DUAL: A Deep Unified Attention Model with Latent Relation Representations for Fake News Detection: 19th International Conference, Dubai, United Arab Emirates, November 12–15, 2018,...

Chapter · November 2018

DOI: 10.1007/978-3-030-02922-7_14

CITATIONS
15

READS
672

6 authors, including:

Manqing Dong
eBay Inc.
33 PUBLICATIONS **471** CITATIONS
SEE PROFILE

Lina Yao
CSIRO Data61 and UNSW
431 PUBLICATIONS **9,379** CITATIONS
SEE PROFILE

Quan Z. Sheng
Macquarie University
630 PUBLICATIONS **14,257** CITATIONS
SEE PROFILE

DUAL: A Deep Unified Attention Model with Latent Relation Representations for Fake News Detection

Manqing Dong[1], Lina Yao[1], Xianzhi Wang[1], Boualem Benatallah[1], Quan Z. Sheng[2], and Hao Huang[1]

[1] University of New South Wales, Sydney, NSW, 2052, Australia
manqing.dong@student.unsw.edu.au, lina.yao@unsw.edu.au
[2] Macquarie University, Sydney, NSW, 2109, Australia

Abstract. The prevalence of online social media has enabled news to spread wider and faster than traditional publication channels. The easiness of creating and spreading the news, however, has also facilitated the massive generation and dissemination of fake news. It, therefore, becomes especially important to detect fake news so as to minimize its adverse impact such as misleading people. Despite active efforts to address this issue, most existing works focus on mining news' content or context information from individuals but neglect the use of clues from multiple resources. In this paper, we consider clues from both news' content and side information and propose a hybrid attention model to leverage these clues. In particular, we use an attention-based bi-directional Gated Recurrent Units (GRU) to extract features from news content and a deep model to extract hidden representations of the side information. We combine the two hidden vectors resulted from the above extractions into an attention matrix and learn an attention distribution over the vectors. Finally, the distribution is used to facilitate better fake news detection. Our experimental results on two real-world benchmark datasets show our approach outperforms multiple baselines in the accuracy of detecting fake news.

1 Introduction

People are increasingly referring to social media as an alternative to traditional news channels for news seeking. Due to the easiness of spreading online, large volumes of fake news are generated on a daily basis for various purposes such as rumor spreading, new product promotion, or misleading people to believe some ideas [13]. The spread of fake news can adversely affect our lives in profound ways, making it one of the most significant issues in both the social and trust computing domains. First, fake news can be easily manipulated by advertisers with bad intentions to induce people to make unwise decisions. For example, some advertising news intentionally persuades consumers to accept biased or false beliefs [13]; many people tend to follow their like-minded peers and receiving news that promotes their favored existing narratives, resulting in an echo chamber effect. Second, online news, including fake news, could spread fast and quick to make an impact. Third, fake news affects the way people interpret and respond to the real news in the long term and undermines the foundation of an

honest society. A fake news not only makes it harder for people to distinguish the real news[3] but also causes confusion and triggers people's distrust in one another.

Considering the negative impact of fake news, many efforts have been contributed to the detection of fake news. The existing approaches generally belong to either news content-based or social context-based models. The former focuses on mining the lexical features, syntactic features of news' headlines, body text, and images, writing styles of news, or more recently, deep syntax and rhetorical structure [10]. The latter, in contrast, focuses on the side information of news, such as user-based (e.g., user's age, number of followers) [1], post-based (e.g., post time)[9], and relationship-based (e.g., user-posts network, friendship network) characteristics. Besides a single type of feature, some work leverages multiple aspects of features to detect fake news [11, 8, 4, 2]. A limitation of such work is that most of them simply treat different types of features equally or combine them linearly before feeding them into classification models. This way, they neglect important clues such as the cross information between heterogeneous features and their structured associations.

We propose a unified framework that comprehensively incorporates the two types of features and their cross information for fake news detection. In a nutshell, we make the following contributions:

- We propose a hierarchical attention-based feature extraction scheme to distill the salient information and relationships from news content and social context, followed by a binary classification method that combines those features via an attention matrix for fake news detection.
- We develop a mechanism for preprocessing the two scopes of features which has suitability for diverse fake news datasets: an attention-based bidirectional Gated Recurrent Units (GRU) model to obtain robust representations of news content and a deep neural network to nonlinearly transform the related social context information into discriminating feature representations, as well as the corresponding learning methods.
- We evaluate our framework on two real-world benchmark datasets. The experimental results demonstrate its effectiveness in detecting fake news and its superior performance to the baselines and other competitive approaches. We have made the related source code[4] publicly available for the reproduction of our model by other researchers.

2 Related Work

2.1 Non-hybrid models for fake news detection

For detecting fake news, we need to consider various domains of information, such as text, relationship, speaker/writer files. There have been plenty of work

[3] https://nytimes.com/2016/11/28/opinion/fake-news-and-the-internet-shell-game.html?_r=0

[4] https://github.com/dongmanqing/A-Deep-Unified-Attention-Model-with-Latent-Relation-Representations

researching with single domain information. Generally, the works could be divided into two aspects: *news content based models* and *social context based models*. For news content based models, some researchers focus on knowledge-based techniques to detect the truthfulness of news. For example, in the work by Shi et al. [12], the authors used a knowledge graph to check whether the claims in news content can be inferred from existing facts in the knowledge graph. Some focused on mining the writing style of the news[10]. For social context based models, the side information is more concerned. For example, Ma et al. [9] use Recurrent Neural Network (RNN) to capture the changes in posts over time and to detect rumors.

2.2 Hybrid models for fake news detection

To comprehensively learn from a dataset, it is better to combine the different type of features. A related problem is how to effectively combine those features. One simple way is extracting different kinds of features and feeds them into traditional classification models. For example, Janze et al. [4] extract different kinds of features for detecting fake Facebook news posted during the U.S. presidential election of 2016[5]. They consider cognitive cues (e.g., word counts, the polarity of opinion), visual cues (e.g., the brightness of the posted picture), affective cues (e.g., the number of votes for the post), and behavior cues (e.g., the number of comments). They combine those features as feature vectors and input them into models like Support Vector Machine (SVM) and Random Forest. These methods neglect the relationship between the features and normally one model could not well grasp all different kinds of features at once. The other way is finding the relationship between the features and dealing with them in a hierarchical way. In Ruchansky's work [11], they considered three aspects of features: the text of an article, the user response it receives, and the source users promoting it. A model called CSI (capture, score and integrate) is proposed in [6], where the 'capture' component uses Recurrent Neural Network (RNN) capture the temporal pattern of user activity on a given article, 'score' learns the source characteristic based on the behavior of users, and 'integrate' unifies these two hidden representations to do the classification.

3 The Methodology

Similar to [13], we define fake news as *a news article that is intentionally and verifiably false* and formulate fake news detection as a classification problem. Given a set of news content $X_c = \{x_{c_1}, x_{c_2}, \ldots, x_{c_N}\}$ and their side information (e.g., speaker's or writer's profiles, reviewers feedback), $X_s = \{x_{s_1}, x_{s_2}, \ldots, x_{s_N}\}$, the goal is to predict the labels for news, denoted by $Y = \{y_1, y_2, \ldots, y_N\}$, where $y_i = 1$ if news i is fake news and otherwise truthful news. The structure of our model is shown in figure 1 .

3.1 Input Preprocessing

Side Information Preprocessing We consider two types of the side information: speakers' profiles and reviewers' feedback. Speaker's profiles usually contain

[5] https://github.com/BuzzFeedNews/2016-10-facebook-fact-check

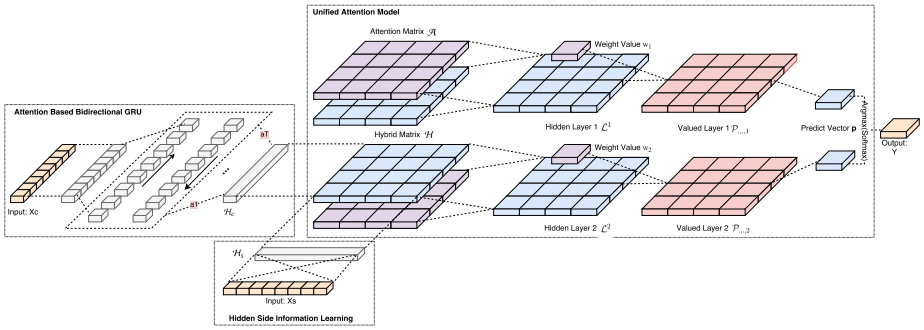


Fig. 1. Example of a binary classification with our model. First we have content information X_c go through with an attention based bidirectional GRU to get hidden representation $\mathcal{H}_c$, and then side information X_s go though with deep neural networks to get hidden representation $\mathcal{H}_s$. Hidden matrix $\mathcal{H}$ is then generated by unifying $\mathcal{H}_c$ and $\mathcal{H}_s$. For binary classification, we calculate two attention matrices $\mathcal{A}$, to get the hidden layer $\mathcal{L}^1$ and $\mathcal{L}^2$. The final prediction $\hat{y}$ is obtained by giving weights $\mathbf{w}$ to the two hidden layers.

categorical features such as speaker's id (or name), speaker's job title, speaker's party, and the news release platform. For categorical variables that with a large number of factors, we use a clustering strategy to obtain the features and consider both counts and the accuracy for each category. For details, we first divide the features into a certain number of categories (here we used 5) according to the counts in training dataset and then set the value of according categories as the history trustworthy values in training dataset. For example, in the training dataset in Table 1, 'Jack' posted 30-39 posts and among those posts, the proportion for truthful news is 0.78. Thus, the posts by 'Jack' is designated the speaker ID feature of $[0, 0, 0, 0.78, 0]$. For every speaker ID that appears only in the test dataset, we set this feature in 0-9 and the credit history as 0.5.

Table 1. Example for Speaker ID preprocessing

Speaker ID	0-9	10-19	20-29	30-39	above 39
Jack	0	0	0	0.78	0

Reviewers' feedback contains features like support/oppose responses or like/-dislike/share toward the articles. By counting the number of features like supports $\#supports$ or oppositions $\#oppositions$, we get numerical variables representing reviewers feedback information for later analysis.

Content Information Preprocessing We use embedding methods for preprocessing the content information [3]. Word embedding is a set of techniques for natural language processing, which represents words or phrases by generating and using some lower dimensional vectors. It has recently gained success in many applications which further boost its application.

3.2 Latent Features Learning

Content Latent Features Learning As mentioned before, the traditional RNN has the ability to better capture contextual information and cannot pay attention to the salient parts of text. Thus, it is inefficient in containing global information [6]. To deal with this inefficiency, researchers have proposed adding attention mechanisms into it, where attention mechanism could give each word/sequence an attention value [15]. Here, we use an attention-based bi-directional GRU model that follows the design of [15] to encode the news content sequences and learn the attention values for words in each sentence. We can get the latent content vector $\mathcal{H}_c$ by summing up the weighted word annotations.

Side Information Latent Features Learning For learning the side information, we mainly used deep neural networks to obtain hidden representations of side information features: $\mathcal{H}_s = \sigma(X_s W_s + b_s)$, where $W_s \in \mathbb{R}^{M_1 \times D_1}$ are weights and $b_s \in \mathbb{R}^{D_1}$ are biases in neural networks. D_1 is the dimension of the latent side information vectors, it is a self defined parameters, and normally we set this dimension based on the dimension of the input vectors. σ is activate function such as sigmoid function [5].

3.3 Unified Attention Model

From above we get hidden representations for side information $\mathcal{H}_s \in \mathbb{R}^{D_1}$ and content information $\mathcal{H}_c \in \mathbb{R}^{D_2}$. To fully obtain the cross-domain information of content features and side information features, we take the cross product of these two hidden representation vectors as a hidden matrix:

$$\mathcal{H} = \mathcal{H}_s \otimes \mathcal{H}_c \tag{1}$$

To match this hybrid hidden representation $\mathcal{H}$ to the final prediction, we construct K attention matrices $\mathcal{A}^k$ to capture the relationship between $\mathcal{H}$ towards each label $k \subseteq K$, separately. Combined with those attention matrices, we obtained the target layer $\mathcal{L}^k$ for label k, by following the next steps:

$$\mathcal{T}^k = \sigma({U^k}^\top \mathcal{H} V^k + B^k) \tag{2}$$

$$\mathcal{A}^k_{ij} = \frac{\exp \mathcal{T}^k_{ij}}{\Sigma_j \Sigma_i \exp \mathcal{T}^k_{ij}} \cdot D_1 D_2 \tag{3}$$

$$\mathcal{L}^k = \mathcal{A}^k \odot \mathcal{H} \tag{4}$$

where we first construct the transform matrix $\mathcal{T}$ for the hidden matrix $\mathcal{H}$ by multiplying two reversible weight matrix U^k and V^k, and then through an activate function. U^k and V^k are weight matrix and $U^k \in \mathbb{R}^{D_1 \times D_1}$, $V^k \in \mathbb{R}^{D_2 \times D_2}$, B is bias matrix and $B \in \mathbb{R}^{D_1 \times D_2}$, and σ is an activation function. The attention matrix $\mathcal{A}$ is calculated by softmax function, which could be regarded as normalized importance weights for each dots in the hidden matrix $\mathcal{H}$. When D_1 and D_2 are big (e.g. when D_1, D_2 equal to 50, then $D_1 \times D_2$ will be 2500), most of $\mathcal{A}_{ij}$ would be a quite small value that near zero. Thus the former softmax value then multiplies the dimension D_1 and D_2 as shown in equation 11. After all, we

get a weighted version of a hidden matrix $\mathcal{H}$, which is layer $\mathcal{L}^k$, towards label k by taking the dot product of $\mathcal{A}^k$ and $\mathcal{H}$.

We also give each layer a weight towards the final prediction, where for a layer $\|$ be predicted to label k, we want to give more weight for this layer. We defined the weights as $\mathbf{w} = \{w_1, \ldots, w_K\}$, and they are calculate by the following,

$$\mathbf{h} = \sigma(\mathbf{u}^\top \mathcal{H} \mathbf{v}) \tag{5}$$

$$w_k = \frac{\exp h_k}{\Sigma_k \exp h_k} \tag{6}$$

where $\mathbf{u} \in \mathbb{R}^{D_1 \times K}$, $\mathbf{v} \in \mathbb{R}^{D_2 \times K}$ are transform vectors, σ is an activation function. Thus we first learn a hidden weight $\mathbf{h}$ for layer weights $\mathbf{w}$, and then use softmax as a normalization function to get each element in $\mathbf{w}$.

By multiply the weights $\mathbf{w}$ to $\mathcal{L}^k$ we got the final prediction cube $\mathcal{P} \in \mathbb{R}^{D_1 \times D_2 \times K}$, where

$$\mathcal{P}_{:,:,k} = w_k \cdot \mathcal{L}^k \tag{7}$$

For the final prediction, we sum up the elements in each layer in $\mathcal{P}$ by the axis i and j, which means we sum up the elements in $\mathcal{P}_{:,:,k}$. By doing so, we get the final predict vector $\mathbf{p}$ which consists of the probability that a sample to be label k, where is listed below:

$$\mathbf{p} = softmax(\Sigma_i \Sigma_j \mathcal{P}_{i,j,k=1}, \ldots, \Sigma_i \Sigma_j \mathcal{P}_{i,j,k=K}) \tag{8}$$

Then the final prediction is

$$\hat{y} = \text{argmax}(\mathbf{p}) \tag{9}$$

We take cross entropy loss as the loss function for our prediction, where

$$L_p(X, Y, \Theta) = -\Sigma_k y \log \hat{y} \tag{10}$$

In the training process, we replace $\hat{y}$ to $\mathbf{p}$ for back-propagation. Also, we consider regularizing the parameters to prevent over-fitting. Here, we choose ℓ_2 norm for each parameter. Thus, the total loss is summing up the prediction loss and the regularization function. The optimization goal is to minimize the total loss:

$$\underset{\Theta}{\text{argmin}}\, L(X, Y, \Theta) = L_p(X, Y, \Theta) + \lambda \parallel \Theta \parallel_2, \tag{11}$$

here, we use Θ represents all the parameters used in our algorithms, and λ is a hyper parameter. As for learning the parameters, they are mainly updated by back propagation via Adam optimization methods [5].

4 Experiments

4.1 Dataset Description

Evaluations are performed using two real-world manual labeled benchmark datasets for fake news detection task: LIAR dataset [14] and Buzzfeed News[6]. For each

[6] https://github.com/BuzzFeedNews/2016-10-facebook-fact-check

dataset, we consider a binary classification problem to distinguish truthful news from fake news. The LIAR Dataset contains 12,836 short statements from 3,341 speakers covering 141 topics in POLITIFACT [7]. Each news item includes text content, topic, and speaker profile (speaker name, title, party affiliation, current job, the location of the speech, and credit history). The labels take discrete values from 1 to 6 corresponding to pants-fire, false, mostly-false, half-true, mostly-true, and true. Data with the first three labels are labeled as fake news and the others are labeled as truthful news. We further deduct parts of the dataset for balancing the dataset, and we get around 4,000 truthful statements and 4,000 fake news. And Buzzfeed News Dataset contains 2,282 posts published on Facebook from 9 news agencies over a week close to the 2016 U.S. election. Every post and linked article were fact-checked claim-by-claim by 5 BuzzFeed journalists. Each article includes category (e.g., mainstream), public account, post type, spreading counts (number of shares, likes, and comments). We regard data with the label 'mostly true' as truthful news and others ('no factual content' and 'mixture of true and false') as fake news. Among the 2,282 posts, almost 75% of them are truthful news. We balanced the number of truthful and fake news with a 50-50 proportion and finally got 800 items for final prediction.

4.2 Comparison of the results

Our default parameter settings are 2 fully connected layers for extracting side information, word embedding with dimension 200, and the attention matrix with 20×20. And we compare with a set of state-of-the-art methods and baselines. 1. Rubin et al. [10]: Rubin extracted news style-based features by combines the vector space model and rhetorical structure theory, and they used Support Vector Machine for the classification. 2. Castillo et al. [1]: This method predicts news veracity using social engagements. The features are extracted from user profiles and friendship network. 3. Janze et al. [4]: as mentioned in related work, Janze et al. considered various type of features for detecting the fake news and used SVM as the classifier. Since there are no visual cues in LIAR dataset, we didn't extract visual features from Buzzfeed News as well. 4. Yunfei et al. [8] propose to utilized a similar way as ours for predicting fake news. They used attention based LSTM model to learn context hidden representation and LSTM for learning the speaker's file. The difference is they concatenate the hidden representations and feed them into a softmax layer for prediction. 5. X_c only: where only content features are concerned and is processed with attention based bidirectional GRU for prediction. 6. X_s only: where only side information is concerned and is processed with deep neural networks. 7. $X_s + X_c + NN$: the side information and content features are simply concact and feed into deep neural networks. 8. $X_s + X_c + biNN$: first learn the hidden representations for both side information and content with the same methods as our model's, and then processed them with bilinear neural networks [7].

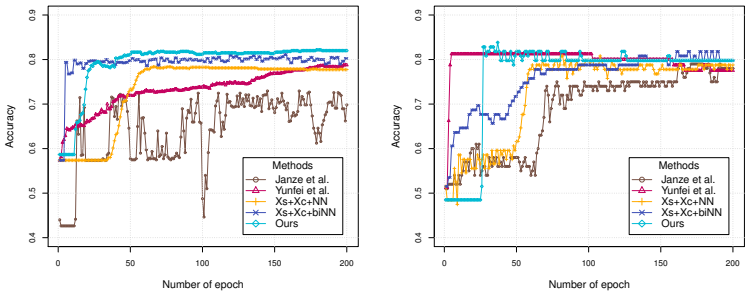
The results are listed in Table 2. Generally, model concerned with side information performs better than model concerned with only content features,

[7] http://www.politifact.com/

Table 2. Prediction comparison in two datasets

	LIAR Dataset				Buzzfeed News			
Methods	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Rubin et al.	0.5719	0.5953	0.5336	0.5445	0.6108	0.6022	0.5612	0.5551
Castillo et al.	0.7798	**0.7772**	0.7914	0.7837	0.7475	0.7357	0.7833	0.7562
Janze et al.	0.7319	0.6433	0.7029	0.6718	0.7900	0.8163	0.7692	0.7921
Yunfei et al.	0.8063	0.7361	0.7943	0.7641	0.8081	0.8125	0.7959	0.8041
X_c only	0.6225	0.5619	0.5191	0.5396	0.6531	0.5417	0.6842	0.6047
X_s only	0.7300	0.6686	0.6786	0.6722	0.7959	0.7500	**0.8182**	0.7826
$X_s + X_c + NN$	0.7838	0.7625	0.7386	0.7504	0.8081	0.8542	0.7736	0.8119
$X_s + X_c + biNN$	0.8113	0.7361	0.8045	0.7688	0.8182	0.8750	0.7778	0.8235
Ours	**0.8283**	0.7621	**0.8112**	**0.7859**	**0.8384**	**0.9167**	0.7857	**0.8462**

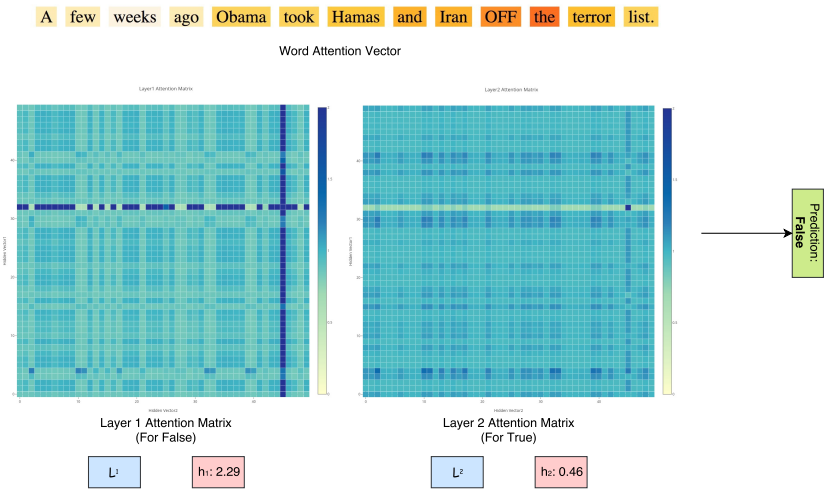
and hybrid model performs better than singleton aspect model. Compared with several models, we draw the following conclusions (i) our proposed model outperforms all the comparison methods, which indicate the attention matrix does capture more effective relationships between side information and content features; (ii) the proposed structure is fixable in different datasets. Figure 2 gives examples of the comparison between hybrid models (Janze's model, Yunfei's model, $X_s + X_c + NN$, $X_s + X_c + biNN$) and our model on two datasets. We can see that our model converges to an optimal accuracy rapidly and performs better than others on LIAR dataset. As for Buzzfeed News dataset, our model's prediction is stable and produces good prediction, while Yunfei's work performs well in the beginning but becomes over-fitting after several iterations.



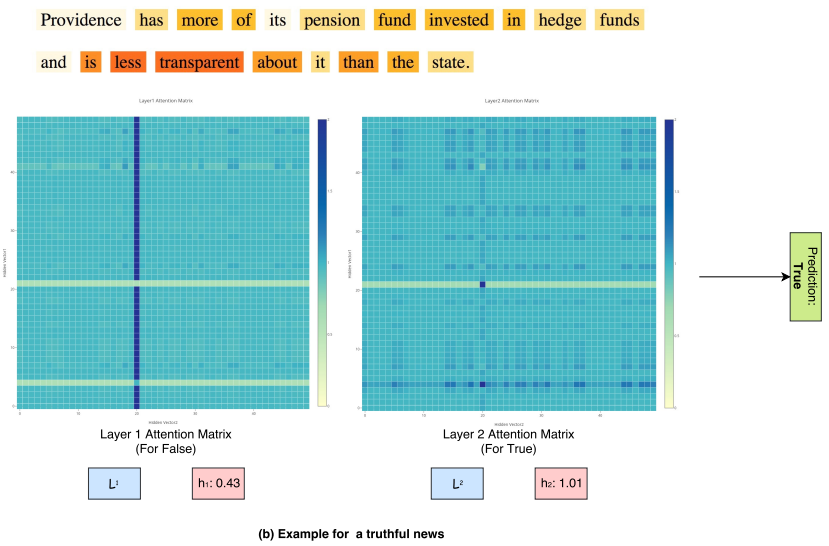
(a) LIAR Dataset (b) Buzzfeed News

Fig. 2. Comparing with hybrid methods

We are giving examples of two news items in LIAR that correctly predicted to be (a) fake or (b) truthful, which shows our model's ability to visualization. The examples are shown in Figure 3. We could see that when determining whether "A few weeks ago Obama took Hamas and Iran off the terror list" is fake or not, we give much attention on "off the terror" (Figure 3 (a)), which is consistent with our life experiences. When we make a decision, description "off" or "on" will give two different concepts, thus it's the key to give the final judgment.



(a) Example for a fake news



(b) Example for a truthful news

Fig. 3. Examples for news of LIAR that are correctly predicted to be (a) fake or (b) truthful. In each sub-figure, the first line shows the attention to the content information. The highlight color is changing from shallow yellow to dark orange, and the darker the highlight, the bigger the attention value. Two matrices show the attention matrices when doing the prediction. And the red rectangles below shows the weights for the attention matrices. Layer $\mathcal{L}^1$ is for fake and layer $\mathcal{L}^2$ is for true. For example, $\mathcal{L}^1$ presents the attention when we search for the cues for false information. And then followed with red rectangles, which show the weights for the attention matrices, to enhance the former judgment.

5 Conclusions

In this paper, we have proposed a novel model DUAL for detecting fake news which combines two scope of features with unified attention for prediction and simultaneously learns the hidden representation of these two features. Specifically, we extract features with adaptive methods, where content features are learned by an attention-based bidirectional GRU and the side information is learned from deep neural networks. Then we unify the hidden representation of these two features as the representation matrix and introduce attention matrix for learning the attention distribution over vectors. We derive the weight vectors for the prediction layer to enhance the performance. The experimental results on two benchmark real-world datasets demonstrate that our method consistently beats a series of state-of-the-art methods and baseline methods.

References

1. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: World wide web. pp. 675–684. ACM (2011)
2. Dong, M., Yao, L., Wang, X., Benatallah, B., Huang, C., Ning, X.: Opinion fraud detection via neural autoencoder decision forest. arXiv:1805.03379 (2018)
3. Huang, C., Yao, L., Wang, X., Benatallah, B., Sheng, Q.Z.: Expert as a service: Software expert recommendation via knowledge domain embeddings in stack overflow. In: ICWS. pp. 317–324. IEEE (2017)
4. Janze, C., Risius, M.: Automatic detection of fake news on social media platforms. Pacific Asia Conference on Information Systems (2017)
5. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: AAAI. vol. 333, pp. 2267–2273 (2015)
7. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: International Conference on Computer Vision. pp. 1449–1457 (2015)
8. Long, Y., Lu, Q., Xiang, R., Li, M., Huang, C.R.: Fake news detection through multi-perspective speaker profiles. In: International Joint Conference on Natural Language Processing. vol. 2, pp. 252–256 (2017)
9. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks. In: IJCAI (2016)
10. Rubin, V.L., Lukoianova, T.: Truth and deception at the rhetorical structure level. Journal of the Association for Information Science and Technology (2015)
11. Ruchansky, N., Seo, S., Liu, Y.: Csi: A hybrid deep model for fake news detection. In: CIKM. pp. 797–806 (2017)
12. Shi, B., Weninger, T.: Fact checking in heterogeneous information networks. In: WWW. pp. 101–102 (2016)
13. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter (2017)
14. Wang, W.Y.: " liar, liar pants on fire": A new benchmark dataset for fake news detection. arXiv preprint arXiv:1705.00648 (2017)
15. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: HLT-NAACL. pp. 1480–1489 (2016)