Busting Fake News in a Digital World

CS3244 | Group: 3244 - 2010 - 0010

Chua Cheng Ling (e0323077@u.nus.edu) | **Colin Ng Chenyu** (e0325455@u.nus.edu) | **Tan Wei Zhong Kevin** (e0309723@u.nus.edu)
Nguyen Thanh Duc (e0313523@u.nus.edu) | **Tran Quang Thanh** (e0313566@u.nus.edu) | **Le Hong Long** (e0313558@u.nus.edu)

Introduction

A Straits Times article in 2018 reported how even though 4 in 5 Singaporeans are confident in spotting fake news, over 90% identified at least 1 out of 5 fake headlines as real. Our objective is to develop a reliable model that can classify a news article, bringing about greater reliability and accuracy in the information that is reported.

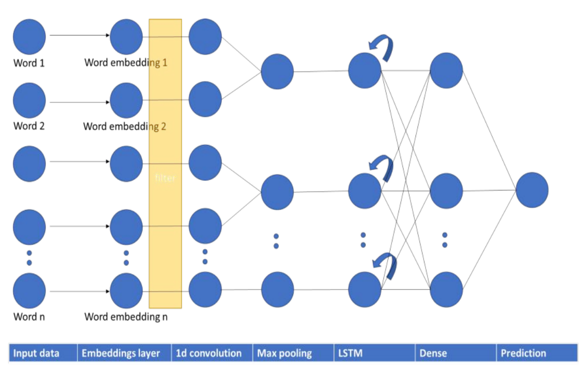
Data

- **Data Collection:** Pre-labeled data was collected from Kaggle. We ensured that there is no bias in the dataset that can potentially lead to skewed results. Overall, our training dataset has 2014 fake and 2972 real articles. The feature that we use for this project is the body (text) of the article. Each word of the article is a token.
- **Data Cleaning:** We chose to remove punctuation, except for exclamations. We chose not to remove stopwords and chose not to lemmatize and stem the text as we want our model to learn the syntactic meaning of the sentences.

Models

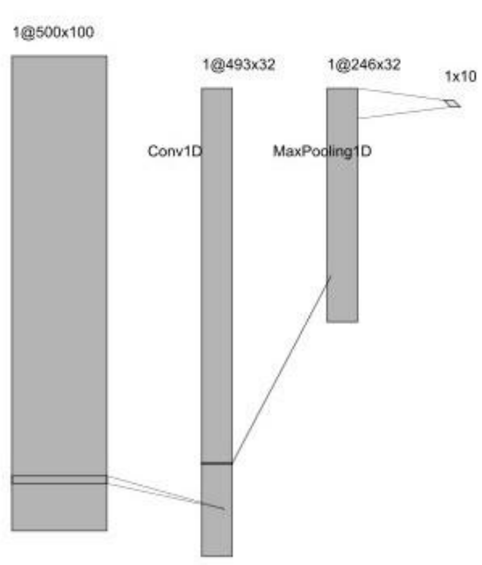
All our models used an embedding layer to convert words using GloVe embeddings into arrays of numbers.

- **Recurrent Neural Network:** The diagram on the right describes the layers of RNN. Convolution first receives the words and attempts to extract the features, before doing a max pooling where the max values in clusters are extracted putting them in **LSTM** where it remembers values over arbitrary time intervals.

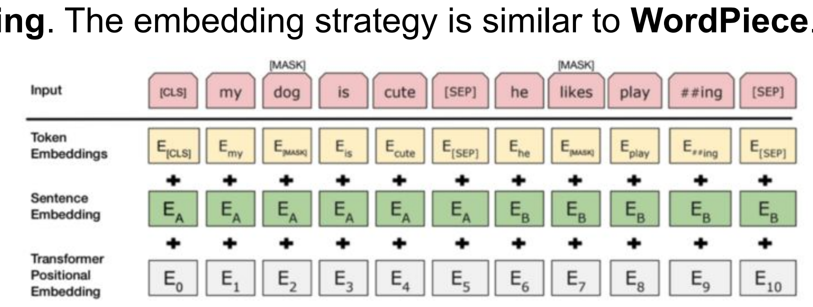


The 3 gates (input, output, forget) regulate information in and out of the cell. The second last layer has all nodes connected with each other before coming with the final prediction.

- **Convolutional Neural Network:** The diagram on the right shows the **CNN** model for Fake News Detection. After embedding, there are 100 layers of texts where each text has a size of 500. Then, a **convolution layer** having a kernel size 8 will measure relationships of nearby words before passing to a **max pooling** layer with size 2. Finally, the data is flattened and passed to a dense layer of size 10 with **ReLU** activation and another dense layer with **Sigmoid** activation for binary classification.

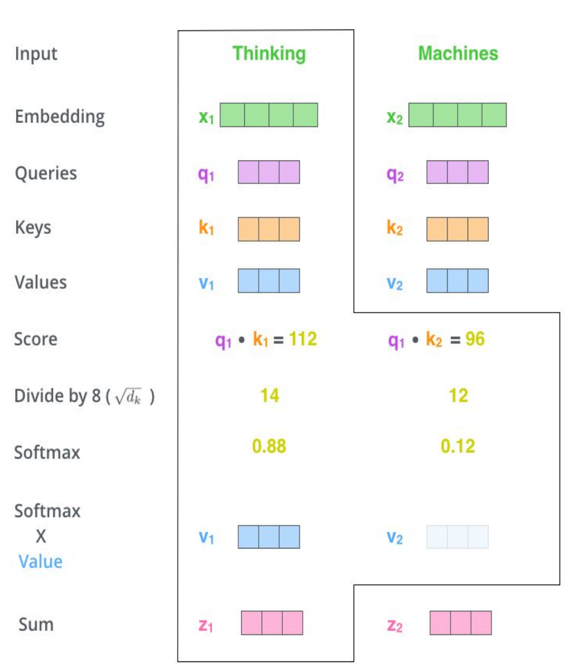


- **BERT (Transformer Neural Network):** The first diagram (below) is the embedding layer of Bert, which is the sum of **word embedding**, **token (sequence) embedding** and **position embedding**. The embedding strategy is similar to **WordPiece**.

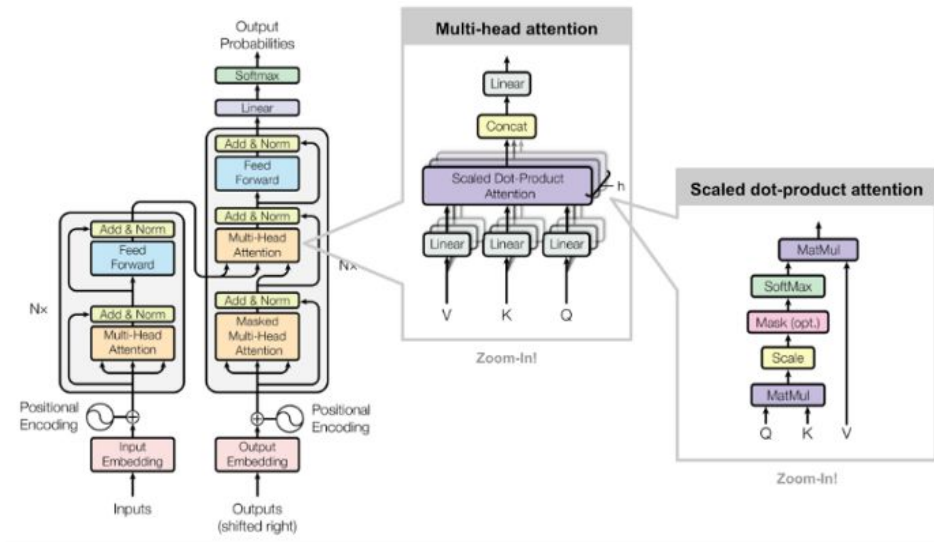


The second diagram (below) is the architecture of a single **Self-Attention** unit. It constructs 3 feature vectors **Q**, **K**, **V** from the product of the input with 3 corresponding 2D matrices W^Q, W^K, W^V. The output is then

$$Z_i = \sum_j softmax(\frac{Q_i \times (K_j^T)}{\sqrt{d}}) \times V_j$$



The third diagram (right) is the overall architecture of Bert. The last **softmax layer** output the probability for 2 classes - True and False.

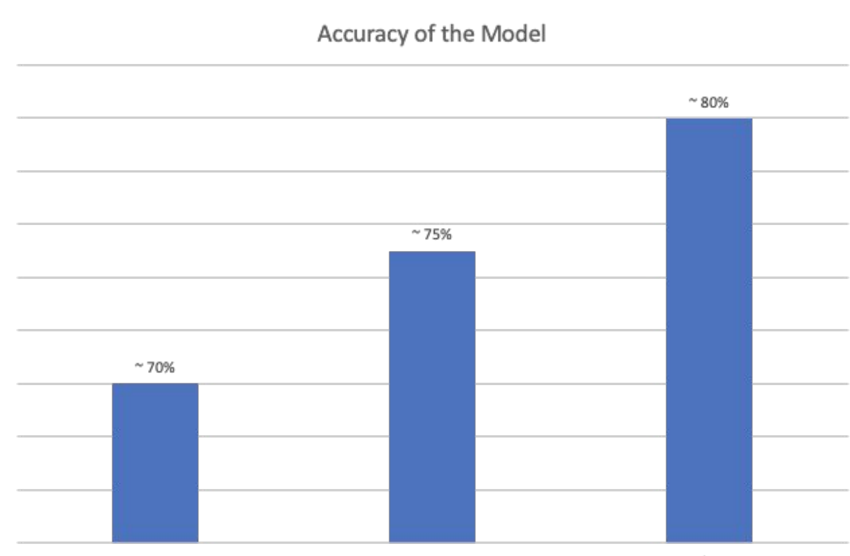


Since pre-trained BERT only have a maximum embedding size of 512, we used a **hard-voting** strategy on the classifications of output features of the **overlapping news segments** to decide on the final classification.

Besides the above models, we have also tried **Naive Bayes Classification** but we realized that naive models do not generalize well since they only treat embedded words as an array of separate numbers, but do not consider the structure of the sentences and the relationship between neighbor words.

Result & Discussion

- **Recurrent Neural Network:** We were able to reach a validation accuracy of around 77% for the validation data and an accuracy approximately **75%** for the test data within 5 epochs.
- **Convolutional Neural Network:** We were able to reach an accuracy of over 80% for the validation data and an accuracy of more than **70%** for the test data within 5 epochs.
- **BERT (Transformer Neural Network):** We were able to reach an accuracy of 100% for the validation data and an accuracy of **80%** for the test data after 1 epoch with a learning rate of 1e-6.



Currently, we are only training our model based on the text of the article. We believe that we can extend our model to include other features such as author, publishing company, etc. to improve our accuracy. However, given the limitation in time and resources, we couldn't explore further at the moment.

Conclusion and Future Work

In conclusion, we have implemented various machine learning and deep learning models to achieve our goal of detecting fake news. We managed to obtain a fairly high accuracy for majority of our models and as such are able to identify which models work best for our purposes.

We aim to extend our project to other platforms such as Facebook and Twitter, where there is a high volume of information flow on a daily basis.

Acknowledgement

Special thanks to Associate Professor Kan Min Yen, teaching assistants Zhu Tongyao, Abhinav Ramesh Kashyap and Zhang Tianyang for all their assistance in the project.