FNC-1: Stance Detection

MYTH BUSTERS: Jayanth Reddy Regatti, Meghan Day, Dushyanta Dhyani

***Abstract*—In this project, we present our approach to solving the challenge of fake news. We are solving this problem as a part of the Fake News Challenge (FNC) Stage 1. FNC aims to explore how machine learning and natural language processing can be used to identify fake news. Stage 1 of the challenge focuses on classifying the stance of a news article body relative to a headline as $agree$, $disagree$, $discuss$, or $unrelated$. We present three methods for solving this task: a modified baseline with hand-engineered features, a word-embedding based model, and an LSTM with conditional encoding. The accuracy achieved by our best model around 87%, constituting a 8% improvement on the provided baseline implementation. We also present ideas for further refinement of our models and the limitations associated with the problem statement.**

I. Introduction

"The FAKE NEWS media... is not my enemy, it is the enemy of the American People!" - Donald J Trump

Inaccuracies in news reporting have always occurred, but discourse around fake news has crescendoed during the latest U.S. election cycle. Presidential candidates, their supporters, and their opposition all fervently claimed that news articles contradicting their world view or presenting disagreeable facts were fake, and indeed many media agencies published conflicting reports of events. Ongoing investigations by U.S. intelligence agencies are still determining the impact of these allegations and searching for potential involvement by hostile foreign agents.

Fact-checking agencies have served a critical role in identifying false claims in the media but have been unable to keep up with the volume of content posted daily in the new digital age. To supplement these efforts, it would be useful to leverage machine learning in order to automatically identify unreliable or unconfirmed information in news articles. A first step in this process is stance detection, the estimation of the perspective of a piece of text in relation to a particular issue. The rationale behind using stance detection for fake news detection comes from the fact that if multiple credible sources "disagree" with a News headline, the trustworthiness of the headline reduces. This is the challenge posed by Stage 1 of FNC. Specifically, we want to classify the stance of a news article body in relation to a claim or statement asserted by a news article headline.

In this project, we used several approaches to better the provided baseline implementation. We first started out augmenting the provided baseline with features derived from the Paraphrase Database [6] and WordNet [13]. In doing so, we were able to better the baseline performance by a margin of around 2%. In order to make our model more robust, we used word embeddings from Word2Vec [9] and GloVe [12]. After analyzing the performance of the above models, we constructed a two stage hierarchical model using different permutations of the features described above in the two stages. This gave a major boost to our performance and bettered the baseline performance by a comfortable 7%. We also tried out features like distance metrics, techniques like data clipping and deep learning models to improve the performance. These methods bettered the baseline performance by a slight margin but did not improve much on the 7% difference. In the following sections, we describe the related work, task description and our approaches in detail.

II. Related Work

Similar problems have already been solved to specific domains or topics. In the SemEval-2016 Task 6, participants built models to label the stance of tweets in relation to five predetermined targets [10]. The highest scoring team for this task achieved an F-score of 67.82 by building a RNN pre-trained on a distantly-supervised auxiliary corpus of hashtags [15]. This task also included a weakly-supervised task of labeling the stance of tweets on a new target for which no training data was provided. The top team for this task used a two-class subset of the training data with 'favor' and 'against' instances and used a CNN to classify tweets with regard to the new target, achieving an F-score of 56.28 [14].

The difference between the top scores for the supervised and weakly-supervised tasks illustrates how challenging stance detection is on an open domain. Augenstein and Rocktäschel used the same dataset from the SemEval-2016 task to tackle the target-independent stance detection problem. Their approach uses LSTMs to conditionally encode representations of both the targets and tweets dependent on the other and outperforms the state-of-the-art with an F-score of 68.03 [3].

III. Task Description

A. Dataset

The task for Fake News Challenge Stage 1: Stance Detection [1] is to build a classifier to identify the stance of a news article body towards a news headline, with possible stances being $agree$, $disagree$, $discuss$, and $unrelated$. Participants are provided 49,972 labeled article headline and body pairs, which are derived from the Emergent Dataset [5]. Table I shows the distribution of the stance classes.

A sample data point looks like this:
{'Headline': "Guantanamo detainee freed in Bowe Bergdahl swap 'back to terrorism'",
'Body': 'According to a statement by his wife, Henning has been found innocent'
'Stance': 'unrelated'}

B. Evaluation

The data is strongly skewed in favor of the class $unrelated$. A classifier that labeled all articles as $unrelated$ would have

an accuracy of 0.73. For this reason, the challenge evaluates models using a weighted, two-level scoring system as shown in Fig 1.

Unrelated	Discuss	Agree	Disagree
0.731	0.178	0.073	0.016

TABLE I
DISTRIBUTION OF CLASSES

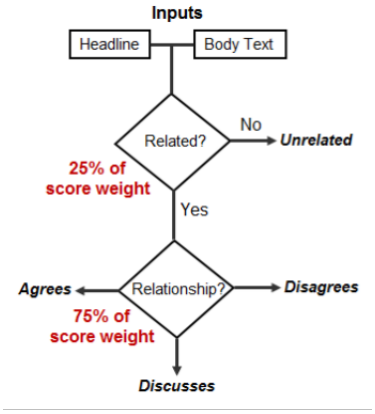

Fig. 1. Evaluation Procedure

C. Baseline

Participants are provided with a strong baseline model to which teams can compare their approaches. This baseline uses hand-engineered features including n-gram co-occurrence counts and indicator features for polarity and refutation to train a GradientBoosting classifier. The weighted score of this baseline is 79.53%.

IV. APPROACH

The baseline implementation does a good job of separating $related$ stances from the rest but performs rather poorly when differentiating between $agree$, $disagree$, and $discuss$. This section presents our approaches to the stance detection problem, both by enhancing the provided baseline system and creating our own word embedding and deep learning systems.

A. Features

1) Modified Baseline (MB): A number of features were added to the baseline to enhance its performance. The existing baseline includes co-occurrences of words, counts of words that appear in both the headline and body. Paraphrase co-occurrence features were added using the Paraphrase Database (PPDB) [6]. These features are counts of the number of times a word in the headline or one of its paraphrases occurs in body.

The list of refuting word feature was also extended with related words from both PPDB and WordNet [13]. Features for disagreeing words and discussing words were added in a similar manner. The results for MB can be found in II.

2) Embeddings: The dataset provided for the challenge is very small compared to the volume of news published every day. Hence, a model trained on such a limited vocabulary may not erform well on a completely unseen dataset. To enable our model to account for new vocabulary in test data, we used word embeddings that are pre-trained over huge datasets. Specifically, we used Google's pre-trained Word2Vec embeddings [9] and Stanford NLP group's GloVe embeddings [12]. We discuss generating embedding features using Word2Vec and the same can be generalized to GloVe.

Plain averaging (PE): Every training instance contains a headline and a body. We calculate the average of the embeddings for every word in the headline that is in the vocabulary, ignoring words which are not present. This returns a 300 dimensional vector representing the headline. The same procedure is carried out for the body. The resulting headline and body vectors are concatenated into a 600 dimensional vector that represents the training sample.

TF-IDF Scaled averaging: An extension of Plain Averaging was to use the TF-IDF scores of the individual words to weight them.

Distance metrics: We also used various distance metrics to measure similarity between the headline and body

- Spatial Similarity Measures like cosine, cityblock, jaccard, canberra, euclidean, minkowski, braycurtis
- More sophisticated Word Mover Distance [8]

3) Body Text Clipping: Based upon the intuition that a human writer would use strong indicators of agreement/disagreement in the initial or last few sentences, we only included the first and last 2 sentences of the body.

B. Models

1) One Stage model: We first started out by trying a four class classifier using RandomForestClassifier [11] that trains on the features discussed above. From Table II we can observe that using modified baseline features performed better than the other two features. The confusion matrices for modified baseline (MB) and embedding features (PE) are shown in Table III and Table IV respectively. Since we used a RandomForestClassifier without a fixed random state, the performance can be expected to lie within ± 1 % of the reported values. Use of GloVe embeddings did not significantly improve the performance for any of our models and hence are omitted from the report.

Features Used	Score
Baseline	79.53
MB	82.01
PE	72.77

TABLE II
ONE STAGE FOUR CLASS CLASSIFIER

2) Hierarchical Model: The performance is higher with the MB features, but a closer look at the confusion matrices reveals an interesting insight into how the distribution of the data affects the performance. Using the MB features, the model does an even better job of separating the $unrelated$ instances from the rest ($agree$, $disagree$ or $discuss$). However, among

	Agree	Disagree	Discuss	Unrelated
Agree	137	0	564	61
Disagree	16	3	129	14
Discuss	77	5	1598	120
Unrelated	3	0	66	6829

TABLE III
CONFUSION MATRIX FOR MB FEATURES

	Agree	Disagree	Discuss	Unrelated
Agree	211	2	171	378
Disagree	7	11	41	103
Discuss	39	1	1241	519
Unrelated	38	0	24	6836

TABLE IV
CONFUSION MATRIX FOR PE FEATURES

those samples which are not classified as $unrelated$, the model is highly biased towards $discuss$. This can be attributed to the uneven distribution of the dataset and the lack of enough features to distinguish the stance. On the other hand, the model using embedding features performed better on classifying the features as $agree$, $disagree$ or $discuss$.

Based on this insight, we constructed a Hierarchical Classifier (HC) consisting of two stages. The first stage is a two-class model ($related$ vs $unrelated$) trained on all the training data, and the second stage is a three-class model ($agree$, $disagree$ and $discuss$) trained only on the samples not labeled as $unrelated$. During testing, only those samples that are classified as $related$ by the classifier in stage 1 are fed to the classifier in stage 2. Multiple variants of the HC were developed using different combinations of the MB and PE features in each stage.

Features in stage 1 → Features in stage 2	Score
MB → PE	86.18
MB + PE → PE	86.60
MB + PE → MB + PE	86.86

TABLE V
HIERARCHICAL CLASSIFIER

	Agree	Disagree	Discuss	Unrelated
Agree	380	2	352	28
Disagree	49	18	86	9
Discuss	115	7	1617	61
Unrelated	40	0	72	6786

TABLE VI
CONFUSION MATRIX FOR MB+PE → MB+PE

The results of the Hierarchical Classifier can be found in Table V. Using a two stage classifier, we were able to negate the bias created by the uneven distribution of the dataset and the performance rose significantly to around 86%. The spread of the stances using MB and PE features in both stages can be seen in Table VI. Table VII discusses other sets of features discussed in the previous section with a two stage classifier discussed in this section.

3) Simplified LSTM Model: Long Short-Term Memory model (LSTM) [7] is best known to handle sequential data overcoming the limitations posed by Recurrent Neural Networks (RNN). Bidirectional Conditional LSTM outperformed the other models in SemEval 2016 Task 6 [3]. Since the problem that we are set to solve is similar (to determine the stance of a piece of text conditioned on another text), we started out to adopt the concept of Bidirectional Conditional LSTM to our problem.

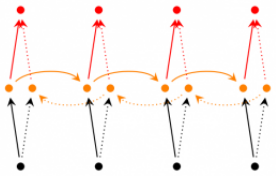

Fig. 2. example of a bidirectional lstm

However, LSTMs require tremendous compute (and memory) for training. With the time and compute constraints that we had, we simplified this model to fit our requirements. In our simplified model, we averaged the word embeddings for the 'Headline' and 'Body' texts separately resulting in two 300 dimensional vectors. We constructed an a bidirectional LSTM model consisting of two time steps, and fed the 'Headline' vector to the first time input and the 'Body' vector to the second time input. Our model looks similar This way, we are able to condition the headline and the body features on one other instead of just concatenating them into a single 600 dimensional vector. To counter the issue of uneven data distribution, this model also involves a two stage classifier similar to the model in the previous section, and the second stage uses the LSTM for classification. The LSTM was trained in tensorflow [2] using a learning rate of 0.005 and 100 hidden layers for 50 epochs. The results of the simplified LSTM are presented in Table VII. However due to the simplification, contextual information in a sequence of words is lost which might have affected the performance of the model.

Features in stage 1 → Features in stage 2	Score
Data Clipping	85.54
Tf-Idf Averaging	86.66
MB → LSTM	82.48

TABLE VII
OTHER FEATURES USED

V. ROAD AHEAD

With strong models using hand-engineered features and embeddings in place, we now aim to improve upon our deep learning models. We also aim to use techniques from closely related task of textual entailment [4]. In the slack group for the challenge, several participants have reported the performance and only a few deep learning models were able to achieve a score better than 85%. With our model already breaching the baseline performance comfortably by a margin of 7-8 %, we aim to extend our work to include more sophisticated models and participate in the FNC-1 contest in June.

VI. REMARKS

The final goal of FNC is to combat fake news. However, it still has a very long way to go since it is often difficult

for humans to differentiate fake news from real news. This problem is not expected to be solved until human level artificial intelligence is solved [1]. The dataset provided as a part of this challenge is a simplification of the real problem at hand and a good performance on this dataset might not ensure a fully functional fake news detector. However, on the brighter side, several institutions have shown interest in combating this problem and this is a great first step in attempting to solve this problem.

VII. ACKNOWLEDGEMENTS

We thank the instructor for providing us with valuable suggestions regarding the project.

REFERENCES

[1] Fake news challenge.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. Stance detection with bidirectional conditional encoding. *arXiv preprint arXiv:1606.05464*, 2016.

[4] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[5] William Ferreira and Andreas Vlachos. Emergent: a novel data-set for stance classification. In *Proceedings of NAACL: Human Language Technologies*. Association for Computational Linguistics, 2016.

[6] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

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

[8] Matt J Kusner, Yu Sun, Nicholas I Kolkin, Kilian Q Weinberger, et al. From word embeddings to document distances. In *ICML*, volume 15, pages 957–966, 2015.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[10] Saif M Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. Semeval-2016 task 6: Detecting stance in tweets. *Proceedings of SemEval*, 16, 2016.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[13] Princeton University. About WordNet, 2010.

[14] Xuqin Liu Wei Chen Wan Wei, Xiao Zhang and Tengjiao Wang. pkudblab at semeval-2016 task 6: A specific convolutional neural network system for effective stance detection. In *Proceedings of the International Workshop on Semantic Evaluation*. SemEval 16.

[15] Guido Zarrella and Amy Marsh. MITRE at semeval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the International Workshop on Semantic Evaluation*. SemEval 16.