Report

Saanika Gupta[*]
Department of Computer Science and Engineering
Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur (IIIT-NR)

1 Libraries Used

- Keras 2.2.4 (1)
- Tensorflow 1.14.0 (2)
- Scikit-Learn 0.21.3 (3)
- NumPy 1.16.4 (4)
- Pandas 0.24.2 (5)
- NLTK 3.2.5 (6)

2 Methodology

2.1 Data Cleaning and Preprocessing

The data seemed raw which caused issues in reading the *.tsv or the converted *.xlsx file. This had to be corrected manually. Missing values in the columns were replaced. Statement was made free from numbers and special characters (like @,,etc.) and the words were stemmed. The attributes subject, speaker, job, state and party were converted to lower case. Finally, the corpus obtained from the above steps was converted into bag-of-words. Columns 'barely_true_c', 'false_c', 'half_true_c', 'mostly_true_c' and 'pants_on_fire_c' are selected and without normalization, stacked horizontally with bag-of-words. Normalization wasn't done as the original values are meaningful. For example, if barely_true_c = 10, false_c = 30, half_true_c = 5, mostly_true_c = 3 and pants_on_fire_c = 25 for a speaker. Just by seeing these numbers, we can say that speaker is more likely to state a false news. Normalizing these values between 0 and 1 would result in losing this information.

For binary classification, some of the label values were changed.

- 'barely-true' was changed to 'False'.
- 'pants-fire' was changed to 'False'.
- 'half-true' was changed to 'True'.
- 'mostly-true' was changed to 'True'.

2.2 Artificial Neural Network

Simple Feedforward Neural Network with one hidden layer is implemented for both the tasks. The bag-of-words (BoW) is extracted from statement, subject, speaker, job, state and party. It is then stacked horizontally with the credit history count (including the current statement). This constitutes the input to the ANN. RMSprop optimizer and categorical_crossentropy loss are used for both tasks. Softmax function is used in the final layer and two dropout layers are added in both the tasks.

[*]Third Year Undergraduate Student in CSE Branch

Table 1: Parameters of ANN

Model Parameters	
Optimiser	RMSProp
Loss	Categorical Cross Entropy
Training Parameters	
Batch Size	500
Learning Rate	0.001

3 Different ideas tried

- Random Forest Classification was giving 26% accuracy in Six-way classification.
- Different learning rates were giving poor results.
- RMSprop optimizer outperformed SGD and Adam.
- Adding another hidden layer decreased the accuracy in Six-way classification.
- Normalization of the columns 'barely_true_c', 'false_c', 'half_true_c', 'mostly_true_c' and 'pants_on_fire_c' gave really poor results.
- Tried different values for max_features parameter in the bag-of-words.

4 Experiments

The LIAR-PLUS (7) dataset was used in a number of ANNs with different hyper-parameters such as different optimisers (ADAM and SGD), different number of hidden layers, different learning rates. The experiments were also conducted on normalised and non-normalised data-set. Random Forest was also tested against the ANN for the classification accuracy.

5 Result Analysis and Conclusion

Among all the experiments conducted, the best results were obtained from the ANN mentioned in Table 1. The ANN with single level of hidden layer was found to be performing similarly as p-BiLSTM. Table 2

Table 2: Result Comparison

Accuracy	Using ANN		P-BiLSTM	
	Six-way	Binary	Six-way	Binary
Train	38.46%	70.15%	-	-
Validation	41.82%	67.60%	37.00%	70.00%
Test	40.84%	69.91%	36.00%	70.00%

References

[1] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[3] 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*, vol. 12, pp. 2825–2830, 2011.

[4] T. Oliphant, "NumPy: A guide to NumPy," USA: Trelgol Publishing, 2006–, [Online; accessed <today>]. [Online]. Available: http://www.numpy.org/

[5] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 51 – 56.

[6] E. Loper and S. Bird, "Nltk: the natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[7] T. Alhindi, S. Petridis, and S. Muresan, "Where is your evidence: Improving fact-checking by justification modeling," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pp. 85–90.