Can Siamese Networks help in stance detection?

T.Y.S.S. Santosh
IIT Kharagpur
Kharagpur, West Bengal, India
santoshtyss@gmail.com

Srijan Bansal
IIT Kharagpur
Kharagpur, West Bengal, India
srijanbansal97@gmail.com

Avirup Saha
IIT Kharagpur
Kharagpur, West Bengal, India
saha.avirup@gmail.com

ABSTRACT

An important component of fake news detection is to evaluate the stance, different news sources take towards the assertion. Automatic stance detection, would facilitate the process of fact checking. In this paper, we present our stance detection system which comprises of siamese adaptation of Long Short Term Memory (LSTM) networks augmented with an attention mechanism, as siamese adaptation forces the LSTM to entirely capture the semantic differences during training, rather than supplementing the network with a more complex learner that can help resolve shortcomings in the learned representations. Our experiments on a public benchmark dataset, FakeNewsChallenge (FNC), demonstrate the effectiveness of our approach. It focuses on classifying the stance of a news article body relative to a headline as agree, disagree, discuss, or unrelated.

CCS CONCEPTS

• **Human-centered computing** → *Social media*; *Collaborative and social computing systems and tools*; *Social media*; • **Computing methodologies** → Reasoning about belief and knowledge;

KEYWORDS

stance detection, siamese networks

ACM Reference Format:
T.Y.S.S. Santosh, Srijan Bansal, and Avirup Saha. 2019. Can Siamese Networks help in stance detection? . In *6th ACM IKDD CoDS and 24th COMAD (CoDS-COMAD '19), January 3–5, 2019, Kolkata, India.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3297001.3297047

1 INTRODUCTION

Recent years have seen the proliferation of deceptive information in the web. The rising influence of fake news poses a clear threat to ethical journalism and the future of democracy. Fact-checking agencies have played 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. As manual fact checking is a very tedious task, 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 which aims to identify the relative perspective of a piece of text with respect to a claim, typically modeled using labels such as agree, disagree, discuss and unrelated. It is a crucial building block for a variety of tasks, such as analyzing online debates [18] [16], determining the veracity of rumors on twitter [9] [5] or understanding the argumentative structure of persuasive essays [17]. In this paper, we present our stance detection system which comprises of siamese adaptation of LSTM networks augmented with an attention mechanism. Results of this stance detection system on publicly available FakeNewsChallenge dataset, demonstrate the effectiveness of our approach.

Contribution: We apply a siamese adaptation of LSTM networks augmented with an attention mechanism for stance detection, as siamese adaptation forces the LSTM to entirely capture the semantic differences during training, rather than supplementing the network with a more complex learner that can help resolve shortcomings in the learned representations. Our model which is very feature-light shows performance close to the state of the art achieving FNC score of 0.85.

2 RELATED WORK

Previous works in stance detection mostly considered target-specific stance prediction, whereby the stance of a text entity with respect to a topic or a named entity is determined. Target-specific stance prediction has been performed for tweets [1] [20] and online debates [18] [15] [16]. Such target-specific approaches are based on structural [18], linguistic and lexical features [15] and they jointly model disagreement only and collective stance using probabilistic soft logic [18] or neural models[20] with conditional encoding [1]. Stance prediction in tweets [1] [10] and in online debates [7] [15]is different from that of stance detection in a news article. The FNC stance detection task aim is to classify the stance of a single sentence of a news headline towards a specific claim. In FNC, however, the task is document-level stance detection, which requires the classification of an entire news article relative to a headline. [2] uses a combination of deep convolutional neural networks and gradient-boosted decision trees with lexical features. [6] uses an ensemble of five multi-layer perceptrons (MLP) with six hidden layers each and handcrafted features and used hard voting for prediction. [14] uses a MLP with single hidden layer and TF-IDF as features. [11] uses End-to-End Memory Networks for stance detection .

3 PROPOSED APPROACH

In our approach, we adopt a Siamese architecture [3], in which we create two identical sub-networks. Each sub-network on input generates a fixed representation. Both sub-networks share the same weights, in order to project both the inputs to the same vector space and thus be able to make a meaningful comparison between them.

CoDS-COMAD '19, January 3–5, 2019, Kolkata, India

ACM ISBN 978-1-4503-6207-8/19/01…$15.00
https://doi.org/10.1145/3297001.3297047

We make our representation learning objective, directly reflect the given semantic similarity labels. While the other neural networks utilize complex learners to predict semantic similarity from the sentence representations, we impose stronger demands: namely, a semantically structured representation space should be learned such that simple metrics suffice to capture sentence similarity. This perspective also underlies the siamese architecture for face verification developed by [4], which utilizes symmetric Convolution networks and for learning sentence similarity using symmetric LSTM networks[12].

Each Sub-network has an embedding layer which takes input, tokenizes them into words and convert each word into vector representation. We initialize the weights of the embedding layer using pre-trained word embeddings. A LSTM takes the words of input $[x_1, x_2, ..., x_N]$ where N denotes number of words in the input and produces the word annotations H = $[h_1, h_2, ..., h_N]$ where h_i, is the hidden state of the LSTM at time-step i, which summarises all the information of the input up to x_i. We also experiment with bidirectional LSTM (BiLSTM) where the forward LSTM ($\overrightarrow{f}$) reads the sequence from x_1 to x_N and backward LSTM ($\overleftarrow{f}$)reads the sequence from x_N to x_1. This helps in incorporating the information for the context words on both the directions.

$$\overrightarrow{h_i} = \overrightarrow{f}(x_i), i \in [1, N]$$
$$\overleftarrow{h_i} = \overleftarrow{f}(x_i), i \in [1, N]$$

The annotation of each word x_i is computed by concatenating the hidden states of each from both forward hidden state $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ in case of BiLSTM and annotation for word x_i in case of LSTM will $\overrightarrow{h_i}$ itself. Then we applied a context-aware attention mechanism as in [19] to the annotations produced. An attention mechanism assigns a weight a_i to each word annotation, which reflects its importance. We compute the fixed representation of the input as the weighted sum of all the word annotations using the attention weights.

That is, we first feed the word annotations h_i through a one-layer MLP to get a hidden representation e_i of h_i.

$$e_i = tanh(wh_i + b), i \in [1, N]$$

where w and b denote trainable weights and bias matrices respectively. Then we measure the importance of the word as the similarity of e_i with a word level context vector u and get a normalized importance weight a_i through a softmax function.

$$a_i = \frac{exp({e_i}^T u)}{\sum_i exp({e_i}^T u)}, i \in [1, n]$$

After that, we compute the output vector o as a weighted sum of the word annotations based on the weights.

$$o = \sum_i a_i h_i, i \in [1, N]$$

The word context vector u is randomly initialized and jointly learned during the training process.

For a given pair of input, our approach applies a pre-defined similarity function to both the representations obtained from each sub-network. Then output is fed to a four single neuron layer, that performs 4 class classification.

Similarities in the representation space are subsequently used to infer the sentences underlying semantic similarity. Thus, the sole error signal backpropagated during training stems from the similarity between input representations and how this predicted label deviates from the ground truth label.

We here experiment with two similarity functions.One being the inverse exponential of the Manhattan distance (M) as suggested in [12] and other being Cosine distance(C).

$$M(o_1, o_2) = exp(-||o_1 - o_2||_1)$$

$$C(o_1, o_2) = \frac{o_1^T o_2}{||o_1||_2 ||o_2||_2}$$

where o_1 and o_2 are output representations obtained from each sub-network. This forces the LSTM to entirely capture the semantic differences during training, rather than supplementing networks with a more complex learner that can help resolve shortcomings in the learned representations. The overall approach is shown in Figure 1.

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

Table 1: claim and text snippets from document bodies with respective stances from the FNC dataset

4 EXPERIMENTS

4.1 Dataset

We use the dataset provided by the FakeNewsChallenge, where each example consists of a claim-document pair with the following labels between them: agree (AGR) when the document agrees with the claim, disagree (DSA) when the document disagrees with the claim, discuss (DSC) when the document discusses the same topic as the claim, but does not take a stance with respect to the claim and unrelated (UNR) when the document discusses a different topic than the topic of the claim. The data includes a total of 49,972 claim-document pairs. Table 1 shows a claim and text snippets from document bodies with respective stances from the FNC dataset. Table 2 indicates dataset statistics and label distribution for the FNC dataset.

4.2 Evaluation Metrics

We analyze the performance of our models using the scoring function defined by FNC. The function evaluates each prediction in a two step process: 1) Correctly classify headline and body text as related (agree, disagree, and discuss) or unrelated: 25% score weighting; 2) Correctly classify related pairs as agrees, disagrees,

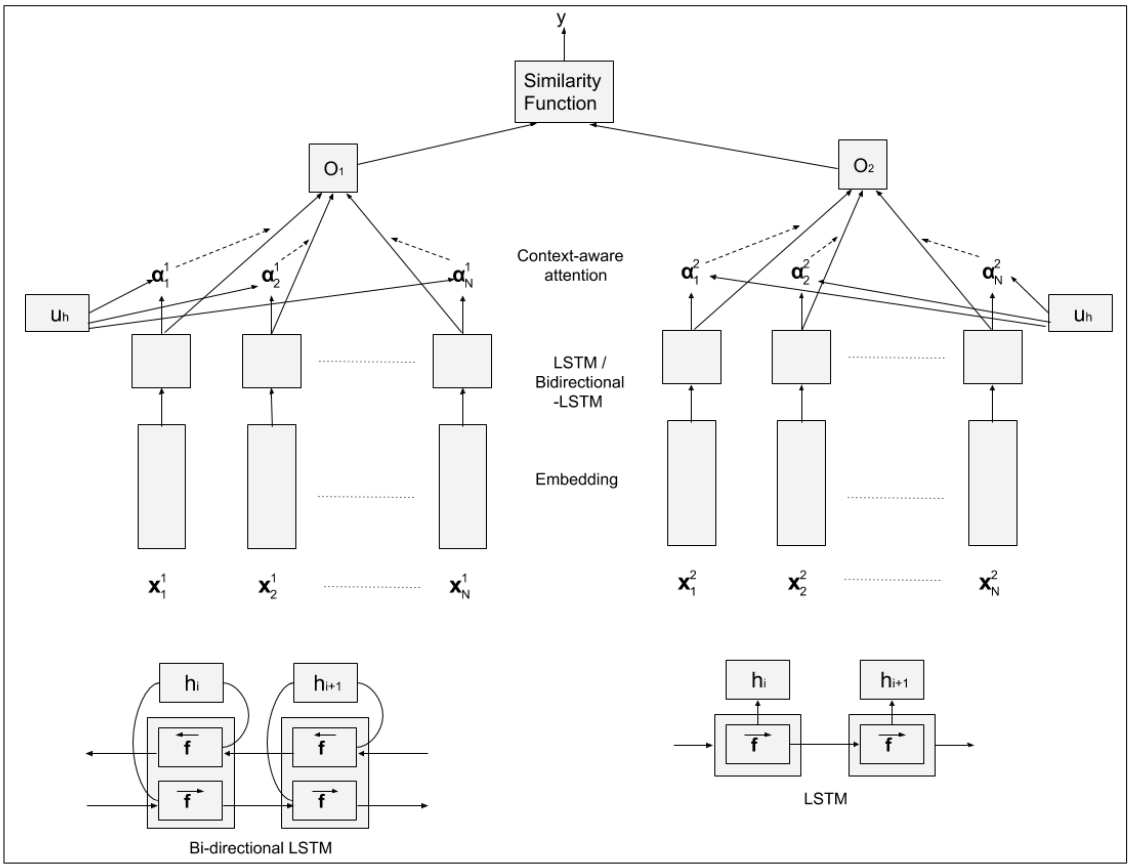

Figure 1: Siamese adaptation of LSTM augmented with attention

claims	2587
documents	2587
claim-document pairs	49972
AGR	7.4%
DSA	2.0%
DSC	17.7%
UNR	72.8%

Table 2: dataset statistics and label distribution for the FNC dataset

or discusses: 75% score weighting. We also report performance of models using Accuracy, the number of correctly classified examples divided by the total number of examples and Macro-F1, where we calculate F1-score for each class, and then we average across all classes.

4.3 Experimental Setup

Baselines: For baseline, we consider FNC top 3 ranked systems. *CNN-GDBT* [2] uses a weighted average model of a deep convolutional neural network(CNN) and a gradient-boosted decision trees model(GBDT). CNN uses pre-trained word2vec embeddings passed through several convolutional layers followed by three fully-connected and a final softmax layer for classification. GBDT is based on word count, TF-IDF, sentiment, and singular-value decomposition features in combination with the word2vec embeddings. *MLP-6* [6] uses an MLP with six hidden layers. They use unigrams, the cosine similarity of word embeddings of nouns and verbs between headline and document tokens, and topic models based on non-negative matrix factorization, latent Dirichlet allocation, and latent semantic indexing as the features. *MLP* [14] uses an MLP with only a single hidden layer. They use term frequency vectors of unigrams of the 5,000 most frequent words for the headlines and the documents, and the cosine similarity between the TF-IDF vectors of the headline and document as feature vector.

Proposed Approach We experiment with 4 variants of proposed

Systems	FNC Score	Macro-F1	Accuracy
CNN-GDBT [2]	0.820	0.582	0.878
MLP-6 [6]	0.819	**0.604**	0.866
MLP [14]	0.817	0.583	0.878
LSTM with inverse exponential of Manhattan distance	0.849	0.590	0.886
Bi-directional LSTM with inverse exponential of Manhattan distance	**0.852**	0.587	**0.899**
LSTM with Cosine Distance	0.796	0.567	0.827
Bi-Directional LSTM with Cosine Distance	0.809	0.550	0.839

Table 3: Results on Test Data

architechure in section 3. We experiment with two similarity functions such as cosine distance and Manhattan distance to the representations obtained from each sub-network. We perform experiments with both LSTM and bidirectional LSTM. We use 100-dimensional word embeddings from GloVe [13], which were trained on two billion tweets. We use Adam as an optimizer and categorical cross-entropy as a loss. We use 100-dimensional units for the LSTM embeddings.

5 RESULTS

Table 3 reports the performance of baseline and the proposed models on the test dataset. Bi-directional LSTM with inverse exponential of Manhattan distance gives the best FNC-score and Accuracy. We observe that cosine distance performs poorly compared to inverse exponential of Manhattan distance. As [4] pointed out, using l_2 norm rather than l_1 norm in the similarity function can lead to undesirable plateaus in the overall objective function. This is because during early stages of training, l_2 based model is unable to correct errors where it erroneously believes semantically different sentences to be nearly identical due to vanishing gradients of the euclidean distance.

6 CONCLUSION

In this paper we presented the approach of siamese adaptation of LSTM networks with attention mechanism for stance detection task. We have experimented with FNC dataset to demonstrate the effectiveness of this approach for stance detection. In future work, we plan to extend this approach with different similarity functions, and other datasets for stance detection. We wish to experiment with Recurrent Entity Network [8] which is equipped with a dynamic long-term memory which allows it to maintain and update a representation of the state of the world as it receives new data. For stance detection tasks, we believe that it can also be helpful in extracting snippets of evidence that can be used to reason about the factuality of the target claim. We would wish to explore this approach in future.

REFERENCES

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

[2] Sean Baird, Doug Sibley, and Yuxi Pan. 2017. Talos targets disinformation with fake news challenge victory.

[3] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*. 737–744.

[4] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1. IEEE, 539–546.

[5] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. *arXiv preprint arXiv:1704.05972* (2017).

[6] Andreas Hanselowski, PVS Avinesh, Benjamin Schiller, and Felix Caspelherr. 2017. âĂIJTeam athene on the fake news challenge.

[7] Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. 1348–1356.

[8] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2016. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969* (2016).

[9] Michal Lukasik, PK Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. 2016. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2. 393–398.

[10] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 31–41.

[11] Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. 2018. Automatic Stance Detection Using End-to-End Memory Networks. *arXiv preprint arXiv:1804.07581* (2018).

[12] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity.. In *AAAI*, Vol. 16. 2786–2792.

[13] 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)*. 1532–1543.

[14] 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 preprint arXiv:1707.03264* (2017).

[15] Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics, 116–124.

[16] Dhanya Sridhar, James Foulds, Bert Huang, Lise Getoor, and Marilyn Walker. 2015. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 116–125.

[17] Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics* 43, 3 (2017), 619–659.

[18] Marilyn A Walker, Pranav Anand, Robert Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics, 592–596.

[19] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1480–1489.

[20] Guido Zarrella and Amy Marsh. 2016. MITRE at semeval-2016 task 6: Transfer learning for stance detection. *arXiv preprint arXiv:1606.03784* (2016).