RNNs for Stance Detection between News Articles

Jason Yu Chen
Stanford University
Stanford, CA 94305
cheson@stanford.edu

Joseph Johnson
Stanford University
Stanford, CA 94305
jjohnso3@stanford.edu

Graham Yennie
Stanford University
Stanford, CA 94305
gyennie@stanford.edu

Abstract

We implement a bag-of-vectors (BOV), LSTM, and neural attention model to predict stances between news article bodies and headlines for the Fake News Challenge Data Set. Our bag-of-vectors model significantly outperforms the FNC-1 baseline model, without requiring feature engineering.

1 Introduction

The spread of fake news articles has generated noticeable concern recently, as false or misleading stories can spread faster and reach a wider audience over social media. Given that changing technology has played a major role in enabling the spread of fake news, a natural question is whether technology can also help warn users about false claims. While technology is certainly not advanced enough to evaluate the truth of a claim on its own, it could be used to aid journalists and make it easier for them to detect and debunk false statements.

One application that we will explore in this paper is to use Natural Language Processing techniques to determine whether a body of text agrees, disagrees, discusses, or is unrelated to another. This model could be applied as an automatic fact checker that could read an article, and then find other articles that either disagree or agree with its content. We will draw on methods from the field of Natural Language Inference to build a model that classifies the relationship between a news article headline and the body of a different news article.

2 Background

While automated fact checking and stance detection has not yet garnered much attention from researchers, previous research in Natural Language Inference has worked on problems that are very similar to ours. NLI attempts to identify the relationship between two statements, by identifying whether two bodies of text support, contradict, or are neutral towards one another. Researchers in NLI have achieved reasonable success on this task using neural network models, and almost all of the the best performing models on the benchmark Stanford SNLI corpus have incorporated neural networks. Matching LSTMs were successfully used for NLI by [1] to achieve a performance of 86.1% on the SNLI dataset. [2] tests a LSTM model, attention model, and a word-by-word attention model on the SNLI corpus. In their paper, word-by-word attention achieves the highest test accuracy of 83.5% on the SNLI dataset. A sequential LSTM-based model combined with a syntactic parsing model was used by [3] to achieve 88.3% accuracy on the SNLI corpus.

3 Approach

We train 3 different models: a Bag of Vectors Model, LSTM, and RNN with Attention.

3.1 Baseline Model: Bag of Vectors

The baseline model utilizes a straightforward bag of vectors approach. The model creates a L2-normalized sum of the embedding vectors for each of the words in the headline and body text. This new vector naively captures the meaning between the texts through summing their embedding vectors. To determine the prediction of the relationships between the headline and the body, the result is passed through a multilayer perceptron and softmax classifier to generate the final output. The intent of this model was to have a working baseline that has shown success in Natural Language Inference applications[4]. Several MLP architectures with both tanh and relu activations were constructed for training.

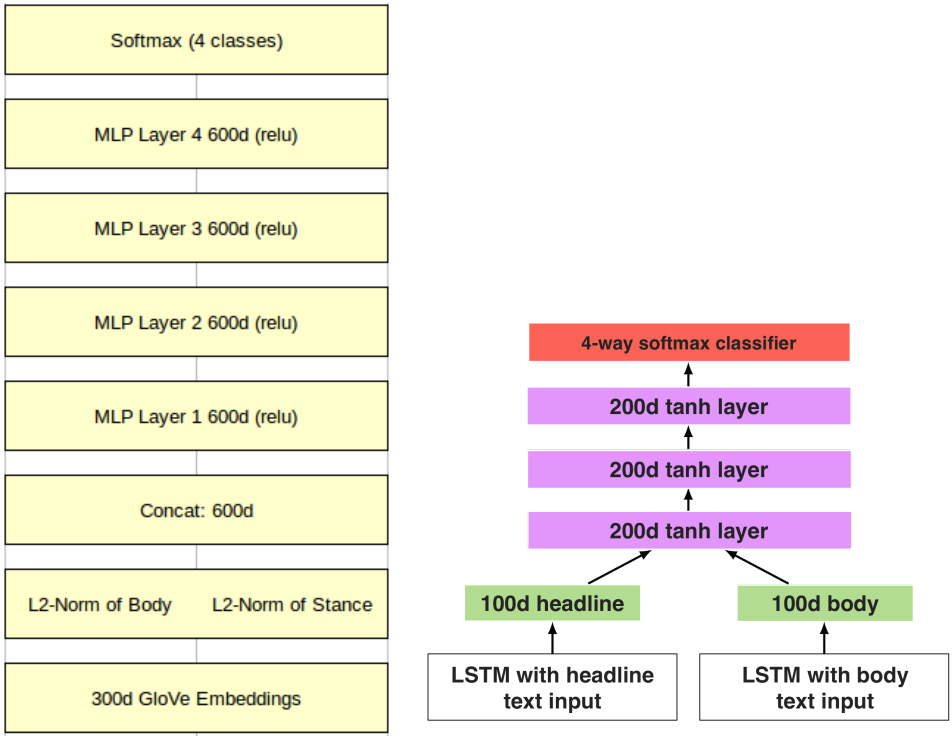

Figure 1: BOV architecture (left), LSTM architecture (right) [4]

3.2 LSTMs

The LSTM model is a sequence-to-sequence model replicated from [4] and modified to our task. It uses two LSTM encoders to generate separate encodings of the headline and body text of dimension *d*. Next, the encodings are concatenated to form a vector of dimension *2d*. Like the baseline model, the concatenated vector is passed through a multilayer perceptron and finally a softmax classifier to generate the final output. However, we modify this model from the original by treating the dimensions as hyperparameters, especially since our headlines and bodies are of significantly different lengths. We experiment by allowing the LSTM processing body text to be a much higher dimension than 100d, hoping that it will allow a better representation of the longer body. Our 4-way classifier is also going to be evaluated by a weighted scoring mechanism because agree/disagree are more telling than related/unrelated, which is different from the equal weighting scheme used by [4]. Finally, we also experiment with implementing multi-layered LSTMs to encode the headline and body.

3.3 RNN with Attention

An often used extension to the sequence-to-sequence LSTM model mentioned above is to add an attention mechanism that allows the body text to attend to the LSTM output layer from the headline to make the final prediction.

The attention model as used in [3] is defined as follows.[1] Let $\mathbf{Y} \in \mathbb{R}^{k \times L}$ be a matrix of the outputs of the LSTM processing the article body with a body length of L and embedding size k. Let $h_N \in \mathbb{R}^k$ be the last output vector of the LSTM processing the headline.

$$\mathbf{M} = \tanh(\mathbf{W^y Y} + \mathbf{W^h h_N} \otimes \mathbf{e_L})$$

$$\alpha = \text{softmax}(\mathbf{w^T M})$$

$$\mathbf{r} = \mathbf{Y}^T \alpha$$

where $\mathbf{e_L} \in \mathbb{R}^L$ is a vector of ones. $\mathbf{W^y} \in \mathbb{R}^{k \times k}$, $\mathbf{W^h} \in \mathbb{R}^{k \times k}$, and $\mathbf{w} \in \mathbb{R}^k$ are weight matrices to be trained. $\mathbf{M} \in \mathbb{R}^{k \times L}$ is a matrix of the intermediate attention representations of each word. $\alpha \in \mathbb{R}^L$ is a vector of attention weights. $\mathbf{r} \in \mathbb{R}^k$ is the weighted representation of the body.

With the weighted representations of each sentence we can predict stances:

$$\mathbf{h}^* = \tanh(\mathbf{W^p r} + \mathbf{W^x h_N})$$

$$\hat{y} = \text{softmax}(\mathbf{h}^* \mathbf{W^{pred}})$$

$\mathbf{W^p} \in \mathbb{R}^{k \times k}$ and $\mathbf{W^{pred}} \in \mathbb{R}^{k \times C}$ are weight matrices to be trained. $\mathbf{h}^* \in \mathbb{R}^k$ is the representation of the headline-body pair, and $\hat{y} \in \mathbb{R}^C$ is the predicted probability that the pair belongs to one of the C stances.

We also test multi-layer LSTMs with attention. Here the attention mechanism acts on the LSTM output in essentially the same manner, but the input article headlines and bodies pass through multiple layers of LSTMs before they are output. This may help us learn higher level structures in the text, at the cost of more parameters and more potential overfitting.

4 Experiment

4.1 Data

Our dataset is obtained from the Fake News Challenge.[2] The dataset consists of 1,684 article bodies and headlines. Bodies and headlines are combined with one another in different ways so that there are 49,973 total body-headline pairs in total. Each pair is labeled as either unrelated, discusses, agrees, or disagrees. Table 1 gives an example of each classification.

Table 1: Example of headline-body pairs

Headline	**Body**	**Stance**
Hundreds of Palestinians flee floods in Gaza as Israel opens dams	Hundreds of Palestinians were evacuated from their homes Sunday morning after Israeli authorities opened a number of dams near the border...	Agree
DHS Rebuffs Congressmans Claim ISIS Infiltrating Southern Border	Judicial Watch has reported that ISIS members crossed the Mexican border...	Disagree
Boko Haram denies it has agreed ceasefire	More than 200 missing schoolgirls kidnapped by the Islamic extremist group Boko Haram may be released as part of an immediate cease-fire agreement...	Discusses
N. Koreas Kim has leg injury but in control	You want a gold Apple Watch, you say? Then it's going to cost you... a lot...	Unrelated

The structure of this dataset causes there to be many more unrelated pairs than pairs that agree or disagree. About 73.13% of article-headline pairs are unrelated, 17.83% discuss, 7.36% agree, and only 1.68% disagree. For fact checking, disagreement or agreement is a far more interesting

[1] Referred to code located at `https://github.com/shyamupa/snli-entailment/blob/master/tf_model.py` for this model, with modifications for our problem

[2] `http://www.fakenewschallenge.org/`

property, so instead of simple misclassification rate, we use a weighted misclassification rate to measure error on the training and test sets. The imbalance of classes in the data was offset by weighting the loss function by class. This weighting was treated as a hyperparameter to shorten the time to train, while maintaining accuracy.

The length of the article bodies had an upper quartile limit of roughly 460 words and the headlines had a limit of roughly 16 words. Although capturing more words in the body should equate to capturing more meaning in the model, later we will discuss how applying a lower cut-off to the body length in the networked models improved the performance of the model.

Pretrained GloVe 100d, 200d, and 300d word vector embeddings[5] were selected as inputs to the RNN and MLP models.

4.2 Hyperparameters

We use a random search to select the number of hyperparameters to use. We search over embedding sizes, body length sizes, learning rates, dropout probabilities, and LSTM cell sizes.

One of the most important hyperparameters is the body size. Too long a body size can make training slower and the words harder to model, but a shorter body length can cut off valuable information about the articles. To demonstrate this, 5 body lengths were trained using the same model for 50 epochs and repeated 5 times. The maximum FNC score by each body length showed that the same model learns faster and with higher score for body lengths near 200 words. The trade-off appears to be the model's ability to capture meaning balanced by the noise of additional words. Figure 2 shows the best accuracy attained for the bag of vectors model trained over different body sizes.

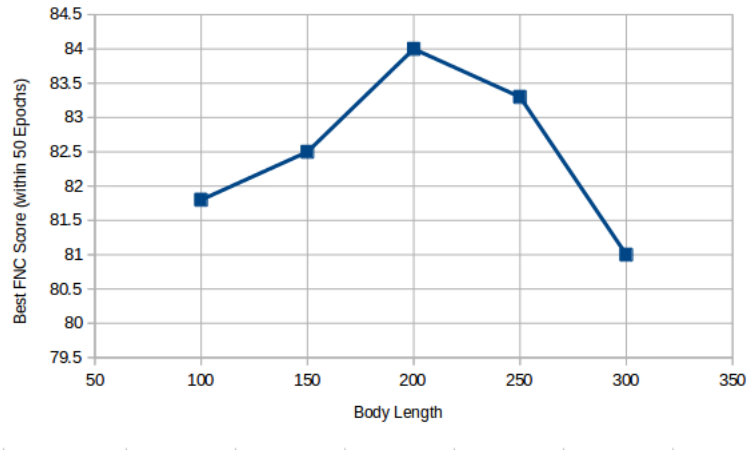

Figure 2: Best score by body length over 50 epochs, repeated 5 times

The figure shows that lower body sizes around 200 perform very well, while higher and lower body sizes yield lower performance over the limited test. This makes sense given that reporters usually are trained to begin their articles with a summary of the news story, while the rest of the article just expands on the story discussed in the beginning.

5 Results

We evaluate our model using a weighted scoring function that matches the scoring function used for the Fake News Challenge (FNC). Distinguishing between unrelated and related articles account for 25% of the score, while distinguishing between agreement, disagreement, and discusses accounts for 75% of the score. That is, for every label that correctly distinguishes whether the headline is related to the article, the score is incremented by 0.25 and if the predicted labels exactly match the true labels and the labels are related (discuss, disagree, and agree) the score is incremented another 0.75. The score is then expressed as a percent of the best possible score where all stances are picked perfectly.

The BOV model, with 4 600D MLP layers scored 86.47 on the dev set. The best model was attained by random search, followed by manual tuning of the hyperparameters. This result translates to an 89.5% overall accuracy. The model achieved F1 scores of 65%, 50%, and 79% on the related labels "agree," "disagree," and "discuss," respectively. The model also took a long time to train to achieve the levels of F1 on labels "agree" and "disagree" required to score high using the FNC metric. Figure 3 shows the development score over training epochs for the best performing model.

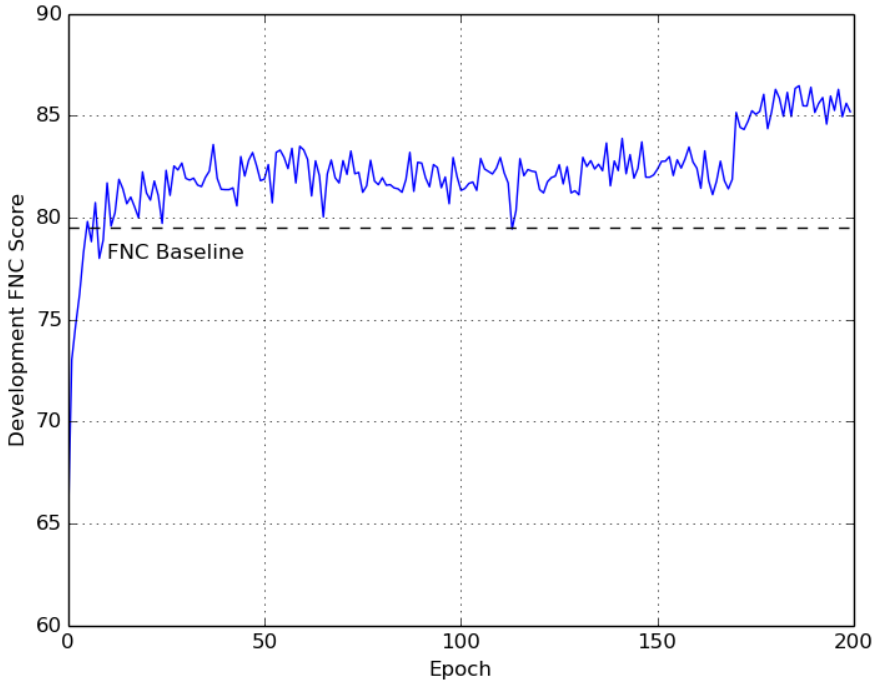

Figure 3: Score on dev set over training epochs for BOV model

The more complex LSTM models unfortunately did not perform as strongly as the bag-of-words model. We started with single layer LSTMs with a dimension of 100 for the hidden layer. This 100d-100d LSTM had a maximum dev score of 76.02% and a maximum test score of 75.16%. Furthermore, using a multi-layer LSTM with the same dimensions did not improve the results at all. Finally, for the asymmetric LSTMs with 100d for the headline and 300d for the body, the dev accuracies were less than ideal. At 25 epochs, dev accuracy plateaus around 55% and peaks at 59.53%.

With a dev score of 85.99, the RNN with attention achieves a score slightly below that of bag-of-vectors on the dev set. The multi-layer RNN with attention does not achieve significant performance gains over the RNN with attention, with maximum dev score of only 82.86. Figure 4 shows the performance of each model over training. On the test set the best RNN with attention model achieves a score of 84.67%, which is also below the simpler BOV model's performance.

6 Discussion

Our best performing model achieves 86.4% score on the test set, which outperforms the FNC benchmark without any feature engineering. This still leaves a lot of room for improvement. The challenges of the data set and the unexpected results of a simple bag of vectors MLP model outperforming state of the art LSTM models with attention warrants creative solutions to the FNC.

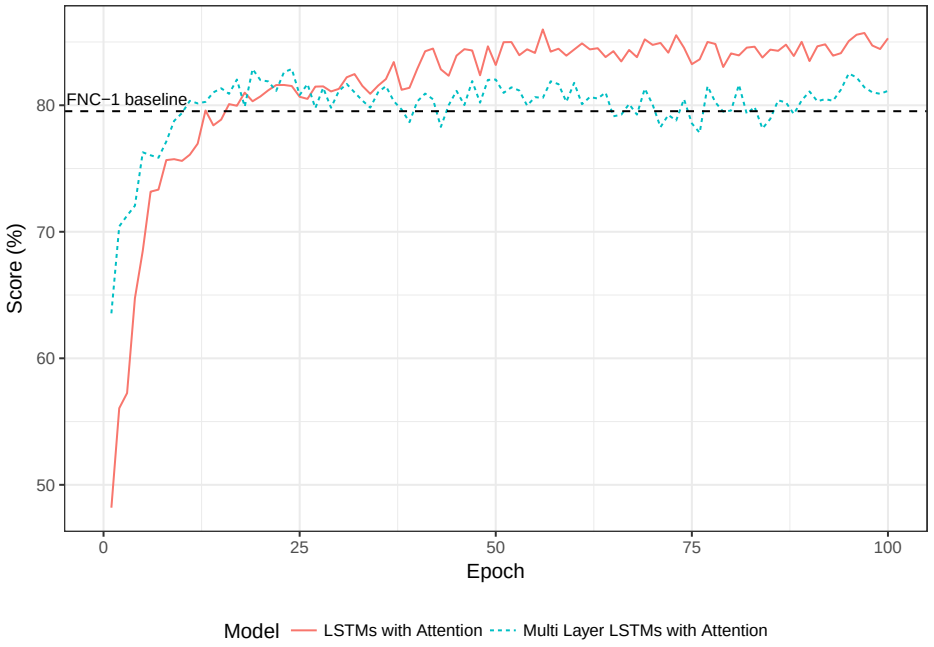

Figure 4: Score on dev set over training epochs for attention model

It is clear that the embedded "meaning" of a news article and a headline come from two populations. Many headlines are designed to sacrificing the actual meaning of the article to enhance potential readers' interest, making the challenge of labeling stances from headlines alone difficult. A more appropriate use of RNN models may be an encoder-encoder or attention LSTM model predicting stances from body to body. Given that accurate predictions can be made judging articles to be related or unrelated, this task could be automated.

7 Future Work

Our bag-of-vectors model worked very well on detecting related/unrelated news articles (93.9% accuracy), but not as well on detecting disagreement, agreement, and discussion. Meanwhile, our RNN with attention model did not do as well at detecting relatedness, but did detect relationships relatively well. A cascading model that incorporates the Bag of Vectors model for relatedness, and then incorporates a attention model for the other categories could potentially improve our score. Perhaps an ensemble approach would improve the score as well.

Acknowledgments

Special thanks to Ignacio Cases for excellent mentorship and advice throughout the project. Thank you to Chris Manning, Richard Socher, and the rest of the CS224N staff for a great quarter.

References

[1] Wang, Shuohang, and Jing Jiang. "Learning natural language inference with LSTM." arXiv preprint arXiv:1512.08849 (2015).

[2] Rocktschel, Tim, et al. "Reasoning about entailment with neural attention." arXiv preprint arXiv:1509.06664 (2015).

[3] Chen, Qian, et al. "Enhancing and combining sequential and tree lstm for natural language inference." arXiv preprint arXiv:1609.06038 (2016).

[4] Bowman, Samuel R., et al. "A large annotated corpus for learning natural language inference." arXiv preprint arXiv: 1508.05326 (2015).

[5] Socher, Richard, et al. "GloVe: Global Vectors for Word Representation." Empirical Methods in Natural Language Processing: 1532–1543 (2014).

8 Contributions

Graham Yennie: Coding training framework for all models, Coding and training MLP neural networks, paper writing

Joe Johnson: Coding and training of Attention RNN, paper writing

Jason Chen: Coding and training of LSTM, paper writing