How to Cite:
Rautela, J. ., Ramalingam, V. V., & Makhdoomi, H. (2022). Fake news detection through deep learning techniques. *International Journal of Health Sciences, 6*(S5), 2107–2111.
https://doi.org/10.53730/ijhs.v6nS5.9091

Fake news detection through deep learning techniques

Jay Rautela
Department of Computing Technologies, SRMIST, Chennai, India
Email: jr1535@srmist.edu.in

Ramalingam.V.V,
Department of Computing Technologies, SRMIST, Chennai, India
Corresponding author email: ramalinv@srmist.edu.in

Hamaad Makhdoomi
Department of Computing Technologies, SRMIST, Chennai, India
Email: hm3014@srmist.edu.in

***Abstract*---**This paper aims to predict whether a given news article is real or fake. We use the dataset available on Kaggle. We then implement several deep learning models (Long short-term Memory (LSTM), Multi-layerPerceptron (MLP), Convolution Neural Networks (CNN), Hybrid CNN-LSTM on this dataset. For these models we examine the effects of character-based vs. word-based models and pretrained embeddings vs. learned embeddings. We also compare the accuracies of various models and report the best accuracy which we would get from a particular model.

***Keywords*---**Long short-term Memory (LSTM), Convolution neural networks (CNN), Support vector machines (SVM), Logistic Regression.

Introduction

Recent trends and advancements in communication and mobile technology, as well as the widespread use of the Internet, have made it easier to access news from around the world. Furthermore, theweb-based life applications are developing at an unregulated rate, andthe competition between the US and China to share and spread news-related data. In the topic of social correspondence, there are a number of worldwide organisations which have influenced the news trustworthiness. Media has evolved into one of the most important providers of

information. Theterm "fake news" indicates purposely disseminating false narratives.

On Digital media, false information is spread to confuse and perplex people. To deceive the reader in order to gain economic or political goals. Moreover, the industry's diversified and rising number of participants in the sphere of news writing and dissemination, this has resulted in the creation ofNews articles about which it's impossible to know whether or not they're trustworthyor not. Fake news has exploded in popularity on social media.Academic and industry researchers who are highly drivendomains in order to find solutions to this problem.

Before the elections in United States, there was a lot of bogus news.The presidential election of 2016 is a contentious topic.This had an impact on popular opinion.The dangerof a catastrophe, the ramifications of the quick dissemination of incorrect information on social media, the number of network sites is rapidly expanding. Detecting bogus news automatically is considered a difficult task.

Challenges

Detecting fake news articleswith the help of Natural Language Processing is a more tangled task than simple articleexamination. Further-more, indications of fake news articles in social media are often subtle, and accordingly not promptly clear to the human peruse. These unpretentious signs, notwithstanding, might be reflected in the subtleties of somebody's language, which we anticipate can be caught by an assortment of profound Deep learning strategies. By utilizing a wide arrangement of models, alongside cutting edge NLP techniques, we desire to make a strongclassifier that is sensitive to the varieties of fake articles on an individual basis.

Problem Statement

Our project aims to do the following:

- To Generate labelledarticles from Scratch in the dataset.
- Construct, neural-based architecture (CNN, LSTM, MLP).
- Tune model parameters to optimize Performance.
- Analyze performance of character-based vs. word-based models.

Collecting Dataset

To collect potentially negative examples i.e fake articles, wesearched for pages on Twitter that had some-thing to do with political agenda; specifically, we would require that the word which was some place in the client's name or title of the page. Models like "Sorrow Quotes", "Melancholy Notes",and "Damn Depression" were all helpful pages in our dataset assortment. We tried not to deletearticles from pagesthat wereoutfitted towards wretchedness mindfulness or those pages oversaw by associations that are attempting to assist individuals with agenda. Part of these pages were not claimed by clients who were conceivably discouraged. Subsequent to investigating the substance of the pages, we had shortlisted as "substantial" pages,compiledthem into a list of potentially positive articles.

For potentially negative examples, we pulled examples from a wide range of pages (the news, sports, dance teams, etc.); we also made sure to get pages that had to do with other emotions (anger, happiness, etc.), so that our model could be able to distinguish between agenda due to fake narratives and other misleading articles.

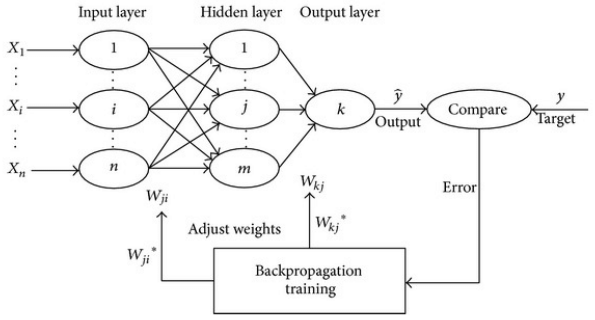
Cleaning the Dataset

Many examples in the initial dataset were not actually relevant for our use case. This is because several of the examples had something like "Here's a new YouTube post!" or "Account got suspended for a while, but now we're back!" that did not seem to indicate anything about the user, their thoughts, or their situation. We filtered our entire dataset to take out these arbitrary examples.

Sanitizing individual examples

Forremaining examples, we sanitized each tweet so that they did not contain irrelevant text, so they would be suitable input for our various models. One category of irrelevant text, for both word-based and character-based models, are hyperlinks, because they do not add much to the actual content of the tweet. On top of that, hyperlinks do not have a corresponding word embedding from Word2Vec or GLoVe, nor could we have generated useful word level embeddings for these hyper-links. Another category of irrelevant text for word-based models are hashtag's and miscellaneous symbols because similarto the hyperlinks they do not have corresponding word embeddings.

Technical Approaches



Convolution Neural Networks, LSTM, MLP

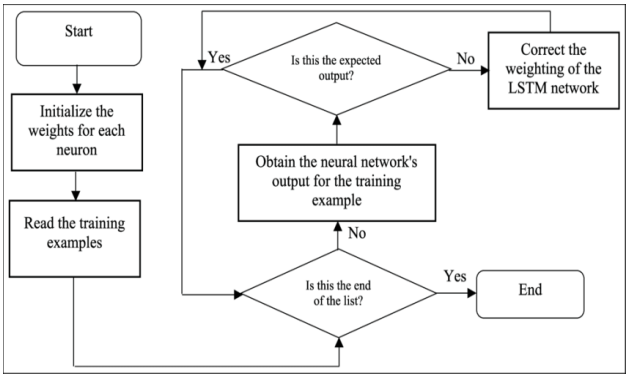
For our baseline algorithm, we used convolution neural networks with the different parametersto perform binary classification. First, for each article in the dataset we clean the data set and then different cleansing techniques are performed.Then, at that point, we get the component vector for each article by averaging all the word vectors inside the articles. At last, we apply our pattern classifiers to the assortment of article highlight vectors and to the relating names to prepare and test our benchmark models.

Hyper-parameters

For the RNN-based models (basic RNN with words) model that we built was a Multi layered Perceptron neural network and LSTM.RNNs have recently been applied to NLP tasks and have shown promising results in text classification. We process the.CSV files similarly like the word-based RNN models. A major difference between this CNN and other RNN-based models is that this model learns the embedding matrix Glove at training time, while RNN-basedmodels used pretrained embeddings

Results achieved

After training different models like CNN, LSTM, Hybrid model CNN-LSTM we got different accuracies in different models. In CNN, we got accuracy a slightly lower than other models of about 64%. But In LSTM, we got an improved accuracy of about 92%.Also in the hybrid model of cnn-lstm an accuracy of about 98% was achieved. We can say in future modelling concept of hybrid modelling should be encouraged more often.



Qualitative Analysis

Using a Glove cell performed substantially better than using a basic RNN cell when we used word embeddings. This is likely due to the fact that using an RNN cell may bias the model into paying more attention to the words that appear at the very end of the sentence, whereas the Gloveembeddings allows different parts of the sentence to play an active role in deciding the final prediction. But when the RNN cell only takes the end part of the sentence into account, it almost ignores the first part. It is likely due to this reasoning, the fake article was misclassified by the RNN cell version, but not by the Glove cell version.

Conclusion

Our model could use some improvements in future iterations. One worthwhile avenue to explore would be to use a CNN-LSTM Encoding-Decoding architecture as we had mentioned in a previous section because that had proven successful while performing article classification. In addition to this, we could also experiment with decaying learning rate to improve the stability of some of our loss

and accuracy graphs (in addition to training for a longer time).We also conclude that in such future studies hybrid modelling should also be encouraged which can help in achieving further accuracy. Also for CNN model the number of epochs should be further increased as convolution architecture needs more training data. So to conclude in our study of detecting fake news by deep learning we found that CNN model got a slightly lesser accuracy with respect to other models i.e. LSTM and hybrid model of CNN-LSTM which got 92% and 98% accuracy respectively. So NLP with LSTM can achieve a better accuracy rather than by using pre trained embeddings in Glove.

Acknowledgments

This work was done under the guidance and mentorship of our supervisor DR.V.V Ramalingam, Associate Professor at SRMIST Chennai.

References

[1] Britz D. (2015) Implementing a CNN for Text Classification in TensorFlow. WILDML. http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/

[2] KimYoon (2014) Convolutional Neural NetworksforSentenceClassification. https://arxiv.org/pdf/1408.5882.pdf

[3] Minimax char embeddings GitHub https://github.com/minimaxirichar-embeddings

[4] Santos C.N. & Gatti M. (2014) Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 6978, Dublin, Ireland, August 23-29 2014. http://www.aclweb.org/anthology/C14-1008.

[5] Library of Twitter scrapper https://github.com/taspinar/twitterscraper

[6] Foroughi S. &Vijaya Raghavan P. & Roy D. (2016) Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder.ACM. https://arxiv.org/pdf/1607.07514.pdf.