2022 Second International Conference on Interdisciplinary Cyber Physical Systems (ICPS) | 978-1-6654-7022-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPS55917.2022.00037

Clickbait Detection Using Long short-term memory

Aromal A Balan, Anoop P
PG Scholar Department of Computer Science & IT
Amrita School of Arts and Sciences Kochi
Amrita Viswa Vidyapeetham
India
E-mail: aromalbalan18@gmail.com
anoop74p@gmail.com

AS Mahesh
Assistant Professor Department of Computer Science & IT
Amrita School of Arts and Sciences, Kochi
Amrita Vishwa Vidyapeetham
India
E-mail: asmahesh@asas.kh.amrita.edu

***Abstract*— The exploitation of clickbait has lately risen on many social media sites. Click bait is catchy titles or headlines with the primary goal of attracting attention and encouraging visitors to "click" on a headline. Media Clickbait is widely utilized, and detecting it is a critical step. This research presents a technique for detecting clickbait headlines on social media that employs a deep learning algorithm, especially a form of Recurrent Neural Network known as Long short-term memory. The method used focuses on textual characteristics, takes word sequence context into account, and derives colloquial expressions from the complete dataset. The headlines are vectorized using Word2vec Word embedding. Our conclusions were quite accurate, with a 96 percent accuracy rate, this is significantly more than conventional Machine Learning algorithms. A comparison utilizing the Naive Bayes classifier, a classification technique, was also performed.**

Keywords- Clickbait detection, Deep learning, Long short-term memory, Word2vec, Naïve Bayes Classifier.

I. INTRODUCTION

Clickbait is an aphorism used in social media to express overly dramatic headlines to entice the reader to "click" on them. Simply a 'Clickbait' title is similar to fishing, except instead of catching fish, it is attempting to grab people's clicks. These deceptive headlines obstruct the online experience by luring users to low-quality material. More online content publishers are using them to gain more page views and thus more ad money without having to provide the supporting material [14]. The website that contains the title earns money from adverts when you click on it, but the material is typically of doubtful quality and veracity. Clickbait headlines are used by websites to attract as many clicks as possible, hence boosting ad income. Technology and social media are advancing at a breakneck pace, the barriers to publishing content have disappeared, and with this, at least part of traditional quality control procedures has disappeared as well. Today anyone, from anywhere, may now create and distribute content for others to read. As a result, it's getting increasingly difficult to tell the difference between fact and fiction simply by reading the headline or title. It is only possible to determine with a decent amount of accuracy if the text being given is real or not by using indications from inside the language elements of the text [2].

This is something that typical neural networks are incapable of, and it appears to be a weakness in their design. Consider the case below: You wish to categorize the many types of events that occur during a movie. It's unclear how a normal neural network might utilize previous movie events to guide future ones. Recurrent neural networks can help with this problem (RNN). They're networks with loops that keep data from being lost. The current state input in an RNN is derived from the previous state output. Traditional Recurrent Neural Networks, on the other hand, struggle with vanishing gradient descent [1]. To tackle the difficulties with regular RNNs, LSTM is introduced. Text processing challenges are more suited to LSTMs [9]. LSTM is a sort of recurrent neural network that outperforms the normal version in many tasks. They are responsible for nearly all of the most exciting recurrent neural network results.

The categorization of deceptive content titles will really be done using LSTM in this research. The lengthier context of textual headlines may be preserved and captured using LSTM. The system takes as input titles of various articles, which are then processed using LSTM. It examines the word's meaning and significance to nearby words. The model's output will be a categorization of titles as either "Clickbait" or "Non-Clickbait."

II. LITERATURE REVIEW

To classify headlines as clickbait or non-clickbait in the past, a number of machine learning and deep learning algorithms were used. SVM [3], Decision Trees [5], Random Forest [4], and Convolutional Neural Network (CNN) [6] are only a few of the algorithms suggested by various experts. Only a few researchers have paid attention to the order of the phrases while detecting clickbait. Because of their small window widths, Convolutional Neural Networks miss long-term dependencies [7].

Potthast et al. [4] described a method for automatically recognizing clickbait. They extracted 2992 messages from Twitter using the clickbait corpus. With 215 features, the Random Forest classifier achieves the best result of 0.79

978-1-6654-7022-3/22/$31.00 ©2022 IEEE
DOI 10.1109/ICPS55917.2022.00037

AUC, 0.76 precision rate, and 0.76 recall rate. Chakraborty et al. [3] established a strategy for detecting and avoiding clickbait depending only on the linguistic clues collected from title content. They were able to detect clickbait with a 93 percent accuracy and prevent it with an 89 percent accuracy, demonstrating that clickbait could be effectively recognized using merely the user's information to lure him into simply clicking without utilizing the actual contents of the page. In 2018, the Clickbait Convolutional Neural Network (CBCNN) was released. The CBCNN method suggests not only comprehensive features, but also individual characteristics from the articles. The assessment's goal is to figure out precision, recall, and accuracy. The findings of the experiment reveal a great level of precision. The planned CBCNN method was limited to articles [12]. For click-bait identification, Fu, Junfeng, et al proposed utilizing an end-to-end CNN. For the classifier, they express titles as fixed-amount, real-valued, and continuous vectors. Most languages are supported by their model. Word vectors are formed by combining words and then used within the model. The CNN [16], which extracts features, utilizes these vectors as inputs. The data is classified using logistic regression.

Pujahari et al have introduced a robust Clickbait detection approach [15]. They offer a hybrid classification approach that integrates multiple characteristics, sentence structure, and clustering to recognize the differentiation between clickbait and non-clickbait publications. The titles are split into 11 categories during preliminary categorization. After then, the headlines get reclassified based on formality and syntactic coherence in sentences. At the final stage, the titles are recategorized by assembling them using a word vector resemblance t-stochastic neighbor embedding (t-SNE) technique. ML models are used to evaluate ML approaches on the dataset after these headlines have been classified. For Clickbait identification, the Lure and Similarity for Adaptive Clickbait Detection (LSACD) approach is proposed. Similarity and lure qualities are combined in the suggested LSACD model. The application of adaptive prediction is aided by this paradigm. A new Chinese clickbait dataset with over 5000 media items was created to test the validity of the proposed LSACD. Effectiveness and dependability have been demonstrated based on the findings. The proposed method, on the other hand, is limited to media and news, and detecting the authenticity or illegitimacy of Clickbait takes longer [11]. Vorakit Vorakitphan et al [8]. have implemented an efficient model for detecting clickbait. They tested a number of algorithms and discovered that a specific combination of LSTM with word embedding, and ontology data was the most effective for detecting clickbait. They used word embedding and ontology characteristics in their model and represented the detection issue as a series of classification tasks. They also used NBC-LF (Naive Bayes with Lexical Features), SVM-SF, LR-WEF (Logistic Regression with Word Embedding Features), and RF-WEF (Random Forrest with Word Embedding Features). Hypernym and Hyponym are employed. A hyponym is a word that has a more particular meaning than a more general or subordinate phrase that applies to it. A hypernym is a term having a wide meaning that belongs to a superordinate category, which includes words with more particular meanings. According to their theory, the number of hypernyms and hyponyms in a word may be used as a metric to decide if a word has a broad or deep sense under its scope. According to this finding, clickbait has a broad and general connotation rather than a specific one. Non-clickbait, on the other hand, could hint to a specific fact or detail about a serious topic

III. Research Methodology

A. Dataset

There are 32,000 headlines in all, including both clickbait and non-clickbait contents, in the dataset. In both groups, there are over 16000 headlines. 'ViralNova', 'Upworthy', 'BuzzFeed', 'Scoopwhoop', 'ViralStories', and 'Thatscoop' are among the websites that have contributed to the dataset. Many reliable information sources, like 'New York Times', 'WikiNews', and 'The Guardian,' are used to compile the relevant or non-clickbait headlines.

B. Feature Extraction

Preprocessing of the data is performed before processing the data. This includes converting the text into lowercase, removing punctuations, removing numbers**,** removing stop words (ex: THE, THAT etc.) and creating a word list of each headline.

C. Word Embedding

The characters in the text are converted into numerical values called word vectors. The classification model cannot function without the word vectors. The classification model cannot function without the word vectors. Word Vectors also could be arbitrarily initialized or pre-trained. The suggested approach makes use of Word2vec word embeddings. Word2vec is a natural language processing approach that was first introduced in 2013. Using a neural network model, the word2vec program finds word associations from a vast corpus of texts. A model such as this can find synonyms and identify new words for a phrase once it has been trained [13].

D. Tokenization of Data

It is the process of dismantling anything into smaller pieces. We conduct word tokenization because our input is already a single sentence, and we need to represent each word separately. Each word in a sentence is separated and treated as a single unit in this method.

E. LSTM Model

Humans do not start thinking all over again every second. there is an importance to the previous thoughts. Likewise, to understand a word in a sentence we need a good understanding of the previous word. traditional ANN can't focus on the previous data output. There comes the use of RNN. But it also has a drawback known as the vanishing gradient problem. LSTM is a perfect way to solve this problem and works on text in an efficient way.

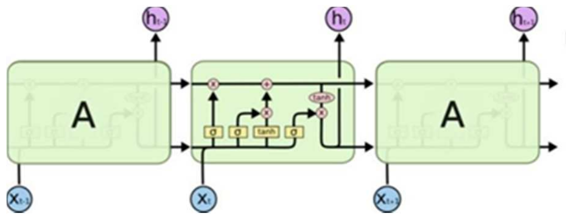

Figure 1. LSTM Architecture

LSTM basically has 3 gates and a cell state. Cell state is a long-term memory, which holds important data throughout the process. The gates are used to update the information in the status of the cell.

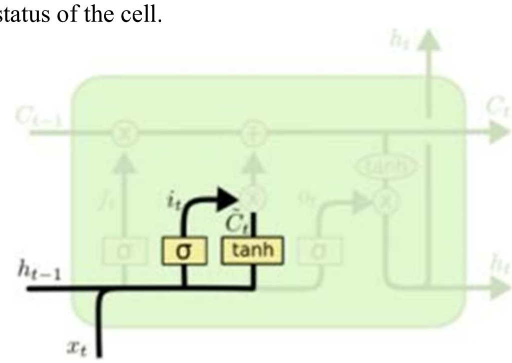

Figure 2. LSTM Input gate

The input gate finds out which data should be included in the cell state. It has two parts. The "input gate layer," a sigmoid layer, chooses which variables to update first. The state is then updated with a vector of fresh candidate values generated by a tanh layer. We'll combine these two in the following step to make a state update.

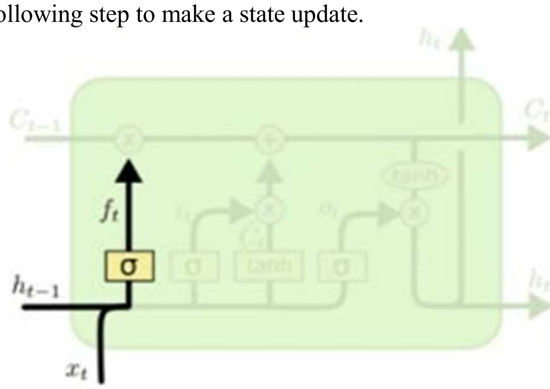

Figure 3. LSTM Forgot gate

The LSTM employs a forget gate to determine whether information from the cell state should be discarded. It uses a sigmoid function on h_{t-1} and x_t and returns a number between 0 and 1. A 1 indicates "totally keep this", whereas 0 indicates "entirely keep this".

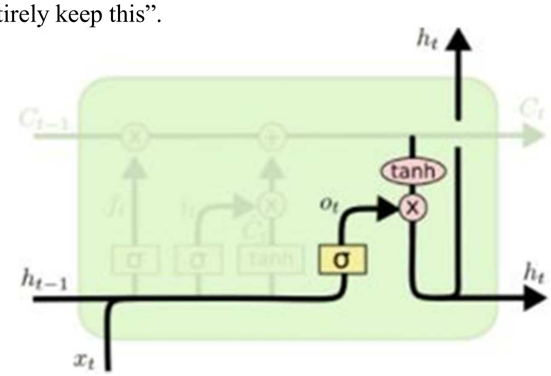

Figure 4. LSTM Output gate

The output will also be filtered and will be based on the cell condition. To begin, we use a sigmoid layer to choose whatever aspects of the cell state will be output. The cell state is then sent through tanh (to compel the values to be within 0 and 1) and multiplied by the sigmoid gate's output, yielding just the parts we want.

The dataset is split as testing and training sets. The percentage of training data is 80% and testing data is 20%. For building an efficient LSTM model different units are tuned several times. Table 1 shows the accuracy of LSTM networks with different LSTM units. By back propagating, LSTM units assist in lowering the error rate.

TABLE I. LSTM DETAILS

Parameter	Value
LSTM	50
Dropout	0.2
Activation function	Sigmoid
Epochs	3

We experimented with different epoch sizes and LSTM units. However, in case of clickbait data, 50 and 50 LSTM units gave the best accuracy. The accuracy of the system decreased when the number of LSTM units was increased. The most successful strategy for regularization is dropout, which skips a certain percentage of neurons throughout the training process [10]. This helps to reduce the chances of overfitting. We also experimented with the Word2vec size to increase the model's accuracy; Table 2 shows the accuracy of models with different Word2vec sizes. The best accuracy is achieved when the Word2vec dimension was 200.

TABLE II. WORD2VEC DIMENSION

Word2vec dimension	Accuracy
50	95.612
100	96.117
150	96.062
200	96.270
250	95.675
300	95.750

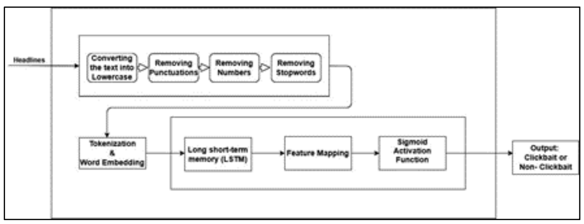

Figure 5. System Flow Diagram

F. *Naïve Bayes Classifier*

The Naive Bayes method is a supervised learning technique that uses the Bayes theorem to solve classification issues. It's typically used in activities that need a large training dataset, such as text categorization. The Naive Bayes classifier is a basic and effective classification method for quickly creating machine learning models that can make accurate predictions. It's a probabilistic classifier, which means it predicts items based on their likelihood. sentiment analysis, Spam filtration and article classification are all instances of the Naive Bayes Algorithm.

IV. RESULTS

Using the LSTM Model, the system seeks to contextually classify clickbait headlines. Word2vec word embedding is used in LSTM. TF-IDF (Term Frequency-Inverse Data Frequency) word embedding is used with Naïve Bayes Classifier. Metrics including F1 score, recall, accuracy, and precision are calculated to gauge the outcomes. The results achieved are shown in Table 3.

TABLE III. RESULTS

	Accuracy	Precision	Recall	F1
NB	0.94	0.94	0.94	0.95
LSTM	0.96	0.96	0.96	0.96

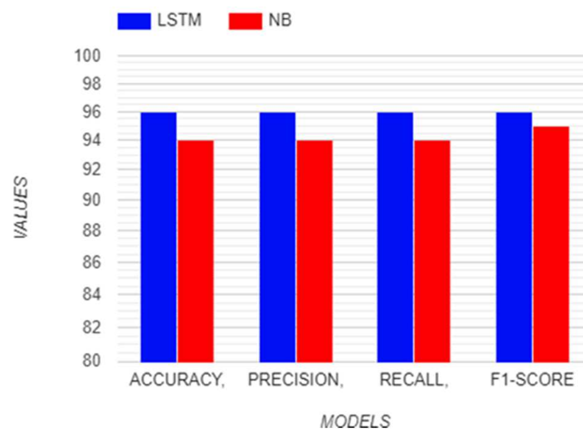

Figure 6. Result Comparison

On each of the metrics considered, our suggested model outperforms the best results by a significant margin. It receives an F1 score of 0.96 and an Accuracy score of 0.96. This out-turn result demonstrates the efficacy of the LSTM model we deployed, which automatically infers valuable information from the final job.

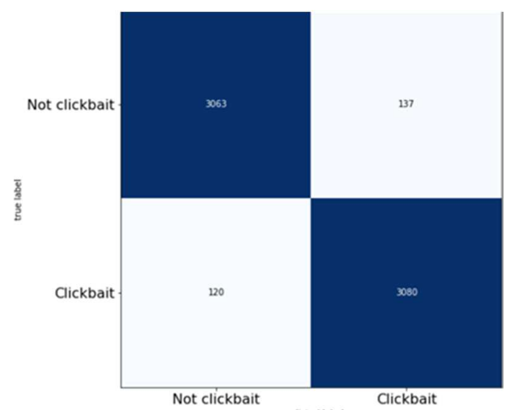

Figure 7. LSTM Confusion Matrix

The Naïve Bayes model performed well, but it's less efficient when compared with the LSTM model. Naïve Bayes Model got an f1- score 0.95 and an Accuracy score of 0.94. which is less than our LSTM model.

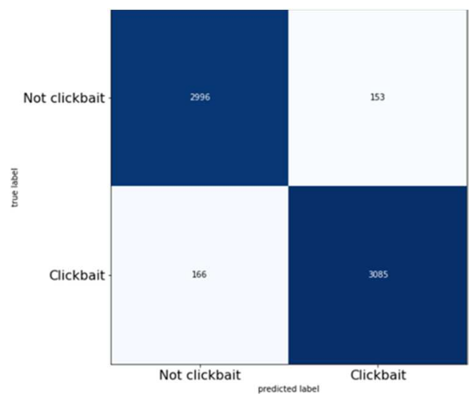

Figure 8. Naive Bayes Confusion Matrix

V. Conclusion and Future Research Directions

As we are in the information era, trustworthy information is the most important thing. Each data has its own power to provide fake or/and trustworthy information to the public. In particular, Clickbait is more of a nuisance to the public as they are manipulating the “fact.” Our proposed model uses the LSTM and word2vec algorithm to find out the clickbait in an efficient way. The LSTM deep learning algorithm is used in our suggested system, and the Naïve Bayes is used to compare results depending on many aspects of the material. We think that by adding a few tweaks to the model, we can significantly improve the system's performance.

In contrast to machine learning classifiers, the suggested technique attained the greatest accuracy of 96.8% using LSTM. This approach may be used to segregate clickbait headlines from non-clickbait headlines.

For the future, we plan to:(1) collect more data to create better models. (2) Create a backup server with something like an internet browser plug-in that takes advantage of the model's capabilities and warns people about Clickbait. (3) Using the most updated word embedding techniques depending on the development of the norm.

Acknowledgement

We would like to express our appreciation to the guide for sharing his pearls of wisdom. We are also appreciative for the feedback provided by our reviewers and the institution.

References

[1] Https;//colah.gihub.io/posts/2015-08-Understanding-LSTMs/

[2] Naeem, B., Khan, A., Beg, M.O. and Mujtaba, H., 2020. A deep learning framework for clickbait detection on social area network using natural language cues. *Journal of Computational Social Science*, *3*(1), pp.231-243.

[3] Chakraborty, Abhijnan, et al. "Stop clickbait: Detecting and preventing clickbaits in online news media." *2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*. IEEE, 2016.

[4] Potthast, M., Köpsel, S., Stein, B., & Hagen, M. (2016, March). Clickbait detection. In *European Conference on Information Retrieval* (pp. 810-817). Springer, Cham.

[5] Biyani, P., Tsioutsiouliklis, K., & Blackmer, J. (2016, February). " 8 amazing secrets for getting more clicks": detecting clickbaits in news streams using article informality. In *Thirtieth AAAI conference on artificial intelligence*.

[6] Fu, J., Liang, L., Zhou, X., & Zheng, J. (2017, July). A convolutional neural network for clickbait detection. In *2017 4th International Conference on Information Science and Control Engineering (ICISCE)* (pp. 6-10). IEEE.

[7] Chawda, Sarjak, et al. "A novel approach for clickbait detection." *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2019.

[8] vorakitphan, Vorakit, Fang-yie leu, and Yoa-Chung Fan.” clickbait detection based on word embedding models.” International Conference on Innovative mobile and internet Service in Ubiquitous Computing. Springer, Cham, 2018.

[9] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[10] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.

[11] Zheng, Jiaming, Ke Yu, and Xiaofei Wu. "A deep model based on Lure and Similarity for Adaptive Clickbait Detection." *Knowledge-Based Systems* 214 (2021): 106714.

[12] Zheng, Hai-Tao, et al. "Clickbait convolutional neural network." *Symmetry* 10.5 (2018): 138.

[13] Lei, Shi. "Research on the Improved Word2Vec Optimization Strategy Based on Statistical Language Model." *2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*. IEEE, 2020.

[14] Scott, Kate. "You won't believe what's in this paper! Clickbait, relevance and the curiosity gap." *Journal of pragmatics* 175 (2021): 53-66.

[15] Pujahari, Abinash, and Dilip Singh Sisodia. "Clickbait detection using multiple categorisation techniques." *Journal of Information Science* 47.1 (2021): 118-128.

[16] Fu, Junfeng, et al. "A convolutional neural network for clickbait detection." *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2017.