Fake News Detection Through Multi-Perspective Speaker Profiles

Yunfei Long[1], Qin Lu[1], Rong Xiang[2], Minglei Li[1] and **Chu-Ren Huang[3]**
[1]Department of Computing, The Hong Kong Polytechnic University
csylong,csluqin,csmli@comp.polyu.edu.hk
[2]Advanced Micro Devices, Inc.(Shanghai)
Rong.Xiang@amd.com
[3]Department of Chinese and Bilingual Studies, The Hong Kong Polytechnic University
churen.huang@polyu.edu.hk

Abstract

Automatic fake news detection is an important, yet very challenging topic. Traditional methods using lexical features have only very limited success. This paper proposes a novel method to incorporate speaker profiles into an attention based LSTM model for fake news detection. Speaker profiles contribute to the model in two ways. One is to include them in the attention model. The other includes them as additional input data. By adding speaker profiles such as party affiliation, speaker title, location and credit history, our model outperforms the state-of-the-art method by 14.5% in accuracy using a benchmark fake news detection dataset. This proves that speaker profiles provide valuable information to validate the credibility of news articles.

1 Introduction

Fake news is written and published with the intent to mislead its readers in order to gain financially or politically, often with sensationalist, exaggerated, or patently false headlines attract readers' attention [1]. Fake news is more dangerous than newspaper gaffes, especially in social media. One of the worst effect of fake news in China is an incidence after the Fukushima Daiichi nuclear disaster in 2011. A fake news in microbiology claims that salt can prevent radiation, but the nuclear disaster makes salt contaminated. This fake news triggered people stockpiling table salt in China, resulted in a huge market disorder. [2]

Fake news detection is defined as the task of categorizing news along a continuum of veracity with an associated measure of certainty (Conroy et al., 2015). Detecting fake news in social media has been an extremely important, yet technically very challenging problem. The difficulty comes partly from the fact that even human beings may have difficulty identifying between real news and fake news. In one study, human judges, by a rough measure of comparison, achieved only 50-63 % success rates in identifying fake news (Rubin, 2017).

The most of fake news detection algorithms try to linguistic cues (Feng and Hirst, 2013; Markowitz and Hancock, 2014; Ruchansky et al., 2017). Several successful studies on fake news detection have demonstrated the effectiveness of linguistic cue identification, as the language of truth news is known to differ from that of fake news (Bachenko et al., 2008; Larcker and Zakolyukina, 2012). For example, deceivers are likely to use more sentiment words, more sense-based words (e.g., seeing, touching), and other-oriented pronouns, but less self-oriented pronouns. Compared to real news, fake news shows lower cognitive complexity and uses more negative emotion words. However, the linguistic indicators of fake news across different topics and media platforms are not well understood. Rubin (2015) points out that there are many types of fake news, each with different potential textual indicators. This indicates that using linguistic features is not only laborious but also topic/media dependent domain knowledge, thus limiting the scalability of these solutions.

In addition to lexical features, speaker profile information can be useful (Gottipati et al., 2013; Long et al., 2016). Speaker profiles, including party affiliations, job title of speaker, as well as topical information which can also be used to in-

[1] $https://en.wikipedia.org/wiki/Fake_news$
[2] https://blogs.wsj.com/chinarealtime/2011/03/17/fearing-radiation-chinese-rush-to-buy-table-salt/

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 252–256,
Taipei, Taiwan, November 27 – December 1, 2017

dicate the credibility of a piece of news. For example, a conservative might neglect the impact of climate change, while a progressive might exaggerate inequality. On some occasions, it is hard to make fake claims like congressional inquiry. But in other cases, the speaker tend to exaggerate facts like the campaign rally. For the study use profile information, Wang (2017) proposes a hybrid CNN model to detect fake news, which uses speaker profiles as a part of the input data. Wang (2017) also made the first large scale fake news detection benchmark dataset with speaker information such as party affiliation, location of speech, job title, credit history as well as topic.

The Long-Short memory network (LSTM), as a neural network model, is proven to work better for long sentences (Tang et al., 2015). Attention models are also proposed to weigh the importance of different words in their context. Current attention models are either based on local semantic attentions (Yang et al., 2016) or user attentions(Chen et al., 2016).

In this paper, we propose a model to incorporate speaker profile into fake news detection. Profile information is used in two ways. The first way includes profiles as an attention factor while the second one includes profiles as additional input data. Evaluations are conducted using the dataset provided by Wang (2017). Experimental results show that incorporating speaker profile can improve performance dramatically. Our accuracy can reach 0.415 in the benchmark dataset, about 14.5% higher than state-of-the-art hybrid CNN model.

2 Proposed Model

Our proposed model uses LSTM model (Gers, 2001) as the basic classifier. The attention models and speaker profile information are added into LSTM to form a hybrid model.

Figure 1 shows the hybrid LSTM. Note that the first LSTM is for obtaining the representation of news articles. The speaker profile is used to construct two attention factors. One uses only the speaker profile and the other uses topic information of the news articles. The second LSTM simply uses speaker profiles to obtain the vector presentations of speakers. The two representations are then concatenated in the soft-max function for classification.

Let D be a collection of news and P be a collection of profiles. A piece of news, d_k, $d_k \in D$, includes both its text as a sentence, denoted by s_k, and a speaker profile , denoted by p_k, $p_k \in P$. A sentence s_k is formed by a sequence of words $s_k = w_{k1}w_{k2}...w_{kl_k}$, where l_k is the length of s_k. The features of a word $w_i \in S_k$ form a word vector $\vec{v}^{w_i}$ with length N, $\vec{v}^{w_i} = [{F_1}^{w_i}, {F_2}^{w_i}....{F_N}^{w_i}]$. Every $w_k \in s_k$ and $p_k \in P$ are fed into the first LSTM. The speaker based profile vector $\vec{s}$ and the topic based profile vector $\vec{t}$ serve as the two attention factors for s_k. The output of the two LSTM models are $\vec{s_k}$ and $\vec{p_k}$. Finally, $\vec{s_k}$ and $\vec{p_k}$ are concatenated to form the representation $\vec{d_k}$. The final layer projects $\vec{d_k}$ onto the target space of L class labels though a soft-max layer.

A LSTM model has five types of gates in each node i represented by five vectors including an input gate $\vec{i_i}$, a forget gate $\vec{f_i}$, an output gate $\vec{o_i}$, a candidate memory cell gate $\vec{c'_i}$, and a memory cell gate $\vec{c_i}$. $\vec{f_i}$ and $\vec{o_i}$ are used to indicate which values will be updated, either to forget or to keep. $\vec{c'_i}$ and $\vec{c_i}$ are used to keep the candidate features and the actual accepted features, respectively.

Each node i corresponds to each word w_i in a given sentence S_k, represented by its word embedding $\vec{w_i}$. The LSTM cell state $\vec{c_i}$ and the hidden state $\vec{h}_{S_k:w_i}$ can be updated in two steps. In the first step, the previous hidden state $\vec{h}_{S_k:w_{i-1}}$ uses a hyperbolic function to form $\vec{c'_i}$ as defined below.

$$\vec{c'_i} = tanh(\hat{W}_c * [\vec{h}_{S_k:w_{i-1}} * \vec{w_i}] + \hat{b}), \quad (1)$$

where $\hat{W}_c$ is a parameter matrix, $\vec{h}_{S_k:w_{i-1}}$ is the previous hidden state and $\vec{w_i}$ is the word vector. $\hat{b}$ is the regularization parameter matrix. In the second step, $\vec{c_i}$ is updated by $\vec{c'_i}$ and its previous state $\vec{c}_{i-1}$ according to the below formula:

$$\vec{c_i} = \vec{f_i} \odot \vec{c}_{i-1} + \vec{i_i} \odot \vec{c'_i}. \quad (2)$$

The hidden state of w_i can be obtained by

$$\vec{h}_{S:w_i} = \vec{o_i} tanh(\vec{f_i} \odot \vec{c_i}). \quad (3)$$

The forget gate $\vec{f_i}$ is for keeping the long term memory. A series of hidden states $\vec{h_1}\vec{h_2}...\vec{h_i}$ can serve as input to the average pooling layer to obtain the representation $\vec{s_k}$. $\vec{p_k}$, the representation of p_k, can be obtained similarly through the same LSTM model. Details will not be repeated here.

Similar to other attention models, speaker profile based attention factors are included in the

first LSTM for each text, s_k. Rather than feeding speaker profiles as hidden states to an average pooling layer, we use attention mechanism which uses a weighting scheme to indicate importance of different words to the meaning of the news.

We use speaker profile and topic information for weight training, which are represented by continuous and real-valued vectors $\vec{s}$ and $\vec{t}$, respectively. Let α_{w_i} denote the weight to measure the importance of w_i in s_k. The attention weight α_{w_i} for each hidden state can be defined as:

$$\alpha_{w_i} = \frac{exp(e(\vec{h}_{S_k:w_i}, \vec{s}, \vec{t}))}{\sum_{k=1}^{l_i} exp(e(\vec{h}_{S_k:w_i}, \vec{s}, \vec{t}))} \quad (4)$$

where e is score function to indicate the importance of words defined by:

$$exp(e(\vec{h}_{S_k:w_i}, \vec{s}, \vec{t}) = v^T tanh(\hat{W}_H \vec{h}_{S_k:w_i} + \hat{W}_s \vec{s} + \hat{W}_t \vec{t} + \vec{b}) \quad (5)$$

where W_H, W_S, W_T are weight matrices, v is weight vector and v^T denotes its transpose. This model can train speaker vector $\vec{s}$ and topic vector $\vec{t}$ at the same time.

Formally, the enhanced sentence representation $\vec{s_k}$ is a weighted sum of hidden states as Formula 6

$$\vec{s_k} = \sum_t \alpha_{w_i} \vec{h}_{S_k:w_i}. \quad (6)$$

Similarly, we use the second LSTM model to get the representation of profile $\vec{p_k}$. The final news representation $\vec{d_k}$ is computed by concatenating $\vec{p_k}$ and $\vec{s_k}$, represented in Formula 7:

$$\vec{d_k} = \vec{s_k} \oplus \vec{p_k}. \quad (7)$$

In LSTM model, we use a hidden layer to project the final news vector $\vec{d_k^f}$ through a hyperbolic function.

$$\vec{d_k^f} = tanh(\hat{W}_h \vec{d_k} + \hat{b}_h), \quad (8)$$

where $\hat{W}_h$ is the weight matrix of the hidden layer and $\hat{b}_h$ is the regularization matrix.

Finally, prediction for any label $l \epsilon L$ obtained by the soft-max function is defined as:

$$P(y = l|\vec{d_k^f}) = \frac{e^{\vec{d_k^f}^T \vec{W}_l}}{\sum_{l=1}^{L} e^{\vec{d_k^f}^T \vec{W}_l}} \quad (9)$$

where $\vec{W}_l$ is the soft-max weight for each label.

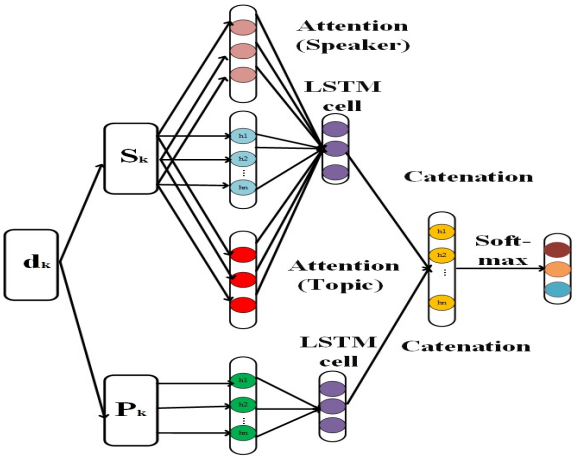

Figure 1: Hybrid fake news detection model with attention mechanism

Statistics	Num	Labels	Num
Training Set	10,269	Pants on fire	1,050
Develop Set	1,284	False	2,511
Testing Size	1,283	Barely-true	2,108
Democrats	4,150	Half-true	2,638
Republicans	5,687	Mostly-true	2,466
Others	2,185	True	2,063

Table 1: The statistics of the LIAR data set

3 Performance Evaluation

Evaluations are performed using the LIAR dataset by Wang (2017). The dataset contains 12,836 short statements from 3,341 speakers covering 141 topics in POLITIFACT.COM[3]. Each news includes text content, topic, and speaker profile. Speaker profiles include speaker name, title, party affiliation, current job, location of speech, and credit history. The credit history includes the historical records of inaccurate statements for each speaker. Annotation is based on evaluations by professional editors. The labels take discrete values from 1 to 6 corresponding to *pants-fire, false, barely-true, half-true, mostly-true*, and *true*. The statistics are listed in the Table 1.

Models	Dev.	Test
Majority	0.204	0.208
LSTM	0.241	0.245
CNN-Wang	0.247	0.247
CNN-WangP	**0.247**	**0.270**

Table 2: The results for baseline models

[3]http://www.politifact.com/truth-o-meter/

The performance of four baseline models are shown in Table 2 including a simple majority model, the LSTM model without using profile information, the hybrid CNN model proposed by Wang (2017) without profile information(CNN-Wang), and the hybrid CNN model by Wang with profile information(CNN-WangP).

Note that firstly, LSTM without profile does not perform better than CNN-Wang. However, other studies show that when attention model is incorporated, LSTM generally outperforms that of CNN model (Chen et al., 2016; Yang et al., 2016) which will be shown later. Secondly, CNN-WangP, which uses speaker profiles has the best performance. For word representation, we train the skip-gram word embedding (Mikolov et al., 2013) on each dataset separately to initialize the word vectors. All embedding sizes on the model are set to $N = 200$, a commonly used size.

In speaker profiles, there are four basic attributes: *party affiliation*(*Pa*), *location of speech*(*La*), *job title of speaker*(*Ti*), and *credit history*(*Ch*) which counts the inaccurate statements for speaker in past speeches. Note that *credit history* is not a commonly available data. It is included here for comparison to CNN-Wang(P). We conduct experiment on using these attributes individually and in combinations.

	Without Att		With Att	
Profile	Dev	Test	Dev	Test
CNN-Wang	0.247	0.247	N/A	N/A
CNN-WangP	0.247	0.274	N/A	N/A
Base LSTM	0.241	0.245	0.250	0.255
Pa	0.247	0.246	0.253	0.257
La	0.243	0.245	0.266	0.268
Ti	0.241	0.247	0.258	0.257
Ch	0.363	0.368	**0.378**	**0.385**
Pa+La	0.250	0.252	0.255	0.261
Pa+Ti	0.261	0.264	0.267	0.265
La+Ti	0.267	0.264	0.268	0.271
Ti+Ch	0.378	0.380	**0.381**	**0.387**
Pa+Ti+Ch	0.385	0.387	0.397	0.395
Pa+La+Ti	0.270	0.275	0.274	0.279
La+Ti+Ch	0.388	0.401	**0.398**	**0.405**
Pa+La+Ti+Ch	0.392	0.399	**0.407**	**0.415**

Table 3: Evaluation on accuracy using different combinations of profile attributes

Table 3 shows the performance of our proposed model with the top performers of the baseline systems put in the first two lines. The basic LSTM model shown as Base-LSTM in Table 3 performs less than CNN-WangP and similar to CNN-Wang without profile information. In other words, LSTM has no obvious model advantage in this set of training data. We may also infer that the lexicon and style differences between fake news and true news are not large enough for detection. And, the difference in the choice of deep neural network models are also not significant if profile information is not supplied.

Table 3 also shows that speaker profile information can improve fake news detection significantly. Besides *credit history*, which gives the largest improvement of 3%, *location of speaker* gives more improvements than *part affiliation* and *job title* with improvement of 2.3%. When all attributes are included in detection, the performance surge to over 40% in accuracy. Obviously, if *credit history* of a speaker is available, it is not hard to see how useful it is for fake new detection. In practice, however, we cannot expect the credit history information to be available all the time for fake news detection. Therefore, it is more important to observe those combinations without *Ch* for *credit history*. The best performers without *Ch* are marked with underlines. The combination of using all three attributes still outperforms CNN-WangP by 16.7% even though CNN-WangP has *credit history* included. This further proves the effectiveness of our proposed method.

4 Conclusion

This paper proposes a hybrid attention-based LSTM model for fake news detection. In our model, speaker profiles can contribute to fake news detection in two ways: One is to include them as attention factors for the learning of news text; and the other is to use them as additional inputs to provide more information. Experimental results show that both methods of using speaker profiles can contribute to the improvement of fake news detection. This can be interpreted as speaker's intention to speak the truth or fake it largely depends on his/her, profiles, especially his/hers credit history. Adopting speaker profiles into an attention based LSTM detection model can reach over 41.5% in accuracy with net increase of 14.5% in accuracy compared to the state-of-the-art model. Even without the use of credit history, the performance net increase is still by 0.5%.

Acknowledgments

The work is partially supported by the research grants from Hong Kong Polytechnic University (PolyU RTVU) and GRF grant(CERG PolyU 15211/14E).

References

Joan Bachenko, Eileen Fitzpatrick, and Michael Schonwetter. 2008. Verification and implementation of language-based deception indicators in civil and criminal narratives. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 41–48. Association for Computational Linguistics.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. EMNLP.

Niall J Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Vanessa Wei Feng and Graeme Hirst. 2013. Detecting deceptive opinions with profile compatibility. In *IJCNLP*, pages 338–346.

Felix Gers. 2001. *Long short-term memory in recurrent neural networks*. Ph.D. thesis, Universität Hannover.

Swapna Gottipati, Minghui Qiu, Liu Yang, Feida Zhu, and Jing Jiang. 2013. Predicting users political party using ideological stances. In *International Conference on Social Informatics*, pages 177–191. Springer.

David F Larcker and Anastasia A Zakolyukina. 2012. Detecting deceptive discussions in conference calls. *Journal of Accounting Research*, 50(2):495–540.

Yunfei Long, Qin Lu, Yue Xiao, MingLei Li, and Chu-Ren Huang. 2016. Domain-specific user preference prediction based on multiple user activities. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 3913–3921. IEEE.

David M Markowitz and Jeffrey T Hancock. 2014. Linguistic traces of a scientific fraud: The case of diederik stapel. *PloS one*, 9(8):e105937.

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

Victoria L Rubin. 2017. Deception detection and rumor debunking for social media.

Victoria L Rubin, Yimin Chen, and Niall J Conroy. 2015. Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news. *arXiv preprint arXiv:1703.06959*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

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*.