FNC1 Revisited: Two-Step Multilayer Perceptron based Stance Detection

Manav Mehra
Department of Computer Science
University of Waterloo
m3mehra@uwaterloo.ca [*]

Rajbir Singh
Department of Computer Science
University of Waterloo
rsrajbir@uwaterloo.ca [*]

Abstract

Factual information for spreading rumors, threats, etc for purposes of political propaganda through social or print media refers to Fake News. Researchers have developed novel techniques to detect fake news and control its spread before it becomes viral. Fake News Challenge was organized in 2017 which involved predicting one of agree, disagree, discuss, or unrelated class given a headline-article pair. This task falls under Stance Detection where the relativeness of two separate textual pieces is estimated. We developed various models and experimented with Recurrent Neural Networks, Convolutional Neural Networks, and Multilayer Perceptrons with a wide variety of text-based features. We analyzed thoroughly and documented the performance and drawbacks of each experiment before settling down for a two-step classification model based on Multilayer Perceptrons. Our best FNC score of 9586.25 was submitted to the CodaLab leaderboard.

1 Introduction

With the Internet being easily accessible in the 21st century to most parts of the world, it has its own advantages and disadvantages. One such disadvantage is the rise of Fake News after the social media boom. A recent article published in Nature Communications (Bovet and Makse, 2019) reveals how influential and threatening fake news was during the 2016 US Elections. They identified 7.5 million tweets that spread fake news from a total of 660,000 accounts on Twitter. Fake News presents factual information that seems to be true but it isn't and a large number of people are influenced by it.

Detecting fake news has been a challenging task which is why it is done manually. (Pomerleau and Rao, 2017) organized the Fake News Challenge - 1 (FNC1) in 2017 to utilize artificial intelligence and natural language processing specifically to filter out fake news posts. The specific task falls under the category of Stance Detection which requires estimating the relativeness of two textual pieces. FNC dataset comprising of short headlines with their corresponding articles and their stances was released along with the challenge. Table 1 describes the four Stances agree, discuss, disagree, unrelated with context to a headline article pair. This makes Stance Detection a complex task as compared to simply classifying a headline as fake news or not.

Several teams had participated in the original challenge and open-sourced their final submission to the general public. We took inspiration from the top three submissions and modeled our solutions based on them. Initially, we experimented with Recurrent Neural Network (hereafter: RNN) based models since they are proven to work well with long sequences of texts. Our final model was based on Multilayer Perceptrons (hereafter: MLP) after having achieved a high FNC score on the CodaLab leaderboard. Finally, we thoroughly analyzed and present a discussion on the experiments we performed along with recommendations for future work. Although we achieved a high FNC score, one of the predicted classes had a low recall and therefore we propose an alternate scoring metric based on a weighted F1 score that takes care of classes with less representation.

2 Background Work

According to (Augenstein et al., 2016), Stance Detection is the task of classifying attitude expressed in a text towards a given target. Classifying Fake News falls under Stance Detection and extensive research has been carried out in this domain. FNC1 involved predicting the stance given a headline-

[*] equal contribution

Headline: Hundreds of Palestinians flee floods in Gaza as Israel opens dams	
Agree	GAZA CITY (Ma'an) – Hundreds of Palestinians were evacuated from their homes Sunday morning after Israeli authorities opened a number of dams near the border, flooding the Gaza Valley in the wake of a recent severe winter storm. The Gaza Ministry of Interior said in a statement that civil defense services and teams from the Ministry of Public Works had evacuated more than 80 families from both sides of the Gaza Valley (Wadi Gaza) after their homes flooded as water levels reached more than three meters [..]
Discuss	Palestinian officials say hundreds of Gazans were forced to evacuate after Israel opened the gates of several dams on the border with the Gaza Strip, and flooded at least 80 households. Israel has denied the claim as "entirely false". [..]
Disagree	Israel has rejected allegations by government officials in the Gaza strip that authorities were responsible for released storm waters flooding parts of the besieged area. "The claim is entirely false, and southern Israel does not have any dams," said a statement from the Coordinator of Government Activities in the Territories (COGAT). "Due to the recent rain, streams were flooded throughout the region with no connection to actions taken by the State of Israel." At least 80 Palestinian families have been evacuated after water levels in the Gaza Valley (Wadi Gaza) rose to almost three meters. [..]
Unrelated	Apple is continuing to experience 'Hairgate' problems but they may just be a publicity stunt [..]

Table 1: Example of Headline Article pair with Stance

article pair using natural language processing. FNC organizers developed a baseline (Byron Galbraith, 2017) model with hand-crafted text-based features and a gradient boosting classifier[1]. (A K Chaudhry, 2017) modeled their solution experimenting with various RNN based Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) configurations. They carried out experiments with independent encoding, conditional encoding, and attention mechanisms. Their work was based on (Augenstein et al., 2016) who provided a good insight into conditional LSTM encodings for stance detection with Twitter tweets and corresponding targets. We referred their work for our LSTM based experiments. (Slovikovskaya and Attardi, 2020) utilized transfer learning through the state-of-the-art BERT (Devlin et al., 2018) transformer model for FNC stance detection and claimed to have achieved very promising results. Experiments were performed on an updated dataset (FNC+ARC) which is different from the challenge dataset.

In the FNC challenge, top teams used text-based features as input to various Neural Network models. Cisco Talos Team (Sean Baird, 2017) stood first in FNC challenge and have open-sourced their submission on GitHub[2]. Based on their approach we experimented with Convolutional Neural Networks ((O'Shea and Nash, 2015)) and documented our results. This paper also served as an inspiration for our two-step classification solution. Team Athene[3] (Hanselowski et al., 2017) who stood second, developed extensive hand crafted features apart from features included in the baseline for headlines and articles independently with some combined(headline+article) features as well. Their MLP architecture was inspired by (Davis, 2017), who experimented with various MLP configurations. (Riedel et al., 2017) used BagofWords (BoW) features (Tf-idf vectors) to get comparable results in the competition and got third place. After the competition (Hanselowski et al., 2018) constructively analyzed these top three submissions in detail and reported various drawbacks along with a recommendation for Stacked-LSTM approach which is proven to perform better than the top three submissions.

3 The Stance Detection Task

FNC1 provides an opportunity to explore different techniques for detecting the correct relationship between news headlines and body text. The relationship can be one of the following types: Agree, Disagree, Discuss, and Unrelated. We experimented

[1] https://github.com/FakeNewsChallenge/fnc-1-baseline

[2] https://github.com/Cisco-Talos/fnc-1

[3] https://github.com/hanselowski/athene_system

with various approaches and have described them in the following sections. Section 3.1 contains the description of our approaches while Experiments section consists the explanation of experiments. We talk about our analysis in section 5 and provide results in section 6. Finally, we conclude and provide recommendations for future experiments.

3.1 Data Description

FNC dataset consists of headline-article pairs along with their stances. The training data consists of 1683 articles and 49972 labeled pairs. The testing dataset consists of 25413 pairs. The training data is quite imbalanced as there is less number of Agree(7.36%), Disagree(1.68%), Discuss(17.82%) examples as compared to Unrelated(73.13%) ones as described in table 2. Headline length ranges from 2 to 40 with an average of 11 words whereas the length of articles ranges from 2 to 5000 with an average of 350 words.

The scoring criteria of FNC is a 2-step process. The figure 1 taken from official FNC webiste[4] details the complete process. 25% of the score is awarded if the model predicts related or unrelated correctly. 75% of the score is further awarded if the prediction within related classes is the same as the true label.

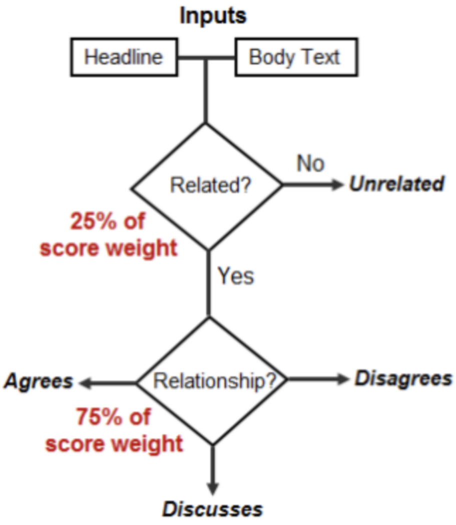

Figure 1: FNC Scoring Criteria

3.2 Approach 1 - LSTMs

The official baseline provided by the FNC team is a single gradient boosting classifier which classifies the headline-body pair into 4 categories. As such, our first approach was to build a single classifier for all categories. (A K Chaudhry, 2017) published a preprint where they modeled their solution based on LSTMs to solve the FNC challenge as part of their CS224N project. Their results are based on a development set created by them as the official test set for FNC was not released until then. Hence, we tried to experiment with basic LSTM configurations to check results on the FNC test set. LSTM (Hochreiter and Schmidhuber, 1997) is an RNN based architecture that can handle the vanishing gradient problem while processing sequences. It has memory units that help to process time-series data. In FNC, article bodies can be long, and hence, we experimented with LSTMs.

3.3 Approach 2 - Tf-idf with Dense Layers

The LSTM approach did not provide decent results on the test set. We came across (Riedel et al., 2017) who happened to experiment with a simple Tf-IDF based MLP approach. We further tried reproducing their results. Tf-IDF stands for Term Frequency-Inverse Document Frequency[5] which calculates the importance of each term in the document. It's value increases with the increase of term frequency in a document, whereas it decreases if more documents contain that term. It is one of the best techniques for term weighting and is primarily used to find relevance between a document and a query. Similarly in FNC, it can be used to find relevance between article body and headline.

3.4 Approach 3 - MLP Classifiers with hand-crafted features

Experiments performed with Classical Machine Learning Algorithms and RNN based architectures as described above resulted in decent testing accuracies but low FNC scores. To further improve our score we explored MLPs. We came across Team Athene's submission on GitHub[6] and utilised their MLP architecture. They also developed an extensive number of text-based features (Hanselowski et al., 2017) such as non-negative matrix factorization cosine distance, latent dirichlet allocation, non-negative matrix factorization concatenated, latent semantic indexing, word similarity on top of the baseline features.

Our base network of dense layers was similar to Team Athene. We introduced more Dense layers

[4] http://www.fakenewschallenge.org/

[5] https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[6] https://github.com/hanselowski/athene_system

	Agree	Disagree	Discuss	Unrelated
Train Set	3678	840	8909	36545
Test Set	1903	697	4464	18349

Table 2: Number of Examples under each Category

Type of example	Up-scaling factor	Number of examples
Agree	1	7356
Disagree	6	5880
Discuss	0	8909
Unrelated	0	36545

Table 3: Training examples in up-scaled data

along with Dropout, BatchNormalization, and Kernel regularization to prevent overfitting. We also divided our solution into a two-step classifier in order to reduce the number of misclassifications. Upsampling the training data also enabled us to improve our performance on the classes with less representation.

4 Experiments

We tried to experiment with various configurations of suggested approaches in the previous section. These experiments provide very good insight into the benefits and limitations of different approaches. All our experiments were performed on the Compute Canada servers with 1 Tesla T4 GPU, 8 CPU cores, and a memory size of 30GB.

4.1 LSTMs with Simple Attention

The first experiment with LSTMs is based on the independent encoding of headline and article using 2 parallel LSTMs consisting of 100 units each. The final hidden states of both LSTMs are concatenated and used for classification using a dense layer with softmax activation. The second experiment uses a single bidirectional LSTM of 100 units which are fed with a combined representation of headline and article body. The maximum length for the headline is taken as 20 and for the article, the body is 600. The learning rate is 0.001 and the number of epochs is 40 with the adam optimizer. For better performance, pre-trained 100-dimensional word embedding vectors from the Stanford Glove dataset(Pennington et al., 2014) are used. The results for both the approaches are being shown in table 4. We experimented with different values for epochs, headline and article length and number of LSTM units or layers. We also experimented with a simple attention layer (Bahdanau et al., 2014) which improved the test accuracy but resulted in no significant improvement in the FNC score.

LSTM Type	Test Set	Validation	Score
Simple	39.37%	73.3%	4587.25
Bidirectional	53.99%	96.38%	6291.50

Table 4: Test and Validation Accuracy for LSTMs

4.2 Tf-idf vectors

We initially reproduced the results mentioned in (Riedel et al., 2017). The Tf vectors of headline and article body are extracted using the vocabulary of 5000 most frequent words in training and test set. Tf-IDF vectors are also extracted in the same manner for both headline and body. A list of stopwords[7] was used to exclude from the vocabulary. Then, tf-vector of the headline(size = 5000), tf-vector of the article(size = 5000), and cosine similarity between tf-idf vectors of headline and body, were combined together to make a final feature vector of size 10001. This vector was fed to an MLP with one hidden layer of 100 units and a Dense layer with softmax activation. We also included a dropout layer with 60% dropout. Dropout is a regularization technique that randomly drops out a select percentage of the nodes to reduce overfitting and improve generalization. The number of epochs was 90 as mentioned in the paper. This approach gave us better results than the LSTMs, but the accuracy was less than what was claimed in the paper. For further experimentation, tf-IDF vectors were used in place of tf-vectors in the final vector representation. We also experimented with Batch Normalization. Batch Normalization is a technique that accelerates training by standardizing the inputs to a layer for each mini-batch. This experiment improved the FNC score as shown in table 5. This was a decent improvement over the baseline accuracy(75.20%).

[7] https://github.com/uclnlp/fakenewschallenge

Configuration	Test Set	Validation	Score
Tf	79.05%	98.6%	9211
Tf-Idf	79.94%	98.78%	9314.75

Table 5: Test and Validation Accuracy for Tf-idf

4.3 Two-step classifier

In the next sub-section, we experimented with two-step classifiers inspired by (Sean Baird, 2017). We combined agree, disagree and discuss examples into a single class called related. The first classifier(hereafter: model1) predicts either related or unrelated class while the second classifier(hereafter: model2) predicts on all four classes together. Two separate approaches have been implemented as described in next sub sections.

4.3.1 Combination of Tf-idf and Dense Layer

The first classifier uses the same strategy as explained in section 3.3 with Tf-idf vectors. Similar model configurations were used. This resulted in an accuracy of 96.68% for the realted/unrealted classification. In the 2nd step, headlines and article bodies were tokenized and concatenated together. After experimenting with different combinations, the headline length was taken as 15 and body length as 700. Pre-trained 100-dimensional word embeddings from Stanford Glove dataset were utilized as feature vectors. For improved performance on agree and disagree classes since they had the least number of examples, we upscaled the examples as described in table 3. Tokenized length vector of size 715 was fed into an embedding layer followed by Dense layer of 64 units with relu activation. Dropout layer(with probability of 0.6) and batch normalization layer was also added. Batch size was 500 and number of epochs was 20. This approach gave us better results(FNC score of 9271.75) than the FNC baseline, but less than Tf-idf approach.

4.4 MLP Classifiers

In section 3.4 we discussed our approach of a two-step MLP classifier. We developed a number of hand-crafted feature vectors which are:

- FNC Baseline Features
  - Word N-Gram Overlap
  - Refuting Features
  - Polarity Features
  - Word Cooccurence Features
- Bag of Words - TF-IDF of 5000 most occurring words between headline and article bodies
- Cosine Similarity of TF-IDF between headline and bodies
- Cosine Similarity of word vector embeddings for tokens in headline and bodies
- KL Divergence of tf-idf of tokens between headline and bodies

Features vectors were developed on the train and test set while the training labels were one-hot encoded. We trained on these features with a simple Dense layer architecture as described in 4.3.1. Table 6 describes the test and validation accuracies. To reduce overfitting and improve model performance we used the same architecture as described in (Hanselowski et al., 2017). With 7 Dense layers with 362, 942, 1071, 870, 318, 912, and 247 hidden neuron units respectively, custom kernel and bias initializer, relu activation function, and an output Dense layer with softmax activation. Table 6 shows the best FNC score and test accuracy achieved by this experiment.

Neural Networks perform best with a large number of examples. To improve on the classifications we upsampled our data as described in table 3. As discussed in section 4.3.1 we divided our approach into a two-step classifier. Model1 consisted of three Dense layers with 250, 50 and 50 hidden neuron units, single dropout layer with 60% dropout, two batch normalization layers, and output Dense layers with softmax activation. The model configurations were: kernel and bias initializers for each layer, 90 training epochs with an early stopping callback, binary_crossentropy loss function, Adam optimizer with a 0.001 learning rate, batch size of 288 and 20% training data reserved for validation. Figure 2 depicts the architecture diagram of Model2. The model configurations were similar as described except categorical_crossentropy loss function.

Figure 3 depicts how a combination of two models was used to predict the test set. Model1 predicts related or unrelated. If the prediction is related, Model2 further predicts the respective related class. This combination helped us achieve our best Codalab score of 9586.25.

Model	Test	Validation	Score
Hand Crafted Features with Simple Dense	87.15%	94.36%	9347
Hand Crafted Features with FEATMLP	88.85%	96.04%	9502
Hand Crafted Features with Dense (related/unrelated	96.86%	99.2%	
Hand Crafted Features with Dense (all classes)	89.15%	96.12%	9548
Two-Step combined Model	89.87%	96.08%	9586.25

Table 6: Test and Validation Accuracy for MLP

5 Analysis of Models and Features

5.1 Model Analysis

5.1.1 LSTMs with Attention

Rigorous experiments and testing helped us explore various neural network models and features. Extensive research on Recurrent Neural Networks based LSTMs has shown that they perform well on long sequences of textual data. We followed suit and experimented with LSTMs for our initial models. Pre-trained glove vectors with various dimensions(50, 100) were used to build the input Bag-of-Words (BoW) features for the Bidirectional LSTM model with concatenated headline and article pairs. The output of the LSTM layers were fed into a single Dense layer with softmax activation to output the probabilities of the classes. As shown in table 4, this model resulted in an FNC score less than the baseline.

We used a simple attention layer as described in section 7 which considerably improved the FNC score and test accuracy. Further adding a Dropout layer after the Bidirectional LSTM slightly improved the performance. We also experimented with different hyperparameters such as dimension size of the word vectors, activation functions, dropout size, regularization, learning rate, and sequence length of the headline-article pair. Interestingly, validation accuracy was around 96% while the testing accuracy was lower. We performed 5-fold cross-validation and to our surprise validation accuracies remained the same.

We analyzed the shortcomings of the LSTM model and found that Headline-Article pairs in the experiments we performed, had huge sequences with nearly 650-750 tokens for each pair. We established from our experiments that LSTMs do not work well with long sequences and tend to forget information. Investigating the reason mentioned by (Davis, 2017), we confirmed that RNNs would quickly learn to fit on the training data but would perform poorly on the test data. Another possible reasoning is that there might not be enough data for the RNNs to generalize unseen data.

5.1.2 CNN and MLP

Following the experiments conducted by (Sean Baird, 2017), we used CNN with the BoW features. We selected a high filter size due to the large size of the input feature vector. Experiments with CNNs did not yield improved results. Surprisingly MLP with lexical hand-crafted features performed well. (Hanselowski et al., 2017), (Riedel et al., 2017) and (Davis, 2017) experimented with various configurations of MLP to achieve the second-best score on the FNC leaderboard. According to (Hanselowski et al., 2018), MLP had the best F1-macro scores across all classes and performed better than Talos and UCL models. Our best strategy was based on the FeatMLP (Hanselowski et al., 2017) model with a two-step classifier.

Our reasoning behind the better performance of the MLP model is based on the inclusion of various hand-crafted lexical features. The increase in features gave the MLP to train better and improve its performance. A two-step approach further helped us reduce the misclassifications from Model1. As shown in figure 8, the validation loss tends to fluctuate which denotes our model was overfitting. We regularized the weights, experimented with various probabilities for Dropout and batch normalization which subsequently reduced the validation loss and overfitting. Complex models like ours tend to overfit with moderately sized datasets. To tackle this, we upscaled our training data which improved test accuracy for agree and disagree specifically. Referring to figure 10 we can see that disagree has the least accuracy because of the least number of data points and confusing examples that the model was not able to generalize upon.

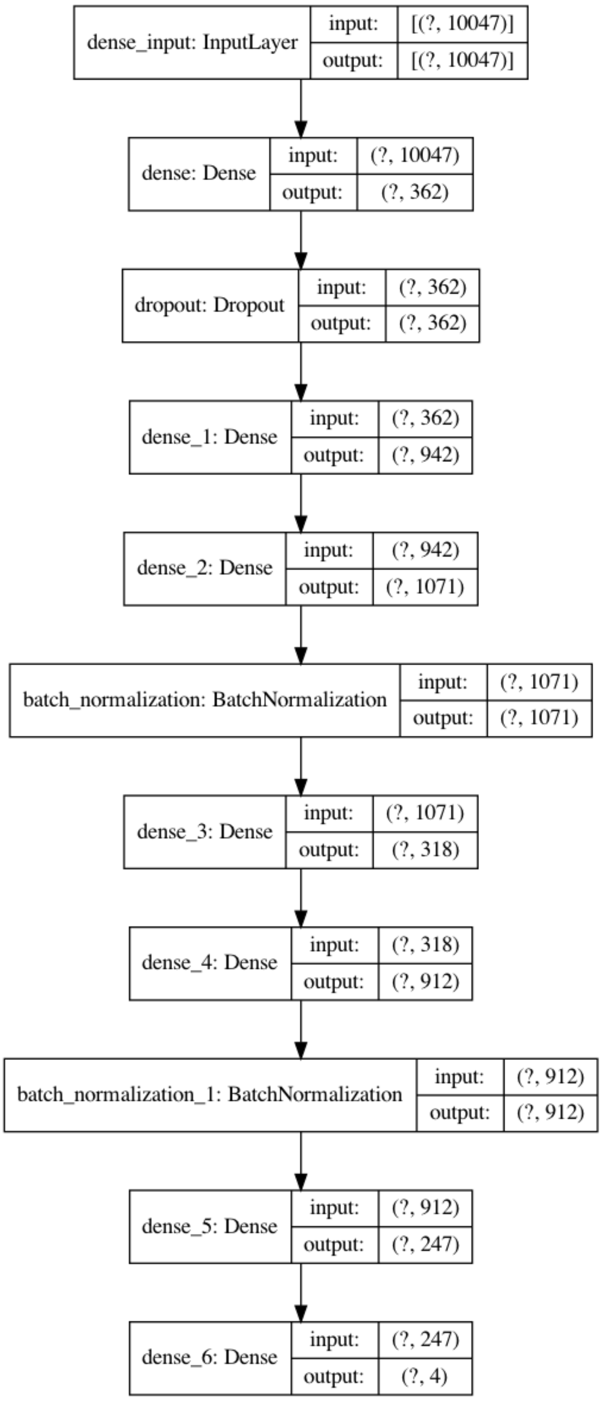

Figure 2: Model2 Architecture

5.2 Feature Ablation Test

We experimented with a number of lexical handcrafted features and documented their performance on the test set. Table 9 describes the performance of each feature either alone or in a combination of features. As we can see, the baseline co-occurrence features contribute towards majority test accuracy while other features slightly improve the performance. We conclude that the combination of all the features together resulted in the best performance. We analyzed the misclassifications on the validation set and saw that features are unable to capture the semantic meaning of the text and hence confuse the model. Example 1 in table 11 in Appendix A shows a headline-article pair that was misclassified by our model. The article contains the exact tokens as the headline and therefore our model predicts it as agree. What the features missed to capture was that the article heading contains the terms 'fake news' meaning the article is not true. We thus came to the conclusion that lexical features do not capture semantic relationships between the words which result in several misclassifications.

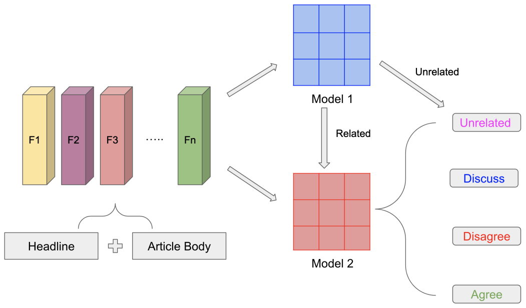

Figure 3: Model Architecture

5.3 Error Analysis - Misclassifications

- Example 2 in table 11 in Appendix A, the headline actually agrees with the article if we read it carefully. The difference between discuss and agree has not been described properly. The predicted stance is agree while the ground truth is discuss for this example.
- Example 3 in table 11 is a typical example of a sarcastic article that confuses the model. Again the difference between agree and discuss is not clearly defined and the model misclassifies this example.
- Example 4 in table 11 contains a very short headline for which there would be no correlation between the article and headline features. The model predicts it as unrelated where the truth label is discuss.
- Example 5 in table 11 contains characters apart from alphabets for which the features developed do not make sense and makes it difficult for the model to predict.
- Example 6 in table 11 is a typical example of how lexical features fail to capture the se-

mantic meaning of the sentence. It is clearly mentioned in the article that the actor ”was not right for the role and wants to withdraw”. The truth stance is agree while the predicted stance is disagree.

6 Discussion and Result

We performed extensive experiments involving RNN, CNN, and MLP with a number of feature vectors. We experimented with attention mechanisms, hyperparameter tuning, regularization, dropout, and batch normalization to achieve our best FNC score. LSTMs and CNN did not result in high test accuracy and we hence settled for MLP with a combination of lexical features built on the headlines and articles as the input data. We analyzed and recorded our observations for all experiments we performed.

Figure 4, 5, 7 and 8 in Appendix A refer to the various accuracy and loss curves for both Model1 and Model2. For Model1, figure 4 shows that the validation accuracy was the same as training accuracy but for Model2, the validation accuracy fluctuates denoting overfitting (7). To overcome overfitting, we introduced weight initializers, regularisation and dropout which did significantly reduce the amount of overfitting. We also plotted the training accuracies with different learning rates (Figure 6, 9) for both the models. Although the curves overlap, we selected 0.001 as the best learning rate since it gave us the best performance.

We used a two-step classifier to improve the overall performance and reduce the misclassifications. Figure 10 shows the confusion matrix for the final predictions. Unrelated examples were predicted with near-perfect accuracy while most of the disagree classes were misclassified. We reported our findings on why that happened. Overall we were able to achieve a final FNC score of 9586.25 which is submitted to the CodaLab leaderboard.

6.1 Alternate Scoring Method

Retrospective analysis conducted by (Hanselowski et al., 2018) shows that the FNC scoring metric is flawed. It favors the classes with maximum representation. If a model randomly predicts all the related classes as ’discuss’ then the test accuracy would be 83.3% which is higher than the team that came first in the challenge. It can be very well seen from our result in table 10 that only 10 disagree examples were correctly classified out of a total of 697 examples with low recall and precision in table 8 but still we have a high FNC score of 9586.25. Therefore the authors propose an alternate scoring method based on the macro and weighted F1 scores that give importance to the number of examples of each class.

7 Conclusion and Future Work

We researched and went through publicly available repositories for Fake News Challenge that was conducted in 2017. We performed an extensive number of experiments mentioned in table 7 in order to develop the best model and achieve a decent FNC score. We experimented with different feature vector and model combinations and finally settled down with hand-crafted feature vectors and two-step Multilayer Perceptron Classifier. Although we achieved a decent FNC score, ’disagree’ examples specifically were mostly misclassified and we discussed the reason behind it. For the future, we recommend experimenting with stacked LSTM that is proven to work well with a long sequence of texts. We also recommend an alternative scoring metric based on a weighted-F1 score that calculates the score based on the number of examples in each class. It is believed that transformer-based models such as BERT, XL-Net, and Roberta perform well on NLP related tasks and can be used to experiment in our scenario. In conclusion, we were able to identify the advantages and drawbacks of our model and have thoroughly analyzed in the discussion section. Complete code for our project is available on Github.

Acknowledgments

We sincerely thank Prof. Olga Vechtomova for her support throughout the entire duration of the course. We were clueless when RNN based approaches weren’t improving our scores and Prof. Olga’s recommendations helped us for our further experiments. We also thank her for granting us access to the Compute Canada servers. Special thanks to our TA Gaurav Sahu for quick replies to our doubts and issues over email.

References

Philipp Thun-Hohenstein A K Chaudhry, Darren Baker. 2017. Stance detection for the fake news challenge : Identifying textual relationships with deep neural nets. In *CS224n: Natural Language Processing with Deep Learning (2017)*.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Alexandre Bovet and Hernán A Makse. 2019. Influence of fake news in twitter during the 2016 us presidential election. *Nature communications*, 10(1):1–14.

HJ van Veen-Delip Rao James Thorne Yuxi Pan Byron Galbraith, Humza Iqbal. 2017. *FNC Baseline*. https://github.com/FakeNewsChallenge/fnc-1-baseline.

Richard Davis. 2017. Fake news , real consequences : Recruiting neural networks for the fight against fake news.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Andreas Hanselowski, PVS Avinesh, Benjamin Schiller, and Felix Caspelherr. 2017. Description of the system developed by team athene in the fnc-1. *Fake News Challenge*.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych. 2018. A retrospective analysis of the fake news challenge stance detection task. *arXiv preprint arXiv:1806.05180*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *ArXiv e-prints*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Dean Pomerleau and Delip Rao. 2017. The fake news challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news. *Fake News Challenge*.

Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *ArXiv*, abs/1707.03264.

Yuxi Pan Sean Baird, Doug Sibley. 2017. Cisco talos. https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html.

Valeriya Slovikovskaya and Giuseppe Attardi. 2020. Transfer learning from transformers to fake news challenge stance detection (FNC-1) task. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1211–1218, Marseille, France. European Language Resources Association.

8 Appendix

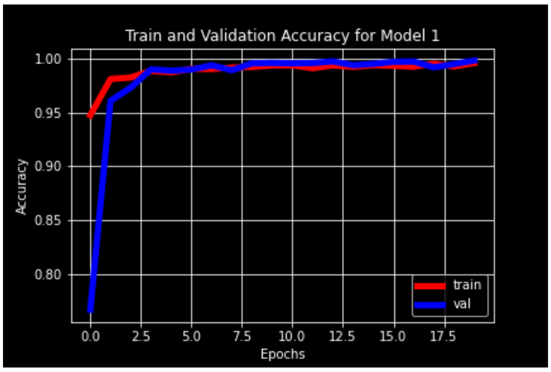

Figure 4: Model 1 Accuracy

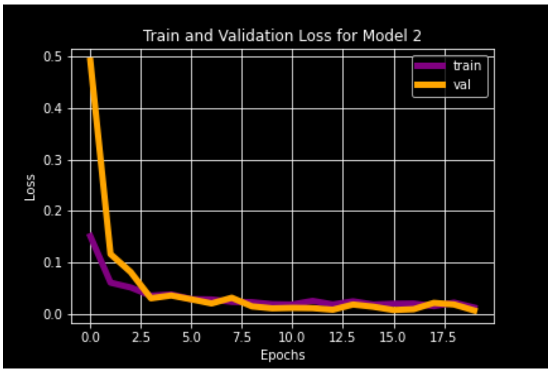

Figure 5: Model 1 Loss

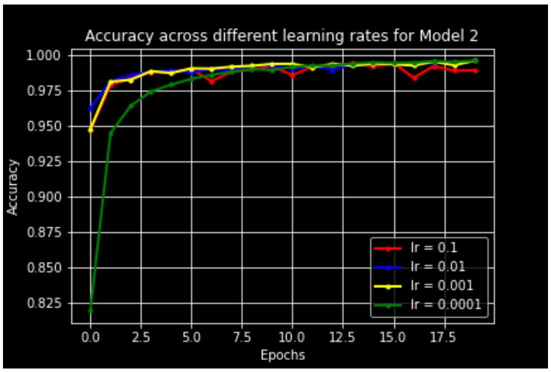

Figure 6: LR vs Epochs for Model 1

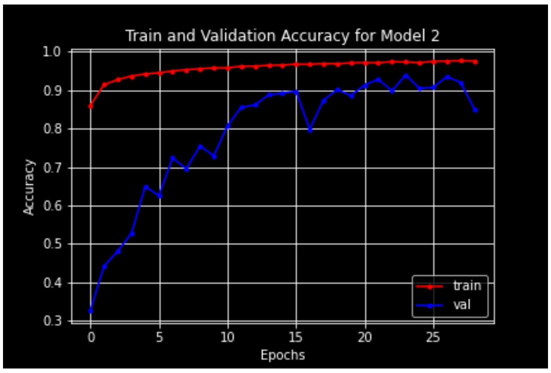

Figure 7: Model 2 Accuracy

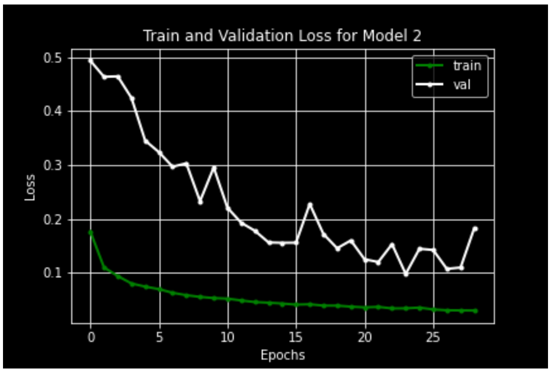

Figure 8: Model 2 Loss

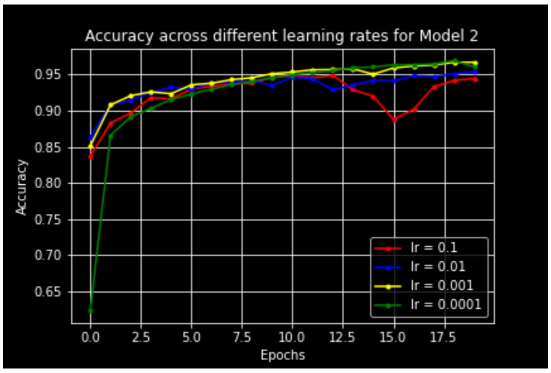

Figure 9: LR vs Epochs for Model 2

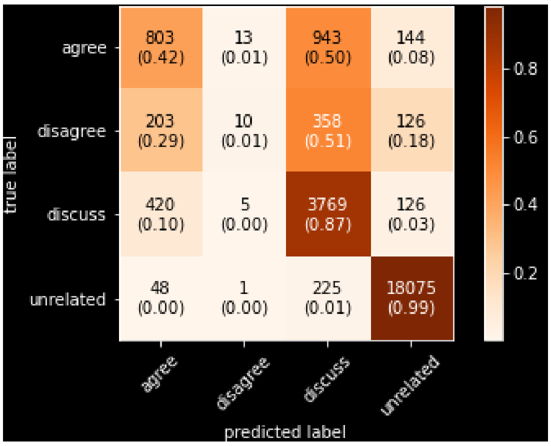

Figure 10: Confusion Matrix

Single step tf-idf approach using tf vectors or tf-idf vectors
Single step simple or bi-directional LSTM(with attention, Dropout, regularization) and GRU cells
2 Step classifier with tf or tf-idf vector, hand-crafted feature, dense layers, LSTMs and CNN
Upscaling agree and disagree examples
CNN with BoW features
MLP with lexical hand-crafted features

Table 7: List of Experiments done

	Precision	Recall	F1
Agree	0.54	0.42	0.48
Disagree	0.34	0.01	0.03
Discuss	0.71	0.84	0.77
Unrelated	0.97	0.99	0.98
Macro Avg	0.64	0.57	0.56
Weighted Average	0.88	0.89	0.88

Table 8: Precision, Recall and F1 score for the Models

Features Included	Test Accuracy (%)
Word Overlap	72
Word Overlap + Refuting	72
Word Overlap + Refuting + Polarity	76
Hand Features - Cooccurrence	78
Word Overlap + Refuting + Polarity + Hand	83
KL Divergence + BoW + Cosine Similarity	79
All	89

Table 9: Test Accuracy with different Features

	Agree	Disagree	Discuss	Unrelated
Agree	803	13	943	144
Disagree	203	10	358	126
Discuss	420	5	3769	270
Unrelated	48	1	225	18075

Table 10: Final Score

	Headline	Article	Headline ID	Article ID
1	Priest who died for 48 minutes says he met God and sheâ$^{\text{TM}}$s a woman	Description: Fake news / Satire Circulating since: Feb. 2015 Status: False (see details below) Example 1: Via WorldNewsDailyReport.com, Feb. 4, 2015: CATHOLIC PRIEST WHO DIED FOR 48 MINUTES CLAIMS THAT GOD IS A WOMAN February 4th, 2015 — by Barbara Johnson Boston— A Catholic priest from Massachussetts [..]	48663	1303
2	Michael Phelps' self-proclaimed girlfriend says she was born a BOY and reveals 'amazing' sex with Olympian	SYDNEY - A girlfriend of Michael Phelps has revealed that she was born and brought up a man, reports said on Thursday. Taylor Lianne Chandler, who dated the 29-year-old US swimmer for months, wrote a long post on her Facebook page on Nov 14 in which she revealed that she was ”born intersex and named David Roy Fitch” at birth. ”I was born with male genitalia with no testicles, but I also have a uterus and no ovaries,” the 41-year-old told Radar Online in an interview [..]	49677	599
3	North Korea dictator Kim Jong-un undergoes surgery after breaking both ankles on military tour	North Korea leader Kim Jong-un is so fat from eating cheese that he has broken his ankles. The tubby tyrant is recovering from surgery after weight gain caused him to fracture his joints. Metro reports that daily binges on Emmental are believed to be responsible [..]	49784	1095
4	Staff Reporter	A photo of a woman sitting in front of what appear to be dozens of iPhone 5Cs has gone viral in China. While few details about the photoâ$^{\text{TM}}$s origins are available, media have speculated she works at a Chinese App Store ranking manipulation farm. [..]	48379	1784
5	Weather Channelâ$^{\text{TM}}$s Mike Seidel was not caught with his pants down	Thereâ$^{\text{TM}}$s getting caught short and then there is this guy. Weather forecaster Mike Seidel obviously thought he had enough time to answer a quick call of nature, but clearly he was wrong. In a video uploaded to YouTube by Kris Tatum, NBC Nightly news reader Lester Holt cuts to Seidel ready for the forecast. Holt says: â˜Letâ$^{\text{TM}}$s bring in Weather Channel meteorologist Mike Seidel heâ$^{\text{TM}}$s in Sugar Mountain, North Carolina. Hi Mike.â$^{\text{TM}}$ But then there is an awkward pause until Seidel eventually says â˜Why?â$^{\text{TM}}$ Holt adds: â˜Well, obviously Mikeâ$^{\text{TM}}$s not ready for us.â$^{\text{TM}}$	38651	1829
6	Christian Bale Pulls Out of Upcoming Steve Jobs Biopic	Christian Bale will not be starring as Steve Jobs in Aaron Sorkin's upcoming Steve Jobs biopic, according to The Hollywood Reporter. The actor has reportedly decided that he was ”not right for the part,” deciding to withdraw from the film. [..]	7855	1846

Table 11: Examples of misclassifications on the Train dataset