements apart from this, to achieve accuracy up to higher extent in natural language processing. Results showed that 27.7% was the accuracy achieved when text and speaker features were combined, and 27.4% when all metadata elements were combined with text. Another approach to stance detection was proposed by Riedel et al., which involved classifying articles as "agree", "discuss", "disagree", or "unrelated" based on the article's headline and text. Linguistic properties were the methods used, along with the term frequency and term frequency-inverse document frequency, as features, and a multilayer perceptron (MLP) classifier with one hidden layer and a SoftMax function on the output of the final layer. The model achieved an accuracy of 88.46% in the overall classification of articles.

2. Literature Review

In the research paper [1], a linguistic model is created to find out the properties of data which contains news articles by which Language driven features will be created. The model finds syntactic features, grammatical features, readability features and sentimental features from the news article to process it into fake or real. Dataset used were Buzzfeed Political News Data and Random Political News Data. Both datasets are available as Horne2017_FakeNewsData1 repository. Methodology was that model was classified into three models. Model 1 contains Syntax based features, Grammatical features and Sentiment based features represented by a group of 13 features. Model 2 contains Readability features represented by a group of 87 features. [1] Model 3 contains all the lingual features, i.e., 20 in count [1]. It results in a total accuracy of

84.52% with Epoch 100 batch size 5 and 10 using model 3. Results are also compared with LSTM model; training time of LSTM model was much more than that of our model therefore LSTM model was excluded.

The research paper [2] employs stance detection technique to identify fake news. The FNC-1 dataset is utilized to train the model, which consists of news articles and their corresponding headlines, as well as a label indicating their relatedness or stance. The dataset is categorized into four stances: agree, unrelated, disagree and discuss. To enhance the results, preprocessing techniques such as stop word and punctuation removal are employed. Various word embedding methods are used to train dense neural networks. In the first study, the TF-IDF approach is utilized for embedding, and the embedded vector serves as the input to the neural network, resulting in an accuracy of 94.3%. In the next study, the bag of words (BOW) technique is combined with a neural network, achieving an accuracy of 89.23%. Finally, the Word2vec method is employed in a model that delivers an accuracy of 75.67%. Next, we picked "Long Short-term Memory": Recurrent Neural Network (RNN) which is a type of neural network that is commonly used to make predictions based on sequential input data. LSTMs is a variation of RNNs that was specifically designed to address the problem of long-term dependencies in large text inputs. Like RNNs, LSTMs also have a chain-like structure, but they use a Cell state to retain relevant information, discard irrelevant information, and update data to a new Cell state. LSTMs have three essential gates, namely the Forget gate, Input gate, and Output gate, to carry out their operations. The Forget gate is responsible for removing extra information from the Cell state.

It uses a Sigmoid function to predict a value between 0 and 1 based on a combined vector of the current input and previous output. The Input gate is used to store information in the Cell state, and it uses two layers, one of which is a Sigmoid layer, and the other is a tan layer. Finally, the Output gate determines how much information to use for output. In conclusion, LSTMs are a type of RNN architecture that are effective at modelling complex sequential data, due to their capability to handle long-term dependencies and learn high-level representations. However, they do have limitations such as the inability to handle long-term temporal dependencies and a limited context window size. To address these limitations, researchers have proposed different solutions such as training larger LSTMs, using Attention-based models or GRUs, or exploring alternative neural network architectures such as the Neural Stack Machine. Despite these limitations, LSTMs continue to be a popular choice in various applications due to their effectiveness in many tasks and their efficient gradient- based learning algorithm.

3. Methodology

3.1 Dataset Used

To pick a perfect Dataset to match our expected accuracy rate we need a dataset which is not too small and contains Real time news from Twitters, News Websites, News channels,

Podcasts and from Newspapers. We can also combine different Datasets to make our news Dataset which has all our desired features to train our model.

Our Dataset is taken from Kaggle named as WelFake_Dataset it contains 72134 entries out of which we have 35028 Real News and 37106 Fake News Table 1.

Table 1. Specifications of Wel Fake_Dataset

Attribute	Type
unnamed	Numeric
title	Text
title	Text
label	Numeric (0 or1)

3.2 Word Coud

A word cloud is a visual representation of the most frequently used words in a text or speech. It typically displays the words in different sizes and colors, with the size and color of each word indicating its frequency of use in the text Figure 1. The idea behind word clouds is to provide a quick, easy-to-understand overview of the key themes and topics discussed in a piece of text.

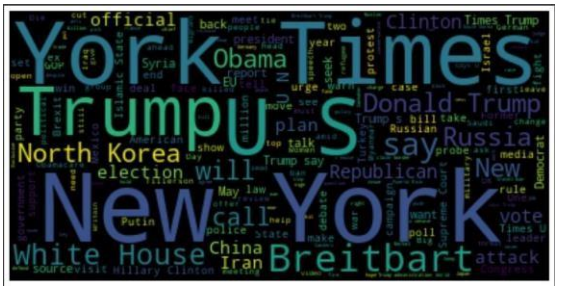

Figure 1. Word Cloud of title attribute with label 0 (Fake)

3.3 Pre-Processing

- First Convert data in text and title attributes to lower case.
- Second merge both 'text' and 'title' attributes of our database. Both attributes contain news so it is perfect to merge them, which will help us in model training and maintain our performance.
- Third removal of null rows from the database.
- Fourth removal of Special characters like, ".; / [] -?! @.

- We used a spaCy, beautifulscoup4 and textblob and a pre-defined library from kgptakie- https://github.com/laxmimerit/preprocess_ kgptalkie.git
- spaCy is an open-source software library for advanced Natural Language Processing (NLP) in Python.
- Beautiful Soup is a Python library used for web scraping and pre- processing of unstructured data, especially HTML and XML.

3.4 Proposed Model

3.4.1 Word2Vec

Word Vectorization is a crucial step in the field of machine learning and nlp (natural language processing). The process of word vectorization involves converting words into numerical vectors, which can then be fed into machine learning models for training and prediction tasks. The purpose of this conversion is to provide the machine with a numerical representation of words so that it can understand and work with textual data. There are several methods for word vectorization, including one-hot encoding, count- based methods, and prediction-based methods. One-hot encoding represents each word in a vocabulary as a binary vector, where each dimension represents a unique word in the vocabulary. However, one-hot encoding is not suitable for representing the semantic meaning of words and their relationships [3]. Count-based methods, on the other hand, represent words as the frequency of their occurrence in each corpus. Although count- based methods capture the frequency of words, they do not consider the semantic meaning of words. Prediction-based methods, such as Word2Vec, are more advanced and sophisticated. Word2Vec is a widely used word vectorization technique that represents words in continuous vector spaces. It operates by learning the relationship between words in a corpus and their context, represented by surrounding words. The technique uses a neural network to determine the target word given its context. The hidden layer of the neural network can be thought of as a dense representation of words, which can be used for information retrieval i.e., to understand the deep contextual meaning of the given input text. The dimensions of the word vectors are an important factor that affects the performance of a word vectorization model. A high-dimensional vector can capture the complex semantic relationships between words, but it also increases the computational cost of the model and can lead to overfitting. On the other hand, a low-dimensional vector may not capture the complexity of the relationships between words. The choice of the dimensions of the word vectors depends on the specific task and the corpus being used. A common dimension used in Word2Vec is 100.

3.4.2 LSTM (Long Short-Term Memory)

Long Short-Term Memory (LSTM) has been widely used in various applications, especially in areas where the input data is sequential in nature such as nlp (natural language processing), speech recognition, and financial time series prediction. It is one of the most popular

deep learning models for sequential data. The main advantage of LSTM over traditional RNNs is its ability to handle long-term dependencies in sequential data, making it more suitable for tasks that require the model to remember information from previous inputs.

The structure of an LSTM model is comprised of a series of memory cells, which are responsible for retaining information from the input sequence, and gates that regulate the flow of information into and out of the memory cells. The gates in an LSTM model are designed to keep important information in the memory cells and forget irrelevant information, allowing the model to learn long-term dependencies in sequential data. There are three types of gates in an LSTM model: the input gate, the output gate and the forget gate.

From memory cell which information should be discarded is determined by the forget gate. It is implemented using a sigmoid activation function and receives inputs from the previous hidden state and the current input. The forget gate computes a probability value, which determines the extent to which the current information should be forgotten. If the probability is high, the memory cell will discard more information and retain less information. If the probability is low, the memory cell will retain more information and discard less information.

The information stored in the memory cell is regulated by the input gate. It is also implemented using a sigmoid activation function and receives inputs from the previous hidden state and the current input. The input gate computes a probability value, which determines the extent to which the current information should be stored in the memory cell. If the probability is high, the memory cell will store more information and discard less information. If the probability is low, the memory cell will store less information and discard more information.

The Forget gate is responsible for removing extra information from the Cell state. It is also implemented using a sigmoid activation function and receives inputs from the previous output and the current input vectors. The output gate computes a probability value, which determines the extent to which the information stored in the memory cell should be passed to the next hidden state. If the probability is high, the memory cell will pass more information to the next hidden state and retain less information. If the probability is low, the memory cell will pass less information to the next hidden state and retain more information.

In conclusion, LSTM is a powerful deep learning model for sequential data. Its ability to Handling long-term dependencies in sequential data makes it more suitable for tasks that require the model to remember information from previous inputs.

3.5 Model Used

For our model, the model used for word vectorization is Word2Vec [1, 3]. This model is known for its ability to capture semantic and syntactic relationships between words. By converting words into vectors, Word2Vec can represent words in a multi- dimensional space, where similar words are close to each other, and dissimilar words are far apart [3].

In this case, the dimensions used for the model are 100. The number of dimensions refers to the size of the vectors that are used to represent words. By using 100 dimensions, the model can capture the complexity of language while still being computationally efficient. The use of the Word2Vec model with 100 dimensions in this case provides a balance between capturing the complexity of language and being computationally efficient [3].

The LSTM model used for training has 256 units, which means it has 256 memory cells that can store information over a period. The quantity of units is an adjustable hyperparameter that is dependent on both the complexity of the task and the amount of data that is accessible for training [2].

The activation function used in this LSTM model is the Sigmoid function. The activation function is a mathematical operation that maps the inputs to outputs and helps the model make predictions. The Sigmoid function is a common activation function used in deep learning models, especially in binary classification problems like fake news detection, where the output is either 0 (Fake News) or 1 (Real News). The Sigmoid function maps the input in the range between 0 and 1, which represents the probability of a certain class. In this case, the model outputs a probability of the given news being fake or real, and the label with the highest probability is considered as the prediction [2].

The optimizer used is Adam. The Adam optimizer is often used in training Long Short-Term Memory (LSTM) networks, as it is well-suited to optimize the complex and non-linear dynamics of these models [2]. LSTMs require accurate and stable weight updates to effectively capture long-term dependencies and model sequential data. The Adam optimizer's ability to dynamically adapt the learning rate for each parameter helps to stabilize the optimization process and prevent oscillations in the training process, leading to faster convergence and improved performance. Additionally, the Adam optimizer's relatively low memory requirements compared to other optimization algorithms make it well-suited for large-scale LSTM networks [2].

Loss function used is Binary cross-entropy. The binary cross-entropy loss function is used for. binary classification problems and measures the difference between the predicted and true probability distributions. The true distribution has a value of 1 for the positive class and 0 for the negative class. The goal of this loss function is to minimize the dissimilarity between the predicted and true distributions to improve the accuracy of the model's predictions. The binary cross-entropy loss function is defined as:

$$Loss = -(y * log(p) + (1 - y) * log(1 - p))$$

where y is the true label and p is the predicted probability of the positive class. The goal of the optimization process is to minimize this loss, resulting in improved model predictions. The binary cross-entropy loss is sensitive to imbalanced class distributions, so it's important to consider the class distribution when using this loss function.

4. Results and Conclusion

The outcome of this project to find the best performing model for fake news detection. We monitor our model's performance during different epochs. Results are noted for all epochs with different batch sizes and found the highest accuracy of all Figure 1. Final accuracy achieved was 98.14 % with 6 epochs Table 1.

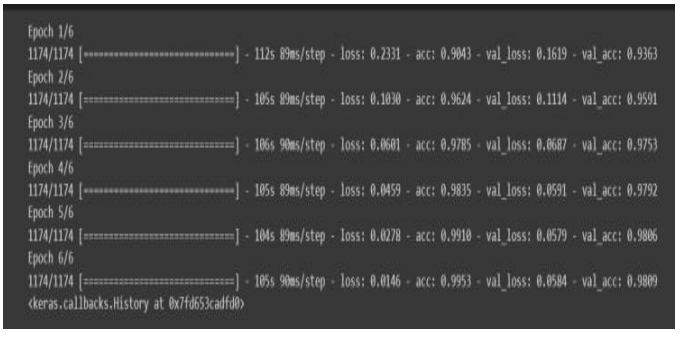

Figure 2. Model's Result after every epoque

Tabel 2. Classification Report

	Precision	Recall	F1- score	support
0	0.98	0.98	0.98	8672
1	0.98	0.98	0.98	9213
Accuracy			0.98	17885
Macro accuracy	0.98	0.98	0.98	17885
Weighted average	0.98	0.98	0.98	17885

This paper represents model to detect fake news using LSTM with the help of text classification problem. We started studying the linguistic feature-based learning model for fake news detection, this study used two political news datasets and divided the models into three, with Model 3 having the best result of 86% average accuracy at epoch size 500and batch sizes 5 and 10 [1]. This model used all lingual features. (20 features), while Model 1 used syntax, grammatical, and sentiment features (13 features) and Model 2 used readability features. LSTMs are more effective than traditional RNNs in handling long-term dependencies and modelling complex sequential data. The model prepared for this research paper detect fake news using a deep learning model based on LSTM. A dataset of news articles and tweets was used to detect fake news. The dataset has 4 columns: unnamed, title, text, and label. The label column contains binary numbers, 0 for fake news and 1 for real news. In the pre-processing stage, special symbols were removed, and the title and text columns were merged. Null lines and extra spaces were also removed. Word vectorization was used to convert words into numerical vectors for training, using the Word2Vec method with 100 dimensions.

5. Conclusion

The training was done using a LSTM deep learning model, which is capable of learning long-term dependencies for sequence predictions. The LSTM model had 256 units and used a sigmoid activation function. The LSTM model was trained for 6 epochs with sigmoid activation function and 256 units, achieving an accuracy of 98.14%. The results of the project are shown in a matrix and a screenshot of the working project is also presented. The system correctly classified a real news article about a recent incident in Uttarakhand.

References

[1] S. Bauskar, V. Badole, P. Jain, M. Chawla, Natural Language Processing based Hybrid Model for Detecting Fake News Using Content-Based Features and Social Features, International Journal of Information Engineering and Electronic Business,11(4), (2019) 1. https://doi.org/10.5815/ijieeb.2019.04.01

[2] A. Bondielli, F. Marcelloni, A survey on fake news and rumour detection techniques, Information Sciences, 497, (2019) 38-55. https://doi.org/10.1016/j.ins.2019.05.035

[3] S. A. Alkhodair, S.H. Ding, B.C. Fung, J. Liu, Detecting breaking news rumors a. of emerging topics in social media, Information Processing & Management, 57(2), (2020) 102018. https://doi.org/10.1016/j.ipm.2019.02.016

Funding

No funding was received for conducting this study.

Conflict of interest

The Authors have no conflicts of interest to declare that they are relevant to the content of this article.

About The License