Stance Detection for Fake News Identification

Damian Mrowca
Department of Computer Science
Stanford University
Stanford, CA 94305
mrowca@stanford.edu

Elias Wang
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
elias.wang@stanford.edu

Atli Kosson
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
akos@stanford.edu

Abstract

The latest election cycle generated sobering examples of the threat that fake news poses to democracy. Primarily disseminated by hyper-partisan media outlets, fake news proved capable of becoming viral sensations that can dominate social media and influence elections. To address this problem, we begin with stance detection, which is a first step towards identifying fake news. The goal of this project is to identify whether given headline-article pairs: (1) *agree*, (2) *disagree*, (3) *discuss* the same topic, or (4) are *not related* at all, as described in [1]. Our method feeds the headline-article pairs into a bidirectional LSTM which first analyzes the article and then uses the acquired article representation to analyze the headline. On top of the output of the conditioned bidirectional LSTM, we concatenate global statistical features extracted from the headline-article pairs. We report a 9.7% improvement in the Fake News Challenge evaluation metric and a 22.7% improvement in mean F1 compared to the highest scoring baseline. We also present qualitative results that show how our method outperforms state-of-the art algorithms on this challenge.

1 Introduction

1.1 Fake news and stance detection

With the advent of fake news being used to influence elections, the identification of false information has become an important task. Governments, newspapers and social media platforms are working hard on distinguishing credible news from fake news. The goal of the Fake News Challenge [1] is to automate the process of identifying fake news by using machine learning and natural language processing. This process can be broken down into several stages. A first helpful step towards the identification of fake news is to understand what other news sources are saying about the same topic [1]. That is why the fake news challenge initially focuses on stance detection. Stance detection comprises the estimation of the relative perspectives of two different text pieces on the same topic as described by [2]. Specifically, the task is to estimate the stance of a news headline, relative to the contents of a news article which can but does not have to address the same topic. Thus, the relative stance of each headline-article pair has to be classified as either *unrelated*, *discuss*, *agree* or *disagree*. An example of these classifications can be found in Appendix A, Table 4. The discovery of a *disagreeing* headline-article pair does not necessarily correspond to the discovery of a fake article,

but it is an automated first step which could make human reviewers aware of a discrepancy. Human reviewers or specialized algorithms can then ultimately decide which articles are fake.

In the following work, we address the problem of stance detection. First, we present a detailed analysis of the stance detection dataset and discuss evaluation metrics. After a brief overview of related work, we describe our bidirectional LSTM model that links local with global features to classify stances of headline-article pairs. Next, we present a detailed evaluation of our model and compare it against several baselines. We show that our model outperforms the best available baseline by 9.7% by combining local word embeddings and global features. Finally, we finish with a discussion of the results and potential improvements to our model.

1.2 Stance detection dataset for fake news classification

In order to understand and effectively solve the problem of stance detection between headline-article pairs, it is crucial to get an in-depth understanding of the dataset.

As illustrated in Table 1 on the following page, the dataset consists of about 50,000 headline-article pairs each labeled with either *unrelated*, *discuss*, *agree* or *disagree*. Two observations must be made here to address challenges down the line.

First, the dataset is highly unbalanced with most of the pairs being *unrelated* and only 1.68% of all pairs labeled *disagree*. As shown in the result section, this can influence results and training time, but can be partially corrected.

Second, the pairs have been generated by repeatedly and randomly pairing headlines and articles. This means that one headline may be paired with multiple articles and vice versa. It turned out to be impossible to split the dataset without either articles or headlines appearing in both training and test sets. Otherwise, a significant number of training examples could not have been used. The organizers of this challenge are aware of this problem, and ended up providing a train/test split in which articles are cleanly split between both sets, but not headlines. Splitting by headlines or randomly would increase data bleeding significantly. Thus, the same headlines appear in both the train and test splits and until the final test set is released in June 2017, these results have to be considered with caution.

In addition to these official statistics, we also analyze the lengths of the headlines and articles in the dataset. This helped us choose a reasonable truncation length for articles and headlines in order to avoid vanishing gradients and to speed up training without losing too much information.

Distributions of article and headline lengths can be seen in Appendix A (Figures 3a and 3b on page 10). Most headlines are about 10 words long and most articles are about 250 words long. The headline lengths are maximally 40 words long. The article lengths are mostly below 700 words and only a few outlier articles exceed that threshold. Intuitively, one might think of clipping the articles at 700 words and to not clip the headlines at all, which would leave over 95% of all articles unclipped. However, as shown in the results section, it is better to clip the articles at 200 words in order to optimize speed and performance.

We further analyze which and how many words in this dataset do not match with any of the GloVe [3] vector representations that will be used for our model. [1] 8,590 word occurrences do not occur in the GloVe vocabulary, which corresponds to about 0.71 % of our data that was mapped to an *UNK* embedding. Out of the 8,590 words, 1,984 were unique. Most of these instances were words that were connected without a space, such as *fidelcastro* or *relatedhalloween*, and random sequences such as *kzhsbgw87a* or *mhcztvzdfd*, which likely correspond to unfiltered hyperlinks. A significant number of unknown words also contained named entities such as *safira* or *9to5mac*, which is typical for a news dataset as news article often report on new named entities which might not have been included at the time of the GloVe vector training.

1.3 Evaluation metrics

In order to get a deeper understanding of the performance of our models, we report multiple metrics as well as confusion matrices.

[1] Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab) set form: https://nlp.stanford.edu/projects/glove/

Table 1: Number of pairs and label distribution in the stance detection dataset [1]

Pairs	Unrelated	Discuss	Agree	Disagree
49,972	73.13%	17.83%	7.36%	1.68%

We use the official evaluation metric of the fake news challenge which incorporates a two-level scoring system. The first level measures the accuracy in classifying a headline-article pair as *related* or *unrelated* and contributes 0.25 points per pair to the overall evaluation metric. For pairs that are *related*, a further 0.75 points are given for classifying a headline-article pair correctly between *agree*, *disagree* and *discuss*. The rationale behind this metric is that classifying a headline-article pair as *unrelated* or *related* is easier than discovering their stance towards an issue. Thus, the credit for labeling a pair correctly as *agree*, *disagree* or *discuss* is weighted more heavily. The scores are then reported as a percentage of the maximum possible score, i.e. the score if the prediction is the same as the ground truth.

Besides the official evaluation metric, we also report the F1 score for each label separately and the mean F1 score overall. Together with confusion matrices, this will allow us to dissect in more detail where the tested models perform best and struggle most.

2 Related Work

As the Fake News Challenge dataset was released only recently,to the best of our knowledge, no current publications address this dataset specifically. However, the domains of stance detection and text classification have been broadly covered in literature.

One closely related task is stance detection in Twitter Tweets [4]. In this task, a stance target - e.g., a politician - is given and the goal is to estimate whether a given Tweet is in favor, against or neutral towards the given target. Obviously, this is similar to deciding whether a given article is *agreeing*, *disagreeing*, *discussing* or *unrelated* to a given headline. [5] tries to solve the Twitter task by using a bidirectional LSTM to read the tweet conditioned on the target. Given the similarity to our task, we draw inspiration from this approach for our model and implement a bidirectional LSTM that conditions the analysis of the headline on the article representation.

Another relevant line of work is text classification. In this domain, [6] used a convolutional neural network to classify sentences. One can think of convolutional filters extracting different n-grams depending on the filter size which are known to be useful for text classification. Thus, we tried adapting this methodology for our approach.

Given our problem of classifying stances of headline-article pairs, attention mechanisms that focus on the relevant parts of headlines and articles could prove to be useful. Despite the fact that [7] focuses on question-context pairs to create an answer to the given question, we try to adapt the dynamic coattention mechanism to stance detection. We also tried to use the attention mechanism introduced by [8].

Only the authors of the fake news challenge have released a simple baseline model for the stance detection dataset at this point that uses a gradient boosting classifier on global co-occurrence, polarity and refutation features, which achieved a score of 79.53% [1] (but only 77.7% on our test split). The features can be seen in Appendix A (Table 6 on page 11). As these global features proved to be useful for this dataset, after a thorough analysis, we integrated those features into our model combining them with the local word-by-word features of our bidirectional LSTM.

3 Methods

In this section, we describe our best performing model: a conditioned bidirectional LSTM with global features. We start with a description of the bidirectional LSTM that extracts features from local word embeddings. We then describe how global, co-occurrence, refutation and polarity features complement the local features, and how this combination ultimately leads to superior performance on the stance detection task.

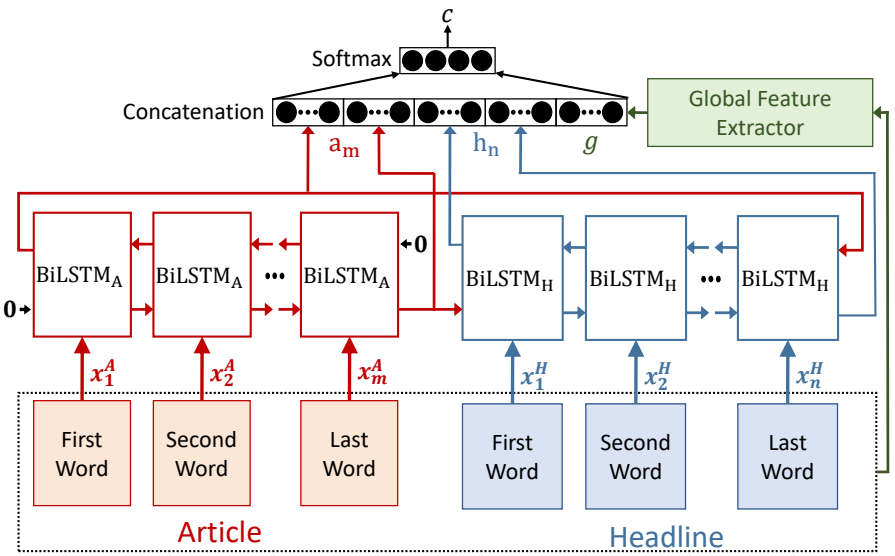


Figure 1: Our bidirectional LSTM with global features model

3.1 Model description

Bidirectional LSTMs have been successfully used across different natural language processing tasks [5, 7]. Thus, we adopt a bidirectional LSTM for the fake news challenge stance detection task. Our model is depicted in Figure 1. Each headline-article pair is processed as follows.

Let $(x_1^H, x_2^H, ..., x_n^H)$ denote the sequence of word vectors corresponding to words in the headline and $(x_1^A, x_2^A, ..., x_m^A)$ denote the same for words in the article. Each word is represented by a D dimensional word embedding that was pretrained using GloVe [3]. Using a bidirectional LSTM [9], we first encode the article as: $a_t = \text{biLSTM}_A(a_{t-1}, x_t^A, x_{m-t+1}^A)$, where we initialize the LSTMs with a zero state. The biLSTM consists of 2 LSTMs. In step t one takes in x_t^A and the other one takes in x_{m-t+1}^A. We define the article encoding as $A = [a_1, ..., a_m] \in \mathbb{R}^{2d \times m}$.

Similar to [5], we then initialize a second bidirectional LSTM biLSTM_H with the last state of the first LSTM a_m and extract the headline encoding as $h_t = \text{biLSTM}_H(h_{t-1}, x_t^H, x_{m-t+1}^H)$. We define the headline encoding as $H = [h_1, ..., h_n] \in \mathbb{R}^{2d \times n}$.

The concatenation of the last states of the forward and backward pass of the article LSTM $a_m \in \mathbb{R}^{2d}$ and the headline LSTM $h_n \in \mathbb{R}^{2d}$ encode the local word embeddings of the headline and article of our model. As the articles tend to be quite long, we condition the headlines on the articles and not vice versa in order to avoid gradients vanishing before they reach the first LSTM. In this regard, it also helps to feed in the outputs of the first LSTM directly into the hidden layer.

To extract global features, we follow the baseline method of [1] and include the features described in Table 6 on page 11, which will be denoted as $g \in \mathbb{R}^f$, where f is the number of features. The global and the local features are then concatenated to form a vector that is multiplied by $W \in \mathbb{R}^{4 \times 4d+f}$ in the softmax layer to produce the classification output c of our model as follows:

$$c = \text{softmax}(W[h_n, a_m, g] + b) \in \mathbb{R}^4 \tag{1}$$

$\mathbb{R}^4$ corresponds to the dimension of our labels *agree*, *disagree*, *discuss*, and *unrelated*. Given c we optimize over a cross entropy loss and train the model with the Adam optimization method [10].

4 Experiments

In this section, we evaluate our model compared to the baseline of [1]. We show that our model outperforms it quantitatively in terms of the fake news challenge score and the F1 score. We also

perform an ablation study showing that both local word embeddings and global features are critical for its success.

4.1 Implementation Details

After performing an extensive grid search over all of our hyperparameters as shown in Appendix A Table 5 on page 10, we chose to configure our model as follows. All parameters mentioned in the following, were chosen based on a hyperparameter search.

For our input words, we use 100 dimensional GloVe vector representations that are updated during training and do not remove stop words. We tried to use 50, 100, 200 and 300 dimensional GloVe vectors with and without stop words, but we found that this combination works best. Despite the previously described intuition that one might want to truncate the articles at 700 words to keep as much of the articles as possible, a maximal article length of 200 words works best for our model. Again, we do not truncate the headlines as they are short enough already. Due to the highly imbalanced labels in our dataset, we also balance our dataset by undersampling frequent labels, which helps during training.

For our model, we use LSTMs with 100 hidden units and only one output layer that transforms the concatenated features into 4 classes. We tried using hidden layers, but this resulted in inferior performance. We clip our gradients at 1.0, use a dropout of 0.3, and train with a learning rate of 0.01.

It is also worth noticing that we tried to optimize our model for speed, to be able to iterate fast through different setups. In this regard, we do not use a fixed length for our article and headline inputs. Instead, we sort all headline-article pairs by their article length and bin them into 5 different buckets. In addition, we precompute all global features, and we cleaned and tokenized inputs. In the end, our model is able to reach peak performance in 5 epochs where each epoch takes about 50 seconds to run.

4.2 Results

After setting up the baseline model of [1], we first use a model similar to the model of [5]. First, we encode the headline using a bidirectional LSTM. Then, we encode the article with another bidirectional LSTM conditioned on the output of the headline LSTM. To our surprise, processing the article first and conditioning the headline on the article encoding worked better than vice versa for our dataset. Just by switching the order in which the article and headline are processed, we were able to increase our performance from a 57.8% score and 41.1% mean F1 score to a 65.3% score and 50.2% mean F1 score. Further fine-tuning, such as optimizing the number of hidden units and the dimension of word embeddings, balancing the dataset during training, and truncating the articles as mentioned in the implementation details maxes out the performance of our bidirectional LSTMs at 70.5% score and 51.4% mean F1 score.

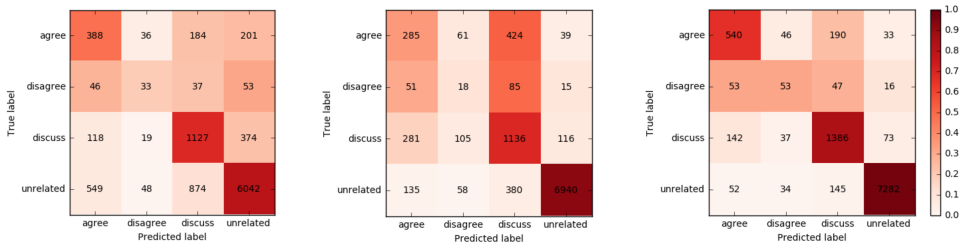
Looking at Table 2 and the confusion matrix in Figure 2a, results show that the conditioned bidirectional LSTM model performs poorly when distinguishing *related* from *unrelated* articles. On the other hand, our previous analysis of the baseline of [1] has shown that its features are especially useful for discriminating *related* from *unrelated* articles, but not the other classes, as can be seen in Table 3. To leverage the baseline's classification accuracy of *related* and *unrelated* articles, we include the features of [1] in our model by concatenating them to the LSTM features before the softmax computation.

As can be seen in Table 2 our final conditioned bidirectional LSTM model with global features outperforms all other baselines and models with an overall score of 87.4% and a mean F1 score of 69.5%. Figure 2a shows that the concatenation of global features with the bidirectional LSTM features effectively reduces the number of false positives for the *unrelated* category compared to using the bidirectional LSTM only. At the same time, the LSTM is now able to better focus on the discrimination of the *agree*, *disagree* and *discuss* categories instead of having to deal with the *related*/*unrelated* discrimination. Yet, most confusions happen in the *agree* and *disagree* categories, which is expected given that these two categories have the lowest number of examples in the dataset, as can be seen in Table 1. As a result, these two categories are biased towards being classified as *discuss* and confusions between *agree* and *disagree* are frequent as well.

Table 2: Performance in stance detection score and F1 score of various models in %

Models	Score	Mean F1	Agree F1	Disagree F1	Discuss F1	Unrelated F1
Baseline [1]	77.7	46.8	19.0	1.1	70.0	97.0
Two-step Baseline	78.4	46.8	18.6	1.1	70.4	97.0
BoW	83.2	66.3	54.5	**39.9**	54.5	95.7
BoW + global features	85.2	65.8	58.7	29.6	76.8	**98.1**
biLSTM	70.5	51.4	40.6	21.6	58.4	85.2
Hidden + global features	76.2	50.5	36.5	8.8	62.0	94.9
biLSTM + global features	**87.4**	**69.5**	**67.7**	31.3	**81.4**	97.6

As a sanity check, we also evaluated a model using only the global features as an input to a network with 3 hidden layers. As expected, the confusion matrix in Figure 2a b) as well as its scores in Table 2 show that this is inferior to our final model. It is also worth noting that the confusion matrix looks similar to the baseline confusion matrix in Figure 4a, which was expected. This confirms that the features presented in baseline [1] are not well-designed for distinguishing between *agree*, *disagree* and *discuss*.



(a) Only biLSTM (b) Only global features (c) All features

Figure 2: Confusion matrices comparing our complete bidirectional LSTM model with global features (c) with only bidirectional LSTM local features (a) and only global features (b)

4.3 Further experiments

In this section, we describe some exploratory architectures that we tested but did not incorporate into our final model.

4.3.1 Two-step classification

With the intuition that the *related/unrelated* classification task is much easier and slightly different than the subsequent *agree/disagree/discuss* classification, we hypothesized that having a two-step classification procedure may produce better results.

We first tested this by implementing the two-step procedure for the baseline model. As reference, our baseline receives a score of 77.74%, which differs from the official baseline due to a different test set. With this new method, the score goes up to 78.43%. However, if we look more closely at the confusion matrix and per label F1 score, we see that the improvement in score is mainly due to $\sim$ 40 *discuss* examples being classified correctly and not an overall, robust improvement in performance. Figure 4 on page 10 shows the confusion matrix for the baseline and two-step baseline. By splitting the classification into two steps, we can also gain insight about the impact of the global features. Specifically, we saw that the baseline classifier was able to get approximately 95% of the *related/unrelated* classifications correct while performance on the second part was much worse, at about 70%. This further supports the effectiveness of these global features for coarse stance discrimination, but not for the finer subtleties in distinguishing between *agree, disagree, and discuss*, a task better suited for the bidirectional LSTM.

Additionally, with the inclusion of global features to the bidirectional LSTM, the two-step method does not improve performance. Originally, the bidirectional LSTM only achieved about 71% in the *related/unrelated* task, but with the addition of global features this jumped to about 97%, which might explain why the two-step method did not perform significantly better than just the bidirectional

LSTM with global features. After additional testing with various architectures that did not lead to improved results, we ultimately decided not to pursue this direction for our final model.

4.3.2 Attention mechanisms

We also tried to integrate different attention mechanisms in our model to further combat vanishing gradients and to make our model more expressive.

The first attention mechanism we tried is described in [8]. This did not improve our results. We also implemented the dynamic coattention mechanism described in [7]. Again, this attention mechanism did not improve our model.

We believe that in the end the combination of global features, truncating the inputs, processing the article first and feeding its representation directly into the hidden layer makes attention obsolete, resulting in no improvement to the performance of our model. It is also possible that richer models simply overfit on the moderately sized stance detection dataset. Indeed, we regularly observed overfitting on the training set when training different kinds of recurrent networks, and we had to strongly regularize our models by introducing dropout and reducing the number of parameters.

4.3.3 Convolutional neural networks for n-grams

Inspired by the success of the baseline with char n-grams and word n-grams [1], we also tried to introduce variously sized word n-grams into our model. We experimented with concatenating word n-grams on top of the word embeddings before feeding them through the RNN. Similarly to [6], we used a convolutional neural network with varying filter sizes to generate n-gram representations. However, n-grams did not improve the performance of our model. Again, the results indicate that introducing more parameters, this time in the form of convolutional filters, can negatively impact the performance of our model on this stance detection dataset.

4.3.4 Bag of Words

Some of our experiments were based on a completely different approach based on bag of words (BoW). Here we describe the most successful model of this kind. A diagram of this model can be seen in Appendix A (Figure 5 on page 11). For word representation, we used a 50 dimensional version of the pre-trained GloVe vectors [3] used in our other models. For each headline-body pair, stopwords are removed from both the headline and the body. The body is split up into sentences and the average word vector is calculated for each sentence. A corresponding vector is calculated for the headline. We then calculate the cosine similarity of the headline vector to each body sentence vector and pick the 3 with the highest similarity. Those vectors as well as the headline vector are then concatenated to create the input vector for our classifier. Optionally we concatenated the global features to the input vector as well. The input vector is then fed into a neural network with a single 100 unit ReLU hidden layer and a softmax output layer.

The BoW model performs surprisingly well given its simplicity and only performs slightly worse than our full model, see Table 2 on the preceding page. A confusion matrix can be seen on Figure 4 on page 10 in Appendix A. The model seems to capture similar information as the global features because adding them only gives a small boost to the performance unlike with the LSTM.

5 Discussion

5.1 Global Features

The baseline with only the global features which are described in Table 6 on page 11 performs surprisingly well compared to the deep learning models. Further adding the global features improves the performance of all our models significantly. To understand why, we analyzed which global features were contributing the most to the baseline. We did this by running the baseline model with a reduced set of features. The results can be seen in Table 3 on the next page. As we can see, the hand features (co-occurrence counts, co-occurrence count without stopwords and ngram/chargram counts) contribute the majority of the performance and the rest only slightly improve the score.

Table 3: Table showing the performance of the baseline scorer with a reduced feature set in %

Features Included	Score	Agree F1	Disagree F1	Discuss F1	Unrelated F1
All features	77.7	19.0	1.1	70.1	97.0
Hand Only	76.8	9.1	1.1	69.1	96.8
↪ Cooccurrence Only	63.2	1.7	1.2	50.1	89.9
↪ Cooccurrence W/O Stopw. Only	74.0	1.0	0.0	64.6	94.9
↪ Countgrams Only	70.2	5.5	0.0	60.2	93.4
Word Overlap Only	62.0	2.4	0.0	50.4	90.3
Hand + Polarity	77.2	14.8	1.1	69.6	96.8
Hand + Refuting	77.6	16.7	1.2	70.4	96.9
Hand + Word Overlap	77.0	9.0	0.0	69.4	96.9
Word Overlap+Polarity+Refuting	61.7	9.0	0.0	49.9	89.9

5.2 Analysis of model performance

Our models predict relevance very well. When they make mistakes, it is usually because the headline either: (1) contains a lot of words that appear in the article even though they are *unrelated* as can be seen in the last example prediction in Appendix B, or (2) when the headline uses different words than the article.

Predicting sentiment is much harder. Our model gets the majority of *agree* and *discuss* classifications right but is not as good at predicting *disagree*. There is often very little information that distinguishes whether a pair would be classified as *agrees* or *discusses*. In the second example in Appendix B, the prediction of individual words like *reportedly* is the only reason that it should be classified as *discusses* instead of *agrees*. We tried to improve this by adding the global *discussion* features.

The model only classifies around 31% of *disagree* pairs correctly. We believe that the limited amount of data for the *disagree* label is the primary reason for this. The dataset only contains 840 *disagreeing* pairs. This can affect the model in several ways. First of all, each headline and body occurs many times and multiple headlines and bodies report on the same thing. For example, the five most common news stories account for about 330 out of the 840 pairs. Out of the 52 pairs in the test set that our model correctly classifies as *disagree*, 18 involve the meteorite story seen in one of the example predictions. If the model had been trained on these examples, it could cause the model to associate the word *meteorite* with *disagree*, which of course does not generalize and could prevent the model from effectively learning features that actually indicate disagreement. Another reason is that we are optimizing for the cross entropy loss and that favors the more common classes. We tried to train on *disagree* examples more frequently to offset this. Overall the score of our classifier (both the competition score and number of correctly classified examples) is only slightly better than we would get by taking every sample that is classified as *disagree* and randomly assigning them to *agree* and *discuss*.

6 Conclusion and Future Work

We have shown that our conditioned bidirectional LSTM with global features outperforms [1] and a BoW model. We have demonstrated that the combination of global features and local word embedding features is better at predicting the stance of headline-article pairs than each of them individually.

In the future, we want to experiment with other recurrent architectures that have proven to be successful in natural language processing. Since the current network performs already almost perfectly on distinguishing *unrelated* from *related* articles, we believe that investigating methods that do particularly well at discriminating positive from negative stances would be most beneficial to improve our current model. Lastly, since news articles contain a lot of named entities that can result in unknown words, a pointer method similar to [11] could help resolve unknown words and better link the headline to the article body.

Acknowledgments

We thank our mentor, Abi See, our TAs, and our instructors for helpful discussions and for the effort they put in to make this class a great experience.

References

[1] D. Pomerleau and D. Rao, "Fake news challenge." `http://www.fakenewschallenge.org/`, 2016.

[2] W. Ferreira and A. Vlachos, "Emergent: a novel data-set for stance classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, 2016.

[3] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[4] S. M. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," *Proceedings of SemEval*, vol. 16, 2016.

[5] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance detection with bidirectional conditional encoding," *arXiv preprint arXiv:1606.05464*, 2016.

[6] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[7] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," *arXiv preprint arXiv:1611.01604*, 2016.

[8] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[11] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

A Additional Tables and Figures

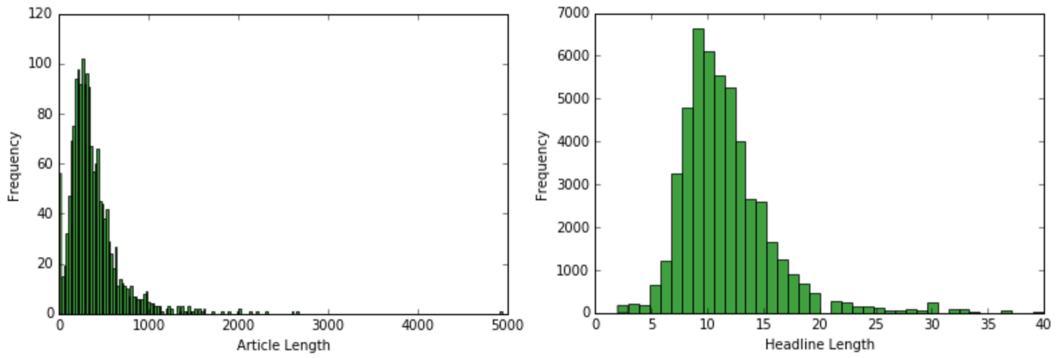
Table 4: Example of *unrelated, agree, disagree, and discuss* article excerpts for a given headline

Headline	**Robert Plant Ripped up $800M Led Zeppelin Reunion Contract**
Unrelated	"... Richard Branson's Virgin Galactic is set to launch SpaceShipTwo today. ..."
Agree	"... Led Zeppelin's Robert Plant turned down £500 MILLION to reform supergroup. ..."
Disagree	"... No, Robert Plant did not rip up an $800 million deal to get Led Zeppelin back together. ..."
Discuss	"... Robert Plant reportedly tore up an $800 million Led Zeppelin reunion deal. ..."

Table 5: Hyperparameter tuning for final model. Varied individual parameters in sequence. Only the most significant hyperparamters are shown. Optimization for [1] score, shown in % on dev set

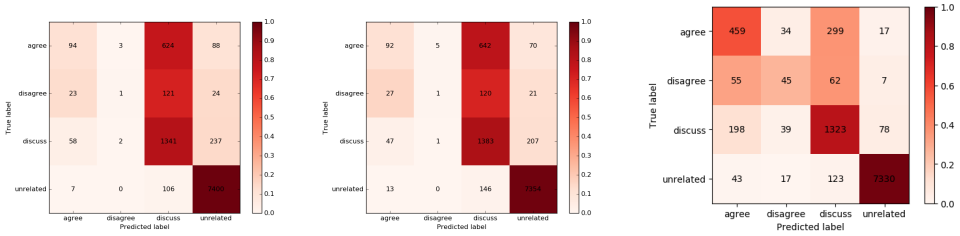
	LSTM hidden units			**embedding dimension**				**truncate length**			**balance data**	
	10	50	**100**	50	**100**	200	300	**200**	400	700	**yes**	no
Score	85.0	86.7	**87.1**	87.6	**87.9**	87.8	87.8	**88.4**	88.2	86.8	**89.4**	87.9

	batch size			**gradient clipping**			**dropout rate**			**learning rate**		
	32	128	**512**	**1.0**	3.0	5.0	0.1	**0.3**	0.5	0.1	**0.01**	0.001
Score	86.5	88.0	**88.8**	**89.2**	88.3	88.9	88.6	**89.6**	88.3	78.9	**88.6**	83.6



(a) Article length distribution (b) Headline length distribution

Figure 3: Distribution of headline and article lengths in the fake news challenge dataset



(a) Baseline [1] (b) Two-step baseline (c) BoW with global features

Figure 4: Confusion matrices comparing the baseline (a) with the two-step baseline (b) and the Bag of Words model (c)

Table 6: A list of global features. All the features except the *discuss* features are used in the baseline

Overlap	$\frac{\lvert\mathbf{B}\cap\mathbf{H}\rvert}{\lvert\mathbf{B}\cup\mathbf{H}\rvert}$, $\mathbf{H}$ is the set of words in the headline, $\mathbf{B}$ is the set of words in the body.
Refuting features	An indicator vector indicating the presence of each of the refuting/polarity words in the headline.
Polarity features	Two numbers, one for the headline, one for the article body. Each is the sum of how often the refuting/polarity words appear in each, modulo 2.
Refuting/polarity words	*fake, fraud, hoax, false, deny, denies, not, despite, nope, doubt, doubts, bogus, debunk, pranks, retract*
Discuss features	An indicator vector indicating the presence of each of the *discuss* words in the body
Discuss words	*according, maybe, reporting, reports, say, says, claim, claims, purportedly, investigating, told, tells, allegedly, validate, verify*
Binary cooccurrence	Two numbers, one counting the sum of how often any word in the headline appears in the entire body, the other only in the first 255 words of the body.
Binary cooccurrence stops	Same as binary cooccurrence but ignores stopwords.
Char grams	A vector specifying the sum of how often character sequences of length 2,4,8,16 in the headline appear in the entire body, the first 100 characters and the first 255 characters of the body.
Word grams	A vector specifying the sum of how often a word sequence of length 2,3,4,5,6 in the headline appears in the body.

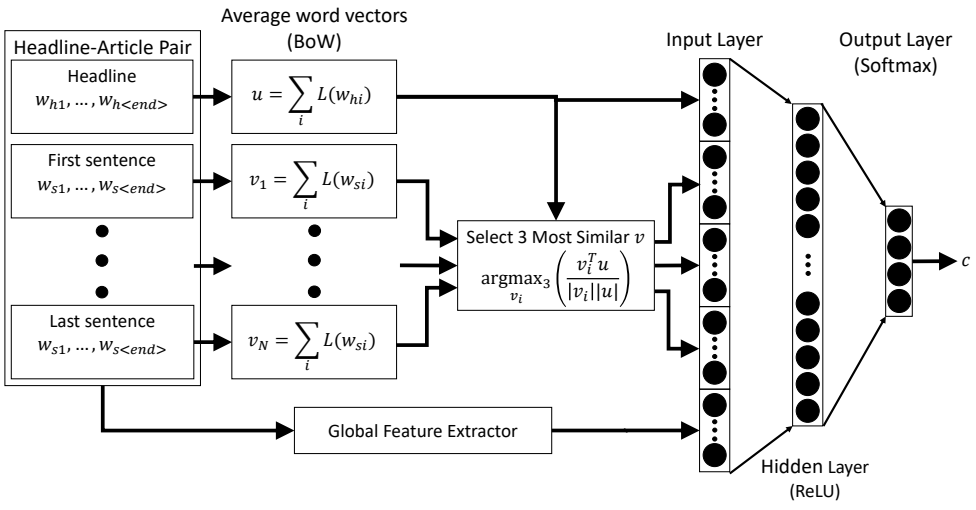


Figure 5: Our Bag of Words with global features model. L is an embedding transformation

B Model Classification Examples

1. Stance (Ground truth/Predicted): *Agree/Agree*

Headline: Israeli-Canadian Woman 'Captured' By ISIS Says She Is 'Safe And Secure'

Body: Gill Rosenberg, a Canadian-Israeli woman, who ISIS had claimed to have abducted, posted on Facebook Monday afternoon that she was safe and secure. . . .

2. Stance (Ground truth/Predicted): *Discuss/Agree*

Headline: In Vogue vs Rats War, the Rats Are Winning

Body: She is rarely seen without her trademark Chanel sunglasses, by day or by night.
So when Anna Wintour stepped out minus her shades on Friday afternoon, all eyes were on her.
The powerful Vogue editor-in-chief gave a steely glare as she left the magazine's new headquarters at One World Trade Center, which has reportedly been overrun with an infestation of rats.
Scroll down for video
Not amused: Anna Wintour did not look at all happy as she left her rat-invested new Vogue office without her trademark sunglasses on Friday
What are you looking at? Anna is said to have had enough of the luxurious One World Trade Center . . .

3. Stance (Ground truth/Predicted): *Unrelated/Unrelated*

Headline: New iOS 8 bug can delete all of your iCloud documents

Body: It's not every night that a pizza delivery driver receives a $2,000 tip.
But that's what happened Thursday to one Ann Arbor driver who delivered a single pizza to a conference room full of more than 200 realtors from Michigan and northern Ohio on Thursday. . . .

4. Stance (Ground truth/Predicted): *Unrelated/Discuss*
Headline: US, UK eye rapper as British-born militant who beheaded journalist James Foley
Body: A British-born U.S. photojournalist held by al-Qaida militants in Yemen has been killed in a failed rescue attempt, his sister has revealed today.

Luke Somers had been held hostage since September 2013 in Yemen's capital Sana'a having moved to the country two years earlier. . . .

5. Stance (Ground truth/Predicted): *Disagree/Disagree*

Headline: Rare meteorite impact causes blast in Nicaragua's capital, Managua

Body: An asteroid about 60 feet in diameter missed hitting Earth Sunday by about 25,000 miles, as predicted by NASAs Near-Earth Object program.
It did not, contrary to some reports Monday, break off a little piece that landed with a bang in Nicaragua. . . .