Term Project – Re-implementation of "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection - *William Yang Wang (2017)*

CAP 5619, Deep and Reinforcement Learning, Spring 2020
Department of Computer Science, Florida State University

By Uzoamaka Faith Ezeakunne

1. Introduction

The proliferation of fake news is a has led to social and political impacts. An example of such impact is the pizzagate incident. In 2016, during the presidential election of the United States, fake news was propagated stating that Hilary Clinton led a child-abuse ring and was keeping them in a pizzeria; this led to a real shooting as a man went to the pizzeria with the intention of rescuing the victims before it was realized that it was fake news.

In order to detect fake news, fact-checking websites were made by trained persons with sufficient journalism training to prove a news as real, fake or in-between. An example of fact-checking website is Politifact[1]. Politifact fact-checks news from several sources and labels them in six categories namely; true, mostly-true, half-true, mostly-false, false and pants on fire ranked from degrees of truthfulness, high to low, respectively. Wang et al [1] introduced 12.8k decade long, real world, manually labeled dataset for fake news detection. The dataset's news source is from the diverse Politifact fact-checker, accumulated using its API. Also, the hybrid-CNN was proposed as a model to detect fake news.

However, fake news propagates fast especially since the invention of social media where news can be disseminated from unauthorized news sources. Hence, the need to detect fake news automatically. In the following sections of this report, I will define the problem, dataset, model used by the author of the original paper [1], finally, I will show our result and conclude the report.

2. Problem Statement and Objective

The fake news detection problem is defined as assigning a correct credibility label (e.g. true, false) to a given piece of news (e.g. a text). In this case, our credibility labels are true, mostly-true, half-true, mostly-false, false and pants on fire and a piece of news contains text and meta-data (e.g. speaker, job).

The research questions in the original paper are: "Based on surface-level linguistic realizations only, how well can machine learning algorithms classify a short statement into a fine- grained category of fakeness? "[1] and "Can we design a deep neural network architecture to integrate speaker related meta-data with text to enhance the performance of fake news detection? "[1].

[1]http://politifact.com

The objective of this report is to re-implement a deep neural network architecture to integrate speaker related meta-data with text for fake news detection which was proposed by Wang et Al in the original paper and named the Hybrid-CNN model.

3. Dataset

Wang et Al (2017) introduced the LIAR dataset from the diverse PolitiFact fact-checker, accumulated using its API containing 12.k news. In order to verify the editor's reliability, Wang et Al (2017) went through a randomly sampled subset of the editor's analysis reports after which they obtained a Cohen kappa's value of 0.82. The dataset has been divided into a training data of 10,269 news, validation data of 1,284 news and test data of 1,283 news. The labels are true, half-true, mostly-true, barely-true, false, and pants-on-fire. The dataset has been described in Figure 1.

```
LIAR: A BENCHMARK DATASET FOR FAKE NEWS DETECTION

William Yang Wang, "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News
Detection, to appear in Proceedings of the 55th Annual Meeting of the Association for
Computational Linguistics (ACL 2017), short paper, Vancouver, BC, Canada, July 30-August 4,
ACL.
=================================================================
Description of the TSV format:

Column 1: the ID of the statement ([ID].json).
Column 2: the label.
Column 3: the statement.
Column 4: the subject(s).
Column 5: the speaker.
Column 6: the speaker's job title.
Column 7: the state info.
Column 8: the party affiliation.
Column 9-13: the total credit history count, including the current statement.
9: barely true counts.
10: false counts.
11: half true counts.
12: mostly true counts.
13: pants on fire counts.
Column 14: the context (venue / location of the speech or statement).
```

Figure 1. Description of the LIAR Dataset
(Source: https://github.com/thiagorainmaker77/liar_dataset)

[1]http://politifact.com

4. Hybrid CNN Model

The hybrid CNN model integrates text and meta-data. In this model, a matrix of embedding vectors is initialized randomly to encode meta-data embeddings. A convolutional layer is used to capture dependencies between the meta-data vectors, then max-pooling layer is attached. Next, a bi-directional LSTM layer is attached .The maxpooled text representation is then concatenated with meta-data representation and fed to fully connected layer with softmax activation function to generate output.

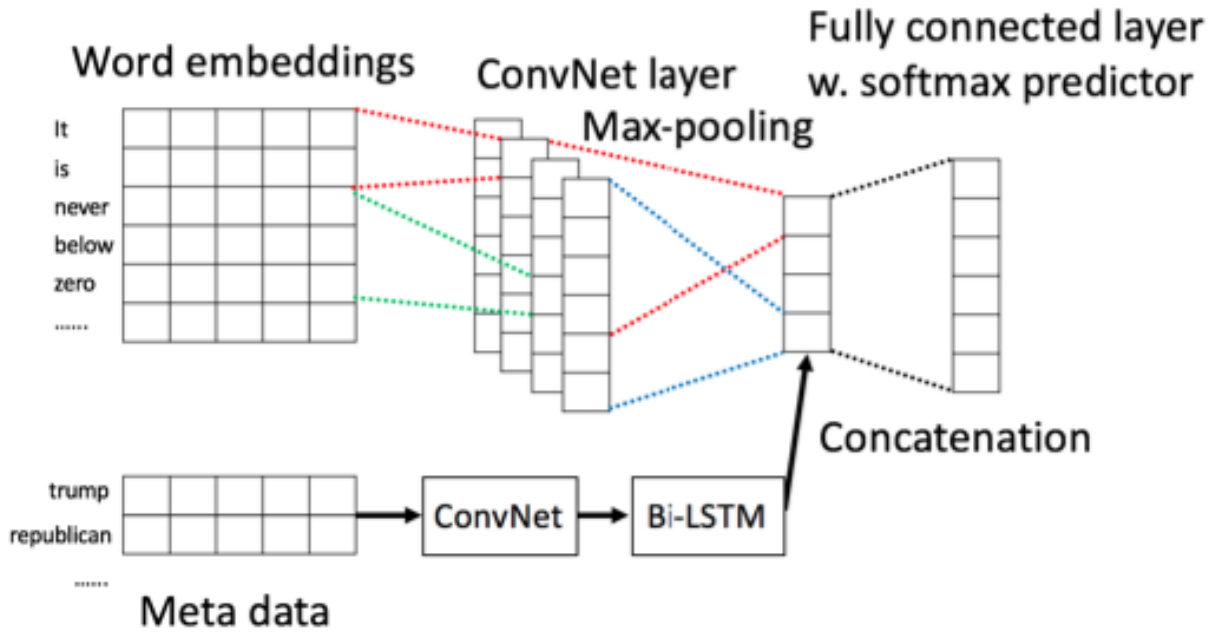

Figure 2. Hybrid CNN model [1]

5. Experiment

For my experiment, data pre-processing was done using code collected from GitHub repository [2] and inspected to fix bugs and improve compatibility with the coding environment. The data pre-processing phase involves reading and embedding the input. Text embedding were aided by pre-trained 300-dimensional word2vec embeddings.

Next, I implemented the model, tuning the hyperparameters. The unchanged hyperparameter is the dropout probabilities of 0.5 and 0.8. The filter sizes were changed to (3, 4, 5), in all cases, each size has 128 filters. The batch size was increased to 400 in order to prevent overfitting. Also, the optimizer used was Adam, finally, number of epochs was set to 10.

[1]http://politifact.com

The author chose the accuracy metric due to his observation that the accuracy results from various models were equivalent to f-measures on this balanced dataset; I will also use only accuracy metric to demonstrate the experiment. The figures below show the training and validation accuracies for the experiment.

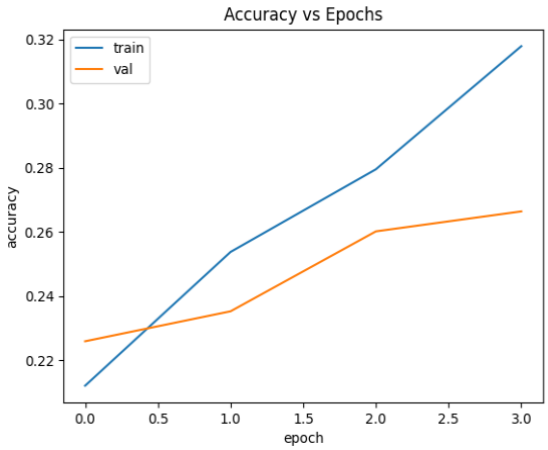

Figure 3a. Text + Subject

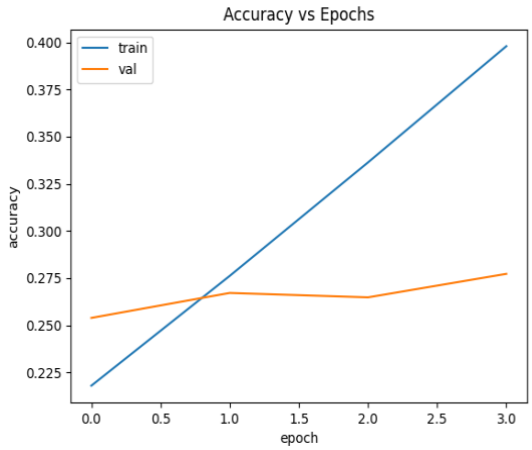

Figure 3b. Text + Speaker

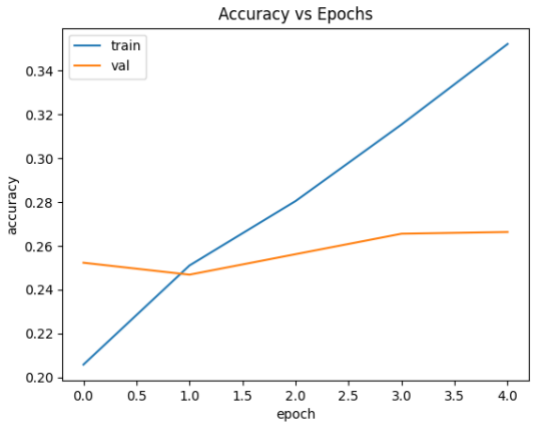

Figure 3c. Text + Job

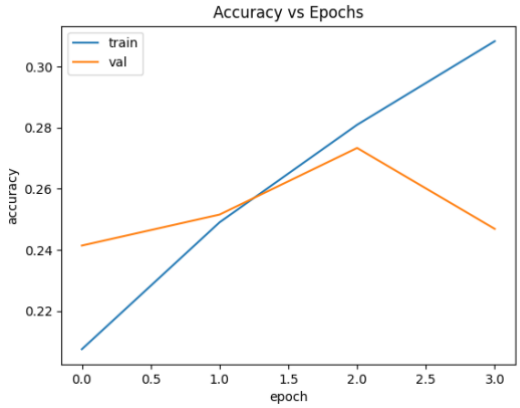

Figure 3d. Text + State

[1]http://politifact.com

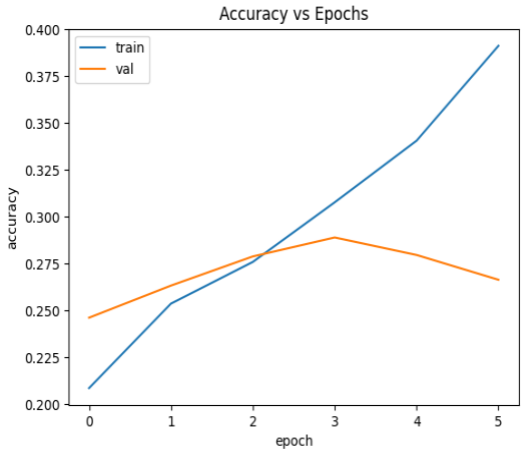

Figure 3f. Text + All

6. Experimental Results

The results of the experiments are shown in Table 1. The rows represent a combination of text and meta data, the columns represent either the original hybrid CNN or the modified Hybrid CNN with modified hyperparameters. Each cell represents the validation accuracy of the corresponding model and feature set or meta data.

	Original Hybrid CNN	*Modified Hybrid CNN*
Text + Subject	0.263	0.266
Text + Speaker	0.277	0.277
Text + Job	0.270	0.266
Text + State	0.246	0.247
Text + Party	0.259	0.258
Text + Context	0.251	0.250
Text + All	0.247	0.264
Average Validation Accuracy	**0.259**	**0.261**

Table 1. Experimental Results (validation accuracy of meta data combinations in the models)

[1]http://politifact.com

7. Conclusion

Fake news detection is an important problem in the society as it can lead to real consequences, hence the need to detect it quickly using automatic methods. This project uses a hybrid CNN model proposed by Wang et Al (2017) [1] to detect fake news. However, this model was re-implemented with hyperparameters to achieve validation accuracy of approximately by 0.8% .

The source code is available on my GitHub repository:

References

[1] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. arXiv preprint arXiv:1705.00648, 2017

[2] Jurat(Jerrat) Shayiding, Deep-Convolutional-Models-for-Fake-news-classification, (2019), GitHub repository, https://github.com/jshayiding/HybridCNN-fake-news-classifier

[1]http://politifact.com