A Natural Language Processing System for Truth Detection and Text Summarization

Rohith H P [1]
Department of Information Science and Engineering
Nitte Meenakshi Institute of Technology
Bangalore, India
rohith.hp@nmit.ac.in

Kavitha Sooda[2]
Department of Computer Science and Engineering, B.M.S. College of Engineering
Bangalore, India
kavithas.cse@bmsce.ac.in

Karunakara Rai B[3]
Department of Electronics & Communication Engineering
Nitte Meenakshi Institute of Technology
Bangalore, India
karunakara.rai@nmit.ac.in

Srinivas D B[4]
Department of Information Science and Engineering
Nitte Meenakshi Institute of Technology
Bangalore, India
srinivas.d.b@nmit.ac.in

2023 7th International Conference on Computing Methodologies and Communication (ICCMC) | 978-1-6654-6408-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCMC56507.2023.10083948

***Abstract*—In the futuristic world, differentiating between accurate information and false information would be challenging. To identify the fake news a truth detection system will be extremely useful. Different artificial intelligence models were compared, and the highest performing algorithm was chosen. This model was subsequently updated using a larger data-set in order to achieve better results in more real-world applications. The most essential assumptions are that the user will only utilize the subject field in which the model will work. Earlier works have implemented the process of fake news and truth detection separately from the process of summarizing the said news/information. No method was available for streamlining the process to detect whether given information was true or not and to then summarizes the given information automatically. This study has compared and contrasted the differences in accuracy and overall performance of each algorithm for fake news detection and summarization of text. Using smaller dataset, LSTM cells outperformed all other models with an accuracy of 99%. For the larger dataset, the modified LSTM has worked really well with 98.6 percent accuracy.**

***Index Terms*—Artificial Intelligence, Natural Language Processing, Fake News, Text Summarizing**

I. INTRODUCTION

The Fake News epidemic has developed rapidly over the previous decade, aided by social media platforms. This fake news can be used for a variety of objectives. To increase the number of clicks and visitors to a website, some news is summoned. Perhaps even, to persuade people to support political or monetary measures. For instance, corporations and institutions' internet reputations may be influenced. Fake health news on social media is a hazard to global wellbeing. The World Health Organization (WHO) warned in February 2020 that the COVID-19 epidemic had been influenced by a huge 'infodemic,' or an enormous amount of knowledge of it credible, some of it not—making it harder for people to find supporting evidence and reliable information when they needed it. Due to the misinformation overload we saw the rise of uncertainty, dread, worry, and intolerance than in previous epidemics.

There are many methods currently being used to solve a fraction of problems as shown in table 1. These solutions include techniques such as Linguistic Modeling, Deceptive, Clustering, Predictive, Content Cues, Non-Text Cues. There are several fake news types and various methods used to tackle each of the individual types. The most reviewed types of fake news:

- Visual Based - Visual based fake news includes pictures, videos and other visual representations
- User Based - Content generated by user.
- User post Based - Content posted by the user on the web.
- Social Network Based - Social network based content, anything posted in any public or private social media websites
- Knowledge Based - Facts and other information that derived from research papers and textbooks
- Style Based - Writing style of particular author or website which delivers content
- Stance Based - The stance taken on a particular argument and the content on that stance

This article provides an innovative approach for identifying false news as well as summarizing the provided text, which employs:

- Text pre-processing: the process of preparing and analyzing text to remove stop words and special characters.
- Encoding of the text: using N-gram analysis of words and then term frequency-inverse document frequency (TF- IDF).
- Extraction of the characteristics: This enables accurate detection of bogus information. As news features, the source, author, date, and tone are conveyed by the material.
- Long Short-Term Memory (LSTM): a recurrent neural network method that enables for the categorization of novel inputs.

The rest of this paper is organized as follows. Section II contains study of relevant work and their challenges. Section III introduces the suggested approach for truth detection and text summarization systems. Section IV compares alternative algorithms, as well as the design and implementation of the suggested system. Section V summarizes the results of a preliminary experiment. Finally, section VI summarizes the results and key topics for further research.

II. LITERATURE REVIEW

In the literature, many automated detection methods for false news and misleading articles have been described. There are numerous click baits accessible on social media networks that increase the sharing and liking of content, spreading false information. A great deal of effort has gone into detecting fake information.

The authors of [1][3][4][5] introduced many detecting algorithms. The authors have listed Fake news methods for various categories of fake news as follows:

Table 1: Current methods used for Fake news detection

Fake News Type	**Fake news detection Method**					
	Linguistic Modeling	***Deceptive***	***Clustering***	***Predictive Modeling***	***Content Cues***	***Non-Text Cues***
Visual-based	No	No	No	No	No	Yes
User Base	No	No	No	Yes	Yes	Yes
User Post Based	Yes	Yes	Yes	Yes	No	Yes
Social Network Based	No	No	No	No	Yes	No
Knowledge Based	No	No	Yes	No	No	No
Style Based	Yes	No	No	Yes	Yes	No
Stance Based	No	No	No	No	No	No

According to the researchers, the precision of these models is 63 percent to 70 percent. Every post or tweet was characterized as a binary classification issue by the authors of [6]. The categorization is completely determined by the origin of the post or tweet. The authors utilized the Twitter API data sets. The ensuing algorithms were tested on information sets.

- XG Boost
- Decision trees
- Random Forest
- Neural Networks
- SVM
- Naive Bayes.

The statistics suggest that 15 percent of tweets were fraudulent, 45 percent were authentic, and the remaining messages were uncertain.

The authors of [7][8][9] offer a straightforward method for identifying bogus news that uses a naive Bayesian classifier. This method is tested using Facebook news post data and obtained an accuracy rate of 74 percent. But various other studies have gotten higher rates using different classifiers.

The authors of [10] devised a strategy for auto-text summarizing in 2017 that included Deep network and Fuzzy logic and resulted in a considerable boost in summary accuracy.

The authors of [11] offer a foundational examination of the principles, methods, and algorithms related to automated summarization. As a prelude to text summary, certain key components about text characteristics and their representations are discussed. Their study examined previous and contemporary literature regarding models and algorithms connected to automatic summarization. Initially, this work explained ML methodologies; however, further ways for doing extractive summarization and abstractive summarization utilizing graph, semantic-based, and optimization are addressed. Some of the findings highlighted in this study include: the majority of summarizing tasks addressed by scholars are domain independent. The data-sets offered are largely from the news domains and are domain-specific.

III. METHODOLOGY

A data gathering including comments and their related metadata, such as date, source, and author, is input into the proposed system. It then converts them into a feature data collection that will be used throughout the learning phase. Pre-processing is a transformation that consists of several processes, including cleaning, filtering, and encoding. The pre-processed data collection is separated into two sections: training and testing. The training module provides a decision model that can be applied to the test data-set using the training data-set and the support vector machine approach. If the model is accepted, it may be kept and the training process can be completed. The suggested system's general scheme is depicted in Figure 1.

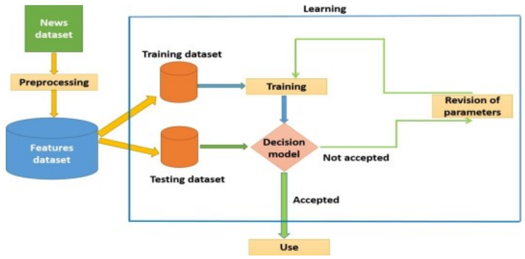

Figure 1: General Scheme of suggested system

A. *Preprocessing Data*

Data cleaning is subdivided into four categories: Removal of HTML Contents As most large datasets are scraped from the web, extraction of the relevant information from the HTML5 specifications becomes a very handy tool. These parsing algorithms usually have two stages: tokenization and tree construction.

- A Removal of Punctuation and Special Characters Upper case letter and converter to lowercase letters to maintain uniformity. Punctuation marks are removed as it would interfere with the categorizing of words.
- Removal of Stop words like “the”, ‘a”, “is” etc, don’t offer much insight so they are removed to make the data set smaller. This helps lower total computations later, which improves efficiency.
- Lemmatization is to correct the use of lexis and structural analysis of words this allows similarity across different word variations. This similarity is subsequently used by the learning algorithms.

B. Modeling

- Train test split: For the learning process, the dataset must be divided into training and testing sets. The training set is the relevant information from which the model learns, while the testing set is used to demonstrate the trained model's accuracy.
- Tokenization: Words in the individual articles are converted into tokens. Tokens are number representations of words. To keep the computations requirements low, 300 words per article is allocated. Articles with more than 300 words are truncated and articles will less are padded.
- The different models to figure out which has the highest accuracy are as follows:
  - Logistic Regression
  - Naive Bayes
  - Decision Tree
  - Random Forest
  - Boosting Ensemble Classifiers
  - Long Short-Term Memory (LSTM)
- Parameter tuning of the best model: This process attempts to increase the model's accuracy by adjusting the parameters of the chosen algorithm.

IV. Implementation

The following learning algorithms were considered for evaluating the performance of fake news detection classifiers. Seven distinct models are developed and validated as displayed in the categorization report.

A. Logistic Regression

The text is categorized using a large feature set that must yield a binary outcome, i.e., true/false or true/false article. As a result, subsequently a logistic regression (LR) model provides a simple equation for categorizing issues into binary or multiple groups.

B. Naive Bayes

One of the supervised machine learning algorithms is the Naive Bayes classifier that leverages Bayes' theorem. The variables used to construct the model are unrelated. This classifier has been shown to deliver good results on its own. Naive Bayes is a well-known approach for determining if news is true or false using multinomial Nave Bayes. Because there are various methods focusing on the same approach, it is not the only way for training these classifiers. To decide if the news is real or false, the naive Bayes approach might be utilized.

C. Decision Tree

Among the different classification methods, the Decision Tree (DT) is one of the most extensively used classifiers. The decision tree classifier is both a supervised learning method and an extremely durable classifier. Like support vector machines, decision tree classifiers may do classification and regression. All of the alternatives to a choice are visually portrayed. It is simple to comprehend since it classifies data through tree analysis. It is a decision-making tool that uses decisions in the form of a tree-like model and their potential consequences, such as random event possibilities, overhead charges, and utility. It is one method for demonstrating a conditional control algorithm.

D. Random Forest

Another supervised learning paradigm is random forest (RF), which is a more complex variant of DT. To anticipate the outcome of a class, RF is considered as the functionality which is based on massive number of decision trees. The final prediction by the algorithm is based on the class that received the most votes. When compared to other models, the random forest has a low error rate because to the little connectivity between trees. [12]. Our RF model was trained utilizing various parameters to obtain high accuracy. There are several strategies for deciding where to divide a decision tree depending on a regression or classification issue. In order to predict a split in the dataset for the classification model, the Gini index is used as a gradient descent. The Gini index (G_{ind}) is calculated using the following formula [13].

$$G_{ind} = 1 - \overset{c}{\underset{i=1}{}} (P_i)^2 \tag{2}$$

E. Boosting Ensemble Classifiers

Another well-known ensemble method for transforming weak models into strong learners is boosting. For this aim, a forest of randomized trees is leveled up. The majority resulting from each tree led to the final conclusion. This strategy enables weak learners to accurately identify data items that are typically misclassified in an incremental manner. All data points are categorized to a specific problem with equal weighted coefficient. In subsequent rounds, the weighted coefficients are reduced for properly categorized data points and increased for incorrectly classified data points [14]. Each succeeding tree constructed in each round learns to eliminate prior round mistakes and boost overall accuracy by properly identifying data points that were mis- classified in earlier rounds. One significant issue with boosting ensemble is that it may overfit to the training data, resulting in inaccurate predictions for unknown events [15][16][17]. There are several boosting algorithms available that may be utilized for classification and regression. For classification, we used the XGBoost and Cat Boost algorithms in our experiments.

F. Long Short-Term Memory (LSTM)

LSTM networks are a kind of Recurrent Neural Network (RNN) that is capable of investigating long-term dependencies [18][19][20]The hidden layer of a basic RNN is replaced with an LSTM cell in LSTM-RNN, as shown in Figure 2. Individual hidden drives in LSTMs have a natural tendency to recall inputs for a long period. A memory cell, also known as an accumulator, is a triggered leaky neuron that has a connection in the following stages with a weight of one. That is, it duplicates its true state and adds an external signal, but this link is multiply coded by another unit that determines when to remove data from memory.

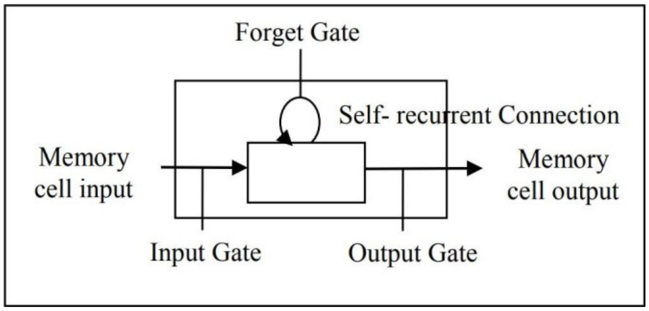

Figure 2: LSTM Cell Structure

V. RESULTS

A. Truth Detection

According to the aforementioned data, the LSTM Model has the highest accuracy of 99.9% in a smaller dataset among the various models. Hence, it is chosen as the final model for generating predictions using final testing data. The least accurate model was the Naive bayes model with an accuracy of 67.4%, this makes logical sense as it does not consider the connections between words in different paragraphs. Logistic Regression only barely performed better Naive Bayes with an accuracy of 72.01%. Cat Boost and XGBoost performed similarly well with an accuracy of 98.7%. Random Forest gave a slightly better efficiency compared to Decision Tree. This also makes sense as the Decision Tree is a single tree and Random Forest is the aggregate of multiple decision trees. This is represented in the graph in Figure 3.

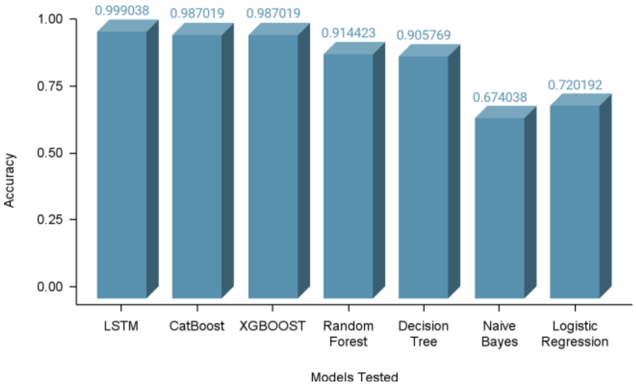

Figure 3: Comparison of different Models

After the comparison of models, we picked LSTM cell to be our primary model. The final model layer after hyper-parameter tuning and layer adjustment is shown in Table 2.

Table 2: Fake news detection Model Specifics

Layer	Output	Parameter
embedding 2 (Embedding)	(None, 300, 100)	1000000
lstm 4 (LSTM)	(None, 300, 128)	117248
lstm 5 (LSTM)	(None, 64)	49408
dense 4 (Dense)	(None, 32)	2080

Total parameters: 1,168,769
Trainable parameters: 168,769
Non-trainable parameters: 1,000,000

B. Summarization

We have also used LSTM model structure for the summarization of text. A special type of LSTM model is used called Encoder Decoder LSTM cell. The high level design of this is depicted in Figure 4.

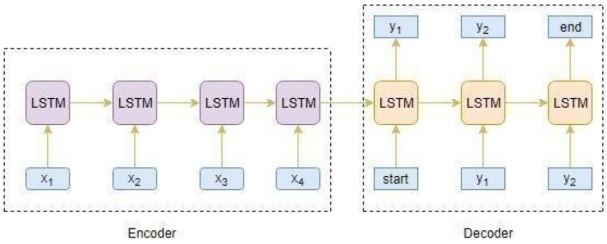

Figure 4: High Level Design of Summarization technique

The final layers for our Summarization model are given in Table 3. The first layer is a simple input layer after which is a combination of embedding and LSTM layers give the output to another input layer. From here the values are passed through an LSTM layer then an embedding layer then another LSTM layer. This final output is passed to the last LSTM layer and the summarized text is finally presented to the user.

Table 3: Text summarization Model Specifics

Layer (type)	Output shape	Param	Connected to
input 1	[(None, 100)]	0	
embedding	(None, 100, 200)	6682400	input 1[0][0]
lstm	[None, 100, 300]	601200	embedding[0][0]
input 2	[(None, None)]	0	
lstm 1	(None, 100, 300)	721200	lstm[0][0]
embedding 1	(None, None, 200)	2316200	input 2[0][0]
lstm 2	(None, 100, 300)	721200	lstm 1[0][0]
lstm 3	(None, None, 300)	601200	embedding 1[0][0] lstm 1[0][1]
			lstm 2[0][2]
time distributed	(None, None, 11581)	3485881	lstm 3[0][0]

Total params: 15,129,281
Trainable params: 15,129,281
Non-trainable params: 0

VI. CONCLUSION

Neural network design can replace traditional data in every industry due to the high availability of data. After comparing all the available high-end ML models, LSTM has performed best giving an accuracy of 0.99. LSTM cell has been modified for the purposes of truth detection on a larger dataset, yielding an accuracy of 0.986. The default LSTM model gave an accuracy of 64 percent for a dataset with 40000 entries with fake news and real news. By modifying or setting the parameters of the LSTM algorithm, this procedure tries to improve the model's accuracy to 98.6 percent.

In the proposed model, the embedding feature vector value = 40, which is the target feature vector for the embedding layer. A single LSTM Layer with 100 nodes is employed. Since this is a binary classification challenge, a dense layer with one neuron and a sigmoid function is utilized. The dropout approach is employed to minimize overfitting, and the Adam optimizer is applied to tune the loss function.

The current truth detection system detects truth only in a trained subject matter. To provide insight in another subject matter we need to train another model. To streamline this process it is better to add more subject matters to the overall corpus. The current model of summarization is limited to two-line extractive. Future work can include adding more embedding layers in the LSTM model to give a detailed summary.

REFERENCES

[1] Parikh, S. B., Atrey, P. K. (2018, April). Media-Rich Fake News Detection: A Survey. In 2018 IEEE Conference on Multimedia InformationProcessing and Retrieval (MIPR) (pp. 436-441). IEEE.

[2] Manzoor, S.I. and Singla, J., 2019, April. Fake news detection using machine learning approaches: A systematic review. In *2019 3rd international conference on trends in electronics and informatics (ICOEI)* (pp. 230-234). IEEE.

[3] Bhogade, M., Deore, B., Sharma, A., Sonawane, O. and Singh, M., 2021. A review paper on fake news detection. *INTERNATIONAL JOURNAL*, *6*(5).

[4] Mridha, M.F., Keya, A.J., Hamid, M.A., Monowar, M.M. and Rahman, M.S., 2021. A comprehensive review on fake news detection with deep learning. *IEEE Access*, *9*, pp.156151-156170.

[5] Sabeeh, V., Zohdy, M., Mollah, A. and Al Bashaireh, R., 2020. Fake news detection on social media using deep learning and semantic knowledge sources. *International Journal of Computer Science and Information Security (IJCSIS)*, *18*(2), pp.45-68.

[6] Helmstetter, S., Paulheim, H. (2018, August). Weakly supervised learning for fake news detection on Twitter. In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 274-277). IEEE.

[7] Mykhailo Granik and Volodymyr Mesyura. Fake news detection using naive bayes classifier. In 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), pages 900–903. IEEE, 2017.

[8] Leung, K.M., 2007. Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, *2007*, pp.123-156.

[9]Murphy, K.P., 2006. Naive bayes classifiers. *University of British Columbia*, *18*(60), pp.1-8.

[10] H. A. Chopade and M. Narvekar, "Hybrid auto text summarization using deep neural network and fuzzy logic system," 2017 International Confer- ence on Inventive Computing and Informatics (ICICI), COIMBATORE, India, pp. 52-56, 2017..

[11] Janjanam, P., Reddy, C. P. (2019). Text Summarization: An Essential Study. 2019 International Conference on Computational Intelligence in Data Science (ICCIDS).

[12] T. M. Mitchell, The Discipline of Machine Learning, Carnegie Mellon University, Pittsburgh, PA, USA, 2006.

[13] B. Gregorutti, B. Michel, and P. Saint-Pierre, "Correlation and variable importance in random forests," Statistics and Computing, vol. 27, no. 3, pp. 659–678, 2017.

[14] L. Breiman, J. Friedman, R. Olshen, and C. Stone, Classification and Regression Trees, Springer, Berlin, Germany, 1984.

[15] Nusinovici, S., Tham, Y.C., Yan, M.Y.C., Ting, D.S.W., Li, J., Sabanayagam, C., Wong, T.Y. and Cheng, C.Y., 2020. Logistic regression was as good as machine learning for predicting major chronic diseases. *Journal of clinical epidemiology*, *122*, pp.56-69.

[16] Song, X., Liu, X., Liu, F. and Wang, C., 2021. Comparison of machine learning and logistic regression models in predicting acute kidney injury: A systematic review and meta-analysis. *International journal of medical informatics*, *151*, p.104484.

[17] R. E. Schapire, "A brief introduction to boosting," IJCAI, vol. 99, pp. 1401–1406, 999.

[18] E. M. Dos Santos, R. Sabourin, and P. Maupin, "Overfitting cautious selection of classifier ensembles with genetic algorithms," Information Fusion, vol. 10, no. 2, pp. 150–162, 2009.

[19] Kumar, Jitendra, Rimsha Goomer, and Ashutosh Kumar Singh. "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters." Procedia Computer Science 125 (2018): 676-682.

[20] Yu, Y., Si, X., Hu, C. and Zhang, J., 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, *31*(7), pp.1235-1270.