Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 1646–1653

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

Stance-In-Depth
Deep Neural Approach to Stance Classification

Gayathri Rajendran [a,c,d], Bhadrachalam Chitturi [a,c,f], Prabaharan Poornachandran [b,c,e]

[a] Dept. of Computer Science and Engineering, Amrita School of Engineering
[b] Amrita Center for Cyber Security & Networks,
[c] Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India

Abstract

Understanding the user intention from text is a problem of growing interest. The social media like Twitter, Facebook etc. extract user intention to analyze the behaviour of a user which in turn is employed for bot recognition, satire detection, fake news detection etc.. The process of identifying stance of a user from the text is called stance detection. This article compares the headline and body pair of a news article and classifies the pair as related or unrelated. The related pair is further classified into agree, disagree, discuss. We call related as detailed classification and unrelated as broad classification. We employ deep neural nets for feature extraction and stance classification. RNN models and its extensions showed significant variations in the classification of detailed class. Bidirectional LSTM model achieved the best accuracy for broad as well as detailed classification.

Keywords: Stance Classification , deep learning, RNN , LSTM , Gated Recurrent Unit (GRU), Bidirectional , word2vec.

1. Introduction

The prevalence of social media and the rapid increase in its users has been increasing the prominence of text analysis. The posts, tweets etc. generally referred as text, posted in social media by users are analyzed in detail to understand the current trends or to get the mass opinion. The results of such analysis have been found to be inaccurate due to various factors such as fake news, satires etc. In order to overcome such inaccuracies, the study of

Corresponding author: gayathrirajendran010@gmail.com

10.1016/j.procs.2018.05.132

text, i.e. NLP, has become a widely studied independent branch of machine learning. The textual study is employed in many fields like spam detection, bot recognition, recommender system, satire detection, fake news detection etc.. The textual study comprises of understanding the textual structure, opinion mining, sentiment extraction, reputation mining, stance detection and classification. Stance detection or classification problem is an interesting wing of textual analysis where the stance of user with respect to another text of another user is extracted and analyzed. It helps in prediction of user behavior or other characteristics like gender, location etc.. Such problems are applicable to debate forums as well. In this paper we deal with comparison of two pieces of text i.e the headline and the body. An example is demonstrated in Table 1. The analysis is used to classify them into:

- Unrelated and Related Classes (Broad classification)
- Related Class is broken down into Agree, Disagree and Discuss Classes (Detailed classification)

Table 1. An example of the 4-class classification given headline and body (two pieces of text).

Body	The Times of India reports that on New Year Eve, an airline passenger heading to US was asked to leave from a plane because he engaged on a fight with workers, who wished him a Happy New Year.
Headline	**Class**
Passenger headed to US rages over 'Happy New Year,' removed from plane.	**Agree**
The Times of India reports about passenger removed from plane to US due to his aggressive behavior.	**Discuss**
Passenger headed to US rages over 'Happy New Year,' were asked to remain calm and continue the journey.	**Disagree**
Prime Minister, Narendra Modi visits Dallas today.	**Unrelated**

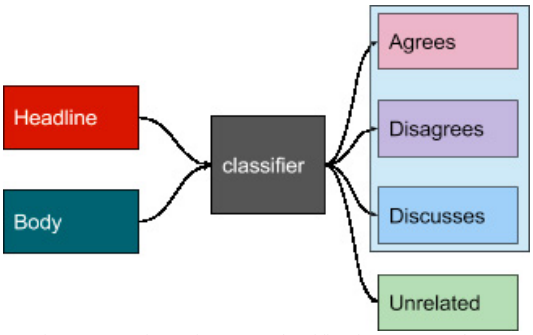


Fig. 1. Overview of Stance Classification Approach

The literature shows that several models have been used for the stance detection problem. Probabilistic [17], markov random process [15], heuristics approach [16], neural network [13] [4] [12] [14] [18] [19] are some of the models that were employed. Probabilistic Soft Logic (PSL), a tool for collective inference in relational data, was employed in [17]. They have created a network graph based on topics, discussion involved under each topic, posts which group together to form one discussion and author connected to each posts with agreement and disagreement links. The graph is traversed as per certain heuristics to predict the stance. Another work [16] compares the rumoured claims and its associated news articles and estimates their veracity using logistic regression classifier with L_1 regularization using two types of features: one extracted from news and other combination of headline and news. [15] work focuses on collective classification of stances on Twitter data, using hinge-loss Markov random fields given graph and relationship of all posts and its associated users. They have predicted the stance of the training tweets using HL-MRF and used the labeled instances as training for a linear text classifier. A deep learning CNN model [5] was developed for stance classification on facebook and twitter data using three features: user associated to posts, topic mentioned in posts and comments received for the posts. They have tested the model for English as well as Chinese languages. [12] is another deep learning work which creates a model with combination of RNN and LSTM and uses target attention model to focus on specific targets. Using target attention models have increased focus on target, based on which stance classification is performed. They have applied the model on English and Chinese languages. [13] used convolutional neural network for detecting stance of tweets and extended the work to rumour verification. They have improved the accuracy by using LOO (Leave One Out) performance validation testing. [14] uses bidirectional LSTM with features like global, co-occurrence, refutation and polarity which complement local

features of textual data. Deep neural network is an extremely expressive non-linear statistical model that can learn highly complex vector-to-vector mappings. Due to its ease of use and advancement of computational power it is gaining popularity. The basic feed forward network is rarely used for textual problem as they lack inter-connectivity between adjacent nodes to understand and process the text. The most popular is recurrent neural network, RNN, where adjacent nodes are interconnected to each other and have self loops as well. The vanilla RNN when applied to text fails to address the long term dependencies. Such problems are overcome by LSTM, GRU RNNs where a neuron is more complex. We compare different RNNs for stance classification. We analyze the impact of the number of layers and we measure the training time required for each of the models. Our model is scalable.

Section 2 presents an overview, framework details, dataset description and experimental results. Section 3 states the conclusions and future work.

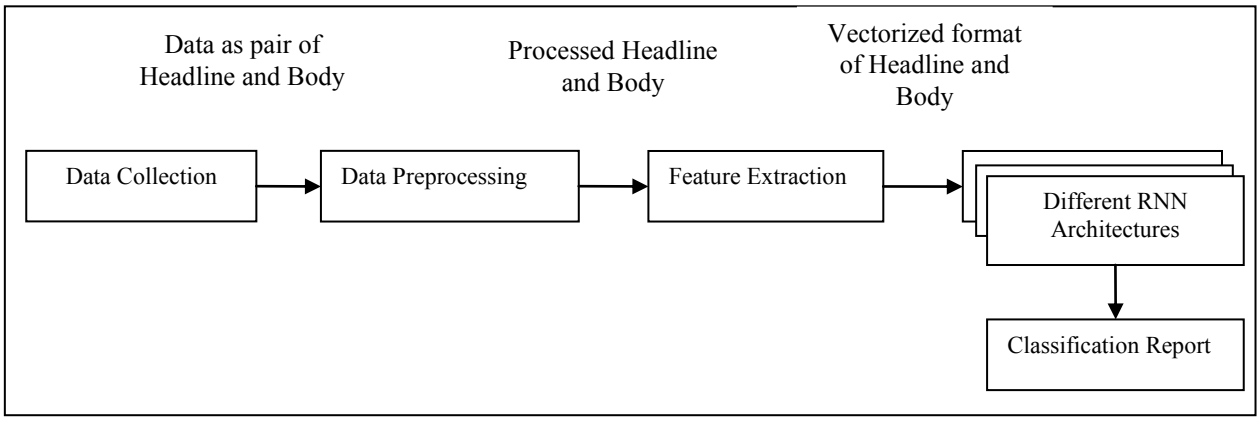


Fig. 2. Summarization of Stance Classification Approach

2. An Overview of Stance Classification Approach

2.1. *Data Collection and Preprocessing*

Data for the research is from news which is in the format of headline and its associated news, like dataset in [8]. Multi-instance of news specific crawlers were deployed for news media like NDTV, CNN etc.. The crawled data is organized in the form of (headline, body) pairs and each headline was associated with multiple bodies (one to many mapping) and the correct relationship between the pairs is labelled. The collected data is in csv format which is initially divided into two sets: training and validation sets in the ratio 7:3 respectively. Both datasets are preprocessed using Python libraries pandas and numpy for tokenization, stopwords and removal. The word embeddings are learned by the word2vec library [2] from gensim model [7] and applied on headlines and body. We used skip gram model of word2vec which helps in word prediction given the context. The feature extraction was performed with word2vec. When headlines and body are vectorized the semantic relationship should be maintained which is well captured using word2vec. [1] uses five different datasets and analyses the semantic relatedness of Glove count model [3] and word2vec predictive model and concluded that word2vec is superior. We tried vectors with dimensions [70 ...300] and chose a value of 200 which yielded best accuracy avoiding over fitting.

Table 2. Statistics of (Headline,Body) pair dataset

Class	Training Dataset	Validation Dataset
Agree	10875	1903
Discuss	7670	4464
Disagree	6589	697
Unrelated	34163	18349

2.2. *Model*

2.2.1. *Long Short-Term Dependency (LSTM)*

The Long Short-Term Dependency [6] (LSTM) is a RNN variant which is capable of storing long term dependencies. A single recurrent unit is a complex combination of three gates passing information through cell state across units and output of each cell is a hidden state. The gates play the role of trimming the data. The initial forget gate identifies the data that is to be removed from the sequence, later, update gate identifies the data that is to be retained and finally output gate decides what data to be passed on through cell state to next unit.

$$f_t = \sigma(W.[h_{(t-1)}, x_t] + b_1) \qquad i_t = \sigma(W.[h_{(t-1)}, x_t] + b_2)$$

$$C1_t = tanh(W.[h_{(t-1)}, x_t] + b_3) \qquad o_t = \sigma(W.[h_{(t-1)}, x_t] + b_4)$$

$$C_t = f_t * C_{(t-1)} + i_t * C1_t \qquad h_t = o_t * tanh(C_t)$$

2.2.2. *Gated Recurrent Unit (GRU)*

A variant of LSTM is Gated Recurrent Unit [9] (GRU) model is a RNN which is also capable of storing long term dependencies. A single recurrent unit is a complex combination of two gates passing entire hidden state of one unit to the next. The initial reset gate identifies the data to be removed from the sequence, and, update gate identifies the data to be kept and passes the entire information to the next unit.

$$z_t = \sigma(W.[h_{(t-1)}, x_t] + b_1 \qquad r_t = \sigma(W.[h_{(t-1)}, x_t] + b_2$$

$$h1_t = tanh(W.[r_t * h_{(t-1)}, x_t]) \qquad h_t = (1 - z_t) * h_{(t-1)} + z_t * h1_t$$

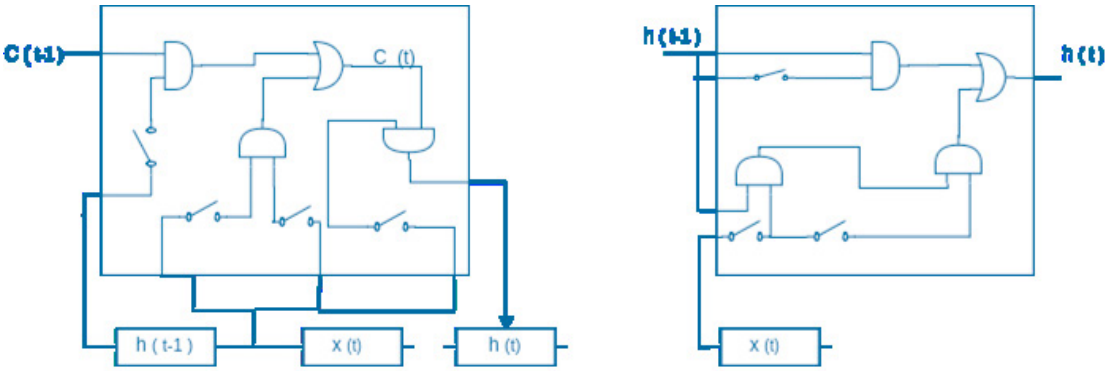


Fig. 3. (a) Single Cell of Long Short-Term Memory (b) Single Cell of Gated Recurrent Unit.

2.3. *Implementation & Results*

The vectorized combination of headline and body, each of dimension 200, is fed to different deep learning models that are listed below.

1. Bidirectional LSTM [6] model as well as plain LSTM model.
2. Bidirectional Gated Recurrent Unit [9] model as well as plain GRU model.
3. Simple RNN model.
4. Combination of Bidirectional GRU and Bidirectional LSTM as well as plain combination.
5. Multi-layered Perceptron (MLP).

All the different deep learning models were tested in Tensorflow with Graphics Processing Unit (GPU). The models are trained in NVIDIA Quadro P5000 GPGPU. We experimented with different set of parameters like batch size, number of layers, memory blocks, and vector dimension. We experiment batch sizes ranging from 5 to 100 and a batch size of 32 resulted in the best accuracy. The number of hidden layers varied from 1 to 8. The best number of hidden layers for each of the models is stated below: for LSTM and Bidirectional LSTM and simple RNN: 4, GRU and Bidirectional GRU: 6 and MLP: 20. The memory blocks tested are: 32, 64, and 128, a block size of 64 was chosen yielded the best accuracy. In word2vec model the vector dimensions of 100, 150, 200, 250, 300 and a few random values between 100 and 300 were tried. The dimension of 200 gave the best accuracy. These are the optimal parameters chosen for all models. The dropout rate is chosen to be 0.1.The activation function used was softmax as it is multi-class classification problem.

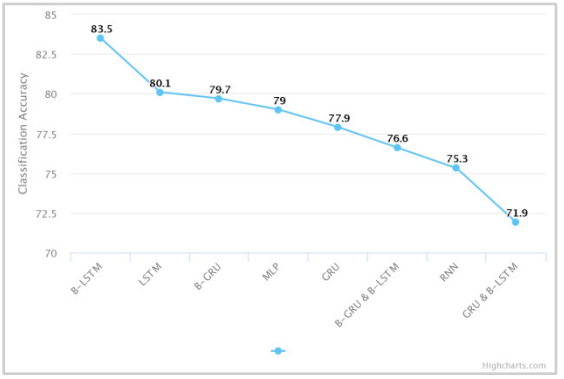
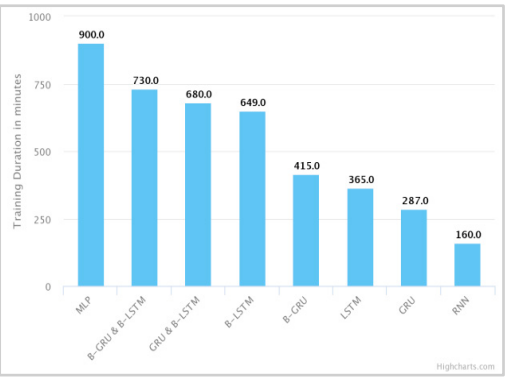


Fig. 4. (a) Classification Accuracy Vs Different Model



(b) Training Duration Vs Different Model.

The best accuracy of 83.5% was obtained by Bidirectional LSTM model. The confusion matrix for this model is depicted in Fig. 5. The Bidirectional approach is better than plain version for LSTM, GRU and a combined model that employs GRU for feature extraction and LSTM for classification. The LSTM and Bidirectional LSTM are found to be superior based on accuracies than the rest of the models. The training time taken was minimum for GRU. RNN [10] [11] which has a simple cell structure yielded the worst accuracy. Bidirectional GRU approximately takes half the time of Bidirectional LSTM, hence 4 layers of Bidirectional LSTM is equivalent to more than 6 layers of Bidirectional GRU with respect to training time but former outperforms the latter with respect to accuracy. Similar comparison applies to LSTM and plain GRU. The reduced time of GRU can be attributed to lesser number of gates (reset and update gates) and one cell variable (hidden state) whereas LSTM has three gates: forget,update and output and two cell variables (cell state and hidden state). The increased accuracy of multi-layered perceptron (MLP) is due to rigorous training of extensive neural architecture. It requires significantly more time when compared to LSTM or GRU or RNN models. The neural architecture of MLP contained 400 input neurons (headline and body word embedding) which pass through 20 hidden layer each with 400 neurons and has 4 output neurons corresponding to 4 classes.

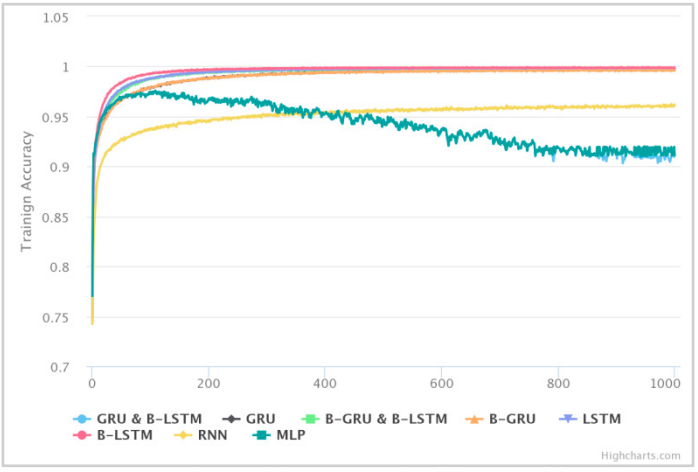


Fig.5. Training Accuracy for different models.

CONFUSION MATRIX:

	agree	disagree	discuss	unrelated
agree	862	21	538	482
disagree	206	10	198	283
discuss	725	34	2577	1128
unrelated	201	16	359	17773

ACCURACY: 0.835

Fig. 6. Confusion Matrix of Bidirectional LSTM Model.

Table 3. Classification Accuracy, Weighted Score, F-Score of different deep learning models

Model [Default : Vector Size : 200, Memory Block : 64 Hidden Layer Count : 4, Batch size : 32]	**Classificat-ion Accuracy**	**Weighted Score**	**F-Score Agree Class**	**F-Score Disagree Class**	**F-Score Discuss Class**	**F-Score Unrelated Class**
RNN	0.753	0.739	0.241	0.055	0.475	0.873
GRU (Hidden Layer : 4)	0.733	0.734	0.201	0.018	0.409	0.799
GRU (Hidden Layer : 8)	0.738	0.732	0.210	0.021	0.422	0.801
GRU (Batch Size : 8, Memory Block 128)	0.739	0.736	0.233	0.031	0.487	0.821
GRU (Hidden Layer : 6)	0.747	0.724	0.355	0.030	0.421	0.875
GRU & B-LSTM	0.719	0.744	0.349	0.167	0.447	0.888
B-GRU & B-LSTM	0.766	0.765	0.263	0.047	0.575	0.884
B-GRU (Hidden Layer : 6, Vector size - 150)	0.757	0.747	0.274	0.026	0.574	0.879
B-GRU (Hidden Layer : 4)	0.751	0.712	0.343	0.061	0.597	0.274
B-GRU (Batch size : 8, Memory Block 128)	0.757	0.747	0.228	0.028	0.499	0.907
B-GRU (Hidden Layer : 6)	0.797	0.773	0.353	0.12	0.588	0.917
MLP	0.79	0.593	0.339	0.0140	0.475	0.885
LSTM (Hidden Layer:2,Batch size:8,Mem Block:128)	0.798	0.743	0.345	0.163	0.445	0.901
LSTM	0.801	0.777	0.415	0.156	0.539	0.913
B-LSTM (Hidden Layer : 8)	0.794	0.746	0.275	0.021	0.571	0.899
B-LSTM	**0.835**	**0.775**	**0.442**	**0.026**	**0.633**	**0.935**

2.4. *Scoring mechanism*

A weighted scoring scheme with respect to the relation obtained is adapted as depicted in figure.

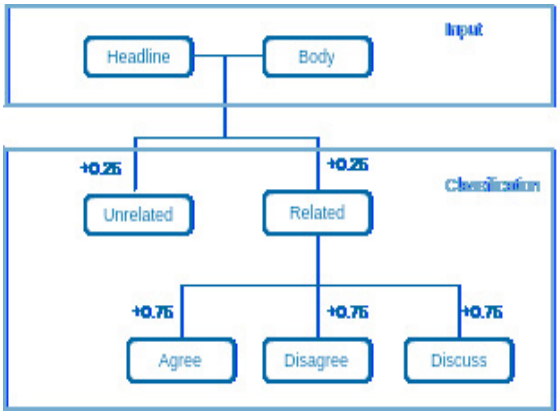


Fig. 7. Weighted Scoring mechanism with weight distribution for each class

More weightage is given to identifying agree, disagree, and discuss relationships (sub-classification) and less to "related" and "unrelated" (main classification). This scheme is morestricter than F score. For correct main classification a score of 0.25 is awarded. In case of "related" the correct sub-classification yields an additional score of 0.75. The weighted score is detailed as :

1. +0.25 for each correct unrelated or related relationship.
2. +1.0 for each correct sub-classification when related.

We have calculated the weighted score for each model to give more importance to detailed classification rather than broader classification. The weighted score for MLP is the least due to its inefficiency to catch inter-connectivity between adjacent nodes.

3. Conclusion and Future Works

Bidirectional LSTM architecture yielded the best results on stance classification and owing to its relatively efficient training process it is better suited for large datasets compared to primitive models. Stance classification is an important domain in NLP and is related to subjectivity analysis and argument mining. We can extend the work on stance classification with extra features like user or topic or sentiment based features. Employing linguistic features in deep learning models is likely to improve the accuracy. Future experiments may employ variants of activation function and its impact. Stance classification can be considered as a key sub problem for larger problems like Fake News Detection etc.. Thus, a good solution to this problem provides a potential solution for problems in several domains.

Acknowledgements

Authors thank Amrita Vishwa Vidyapeetham, Amritapuri, for the infrastructure support to conduct this research.

References

[1] M. Baroni, G. Dinu, and G. Kruszewski, “Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors.” in ACL (1), 2014, pp. 238–247.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, 2013.

[3] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[4] K. Miller and A. Oswalt, "Fake news headline classification using neural networks with attention."

[5] W. Chen and L. Ku, "UTCNN: a deep learning model of stance classification on social media text," in COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, 2016, pp. 1635–1645. [Online]. Available: http://aclweb.org/anthology/C/C16/C16-1154.pdf

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[7] R.Rehurek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.Valletta, Malta: ELRA, May2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[8] Fake news challenge website. [Online]. Available: https://fakenewschallenge.org

[9] J. Chung, C . Gulc ¸ehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," CoRR, vol.abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[10] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 2015, pp. 2342–2350.

[11] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding recurrent networks," arXiv preprint arXiv:1506.02078, 2015.

[12] Y. H. L. G. Jiachen Du, Ruifeng Xu, "Stance classification with target-specific neural attention," in Proceedings of the Twenty-Sixth Interna- tional Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 3988–3994. [Online]. Available: https://doi.org/10.24963/ijcai.2017/557

[13] Y.-C. Chen, Z.-Y. Liu, and H.-Y. Kao, "Ikm at semeval-2017 task 8: Convolutional neural networks for stance detection and rumor verification," in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics, 2017, pp. 465–469. [Online]. Available: http://www.aclweb.org/anthology/S17-2081

[14] D. Mrowca, E. Wang, and A. Kosson, "Stance detection for fake news identification."

[15] J. Ebrahimi, D. Dou, and D. Lowd, "Weakly supervised tweet stance classification by relational bootstrapping," in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1012–1017.

[16] 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 , 2016, pp. 1163–1168.

[17] D. Sridhar, L. Getoor, and M. Walker, "Collective stance classification of posts in online debate forums," in Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media , 2014, pp. 109–11

[18] G. Rajendran, P. Poornachandran, and B. Chitturi, "Deep learning model on stance classification," in Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on.IEEE,2017, pp. 2407–2409.

[19] R.Vinayakumar, S.SachinKumar, B.Premjith, P.Prabaharan, and K.Soman, "Deep stance and gender detection in tweets on catalan independence@ibereval2017,"vol.1881,2017,pp.222–229,citedBy0.[On-line].Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85027881270partnerID=40md5=2742a7b714addedfb3c38de3d14e7918