Call Attention to Rumors: Deep Attention Based Recurrent Neural Networks for Early Rumor Detection

Tong Chen[1(✉)], Xue Li[1], Hongzhi Yin[1], and Jun Zhang[2]

[1] The University of Queensland, Brisbane, Australia
tong.chen@uq.edu.au, xueli@itee.uq.edu.au, db.hongzhi@gmail.com
[2] Swinburne University of Technology, Melbourne, Australia
junzhang@swin.edu.au

Abstract. The proliferation of social media in communication and information dissemination has made it an ideal platform for spreading rumors. Automatically debunking rumors at their stage of diffusion is known as *early rumor detection*, which refers to dealing with sequential posts regarding disputed factual claims with certain variations and highly textual duplication over time. Thus, identifying trending rumors demands an efficient yet flexible model that is able to capture long-range dependencies among postings and produce distinct representations for the accurate early detection. However, it is a challenging task to apply conventional classification algorithms to rumor detection in earliness since they rely on hand-crafted features which require intensive manual efforts in the case of large amount of posts. This paper presents a deep attention model based on recurrent neural networks (RNNs) to *selectively* learn temporal representations of sequential posts for rumor identification. The proposed model delves soft-attention into the recurrence to simultaneously pool out distinct features with particular focus and produce hidden representations that capture contextual variations of relevant posts over time. Extensive experiments on real datasets collected from social media websites demonstrate that the deep attention based RNN model outperforms state-of-the-art baselines by detecting rumors more quickly and accurately than competitors.

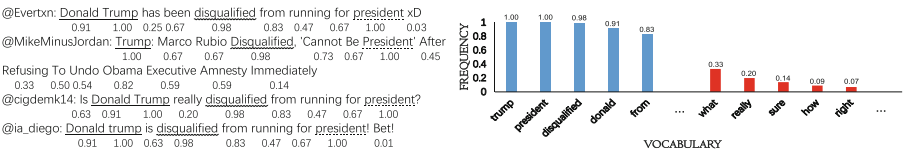
Keywords: Early rumor detection · Recurrent neural networks
Deep attention models

1 Introduction

The explosive use of contemporary social media in communication has witnessed the widespread of rumors which can pose a threat to the cyber security and social stability. For instance, on April 23rd 2013, a fake news claiming two explosions happened in the White House and Barack Obama got injured was posted by a hacked Twitter account named Associated Press. Although the White House and

M. Ganji et al. (Eds.): PAKDD 2018, LNAI 11154, pp. 40–52, 2018.
https://doi.org/10.1007/978-3-030-04503-6_4

Associated Press assured the public minutes later the report was not true, the fast diffusion to millions of users had caused severe social panic, resulting in a loss of $136.5 billion in the stock market[1]. This incident of a false rumor showcases the vulnerability of social media on rumors, and highlights the practical value of automatically predicting the veracity of information.



(a) Streaming posts in regards to an event (b) Statistics on textual phrases

Fig. 1. For social media posts regarding a specific event, *i.e.*,"Trump being disqualified from U.S. election", tokens like "Donald Trump", "Obama" and "disqualified" appear extremely frequently in disputed postings.

Debunking rumors at their formative stage is particularly crucial to minimizing their catastrophic effects. Most existing rumor detection models employ learning algorithms that incorporate a wide variety of features and formulate rumor detection into a binary classification task. They commonly craft features manually from the content, sentiment [1], user profiles [2], and diffusion patterns of the posts [3–5]. Embedding social graphs into a classification model also helps distinguish malicious user comments from normal ones [6,7]. However, feature engineering is extremely time-consuming, biased, and labor-intensive. Moreover, hand-crafted features are data-dependent, making them incapable of resolving contextual variations in different posts.

Recent examinations on rumors reveal that social posts related to an event under discussion are coming in the form of time series wherein users forward or comment on it continuously over time. Meanwhile, as shown in Fig. 1, during the discussion of arbitrary topics, users' posts exhibit high duplication in their textual phrases due to the repeated forwarding, reviews, and/or inquiry behavior [8]. This poses a challenge on efficiently distilling distinct information from duplication and timely capturing textual variations from posts.

The propagation of information on social media has temporal characteristics, whilst most existing rumor detection methodologies ignore such a crucial property or are not able to capture the temporal dimension of data. One exception is [9] where Ma *et al.* uses an RNN to capture the dynamic temporal signals of rumor diffusion and learn textual representations under supervision. However, as the rumor diffusion evolves over time, users tend to comment differently in various stages, such as from expressing surprise to questioning, or from believing to debunking. As a consequence, textual features may change their patterns

[1] http://www.dailymail.co.uk/news/article-2313652/AP-Twitter-hackers-break-news-White-House-explosions-injured-Obama.html.

with time and we need to determine which of them are more important to the detection task. On the other hand, the existence of duplication in textual phrases impedes the efficiency of training a deep network. In this sense, two aspects of temporal long-term characteristic and dynamic duplication should be addressed simultaneously in an early rumor detection model.

1.1 Challenges and Our Approach

In summary, there are three challenges in early rumor detection to be addressed: (1) automatically learning representations for rumors instead of using labor-intensive hand-crafted features; (2) the difficulty of maintaining the long-range dependency among variable-length post series to build their internal representations; (3) the issue of high duplication compounded with varied contextual focus. To combat these challenges, we propose a novel deep attention based recurrent neural network (RNN) for early detection on rumors, namely *CallAtRumors* (**Call At**tention to **Rumors**). The overview of our framework is illustrated in Fig. 2. For one event (i.e., topic) our model converts posts related to one event into feature matrices. Then, the RNN with soft attention mechanism automatically learns latent representations by feed-forwarding each input weighted by attention weights. Finally, an additional hidden layer with *sigmoid* activation function using the learned latent representations to classify whether this event is a rumor or not.

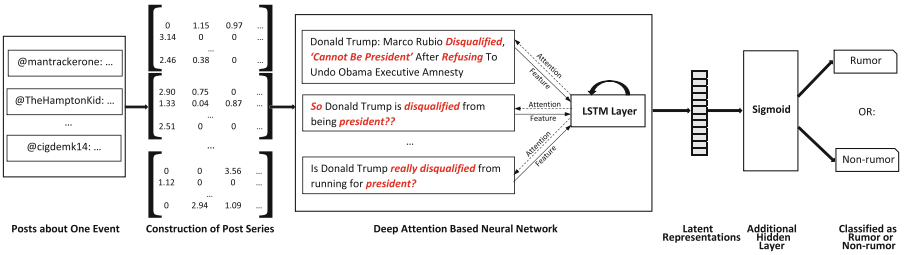


Fig. 2. Schematic overview of our framework.

1.2 Contributions

The main contributions of our work are summarized in three aspects:

- We propose a deep attention neural network that learns to perform rumor detection automatically in earliness. The model is capable of learning continuous hidden representations by capturing long-range dependency an contextual variations of posting series.
- The deterministic soft-attention mechanism is embedded into recurrence to enable distinct feature extraction from high duplication and advanced importance focus that varies over time.

– We quantitatively validate the effectiveness of attention in terms of detection accuracy and earliness by comparing with state-of-the-arts on two real social media datasets: Twitter and Weibo.

2 Related Work

Our work is closely connected with early rumor detection and attention mechanism. We will briefly introduce these two aspects in this section.

2.1 Early Rumor Detection

The problem of rumor detection [10] can be viewed as binary classification tasks. The extraction and selection of discriminative features significantly affects the performance of the classifier. Hu *et al.* first conducted a study to analyze the sentiment differences between spammers and normal users and then presented an optimization formulation that incorporates sentiment information into a novel social spammer detection framework [11]. Also the propagation patterns of rumors were developed by Wu *et al.* through utilizing a message propagation tree where each node represents a text message to classify whether the root of the tree is a rumor or not [3]. In [4], a dynamic time series structure was proposed to capture the temporal features based on the time series context information generated in every rumor's life-cycle. However, these approaches requires daunting manual efforts in feature engineering and they are restricted by the data structure.

Early rumor detection is to detect viral rumors in their formative stages in order to take early action [12]. In [8], some very rare but informative enquiry phrases play an important role in feature engineering when combined with clustering and a classifier on the clusters as they shorten the time for spotting rumors. Manually defined features has shown their importance in the research on real-time rumor debunking by Liu *et al.* [5]. By contrast, Wu *et al.* proposed a sparse learning method to automatically select discriminative features as well as train the classifier for emerging rumors [13]. As those methods neglect the temporal trait of social media data, a time-series based feature structure [4] is introduced to seize context variation over time. Recently, recurrent neural network was first introduced to rumor detection by Ma *et al.* [9], utilizing sequential data to spontaneously capture temporal textual characteristics of rumor diffusion which helps detecting rumor earlier with accuracy. However, without abundant data with differentiable contents in the early stage of a rumor, the performance of these methods drops significantly because they fail to distinguish important patterns.

2.2 Attention Mechanism

As a rising technique in natural language processing problems [14,15] and computer vision tasks [16–18], attention mechanism has shown considerable discriminative power for neural networks. For instance, Bahdanau *et al.* extended the basic encoder-decoder architecture of neural machine translation with attention mechanism to allow the model to automatically search for parts of a source sentence that are relevant to predicting a target word [19], achieving a comparable performance in the English-to-French translation task. Vinyals *et al.* improved the attention model in [19], so their model computed an attention vector reflecting how much attention should be put over the input words and boosted the performance on large scale translation [20]. In addition, Sharma *et al.* applied a location softmax function [21] to the hidden states of the LSTM (Long Short-Term Memory) layer, thus recognizing more valuable elements in sequential inputs for action recognition. In conclusion, motivated by the successful applications of attention mechanism, we find that attention-based techniques can help better detect rumors with regards to both effectiveness and earliness because they are sensitive to distinctive textual features.

3 CallAtRumors: Early Rumor Detection with Deep Attention Based RNN

In this section, we present the details of our framework with deep attention for classifying social textual events into rumors and non-rumors.

3.1 Problem Statement

Individual posts contain very limited content due to their nature of shortness in context. On the other hand, an event is generally associated with a number of posts making similar claims. These related posts can be easily collected to describe an event more faithfully. Hence, we are interested in detecting rumor on an aggregate (event) level instead of identifying each single posts [9], where sequential posts related to the same topics are batched together to constitute an event, and our model determines whether the event is a rumor or not.

Let $\boldsymbol{E} = \{E_i\}$ denote a set of given events, where each event $E_i = \{(p_{i,j}, t_{i,j})\}_{j=1}^{n_i}$ consists of all relevant posts $p_{i,j}$ at time stamp $t_{i,j}$, and the task is to classify each event as a rumor or not.

3.2 Constructing Variable-Length Post Series

Algorithm 1 describes the construction of variable-length post series. To ensure a similar word density for each time step within one event, we group posts into batches according to a fixed post amount N rather than slice the event time span evenly. Specifically, for every event $E_i = \{(p_{i,j}, t_{i,j})\}_{j=1}^{n_i}$, post series are constructed with variable lengths due to different amount of posts relevant to

Input : Event-related posts $E_i = \{(p_{i,j}, t_{i,j})\}_{j=1}^{n_i}$, post amount N, minimum series length Min
Output: Post Series $S_i = \{T_1, ..., T_v\}$

/*Initialization*/;
$v = 1$; $x = 0$; $y = 0$;
while *true* **do**
  if $n_i \geq N \times Min$ **then**
    while $v \leq \lfloor \frac{n_i}{N} \rfloor$ **do**
      $x = N \times (v-1) + 1$;
      $y = N \times v$;
      $T_v \leftarrow (p_{i,x}, ..., p_{i,y})$;
      $v++$;
    end
    $T_v \leftarrow (p_{i,y+1}, ..., p_{i,n_i})$;
  else
    while $v < Min$ **do**
      $x = \lfloor \frac{n_i}{Min} \rfloor \times (v-1) + 1$;
      $y = \lfloor \frac{n_i}{Min} \rfloor \times v$;
      $T_v \leftarrow (p_{i,x}, ..., p_{i,y})$;
      $v++$;
    end
    $T_v \leftarrow (p_{i,y+1}, ..., p_{i,n_i})$;
  end
end
return S_i;

Algorithm 1. Constructing Variable-Length Post Series

different events. We set a minimum series length Min to maintain the sequential property for all events.

To model different words in the post series, we calculate the tf-idf for the most frequent K vocabularies within all posts. Finally, every post is encoded by the corresponding tf-idf vector, and a matrix of $K \times N$ for each time step can be constructed as the input of our model. If there are less than N posts within an interval, we will expand it to the same scale by padding with 0s. Hence, each set of post series consists of at least Min feature matrices with a same size of K (number of vocabularies) $\times$ N (vocabulary feature dimension).

3.3 Long Short-Term Memory (LSTM) with Deterministic Soft Attention Mechanism

To capture the long-distance temporal dependencies among continuous time post series, we employ following Long Short-Term Memory (LSTM) unit which plays an important role in language sequence modelling and time series processing [22–26] to learn high-level discriminative representations for rumors:

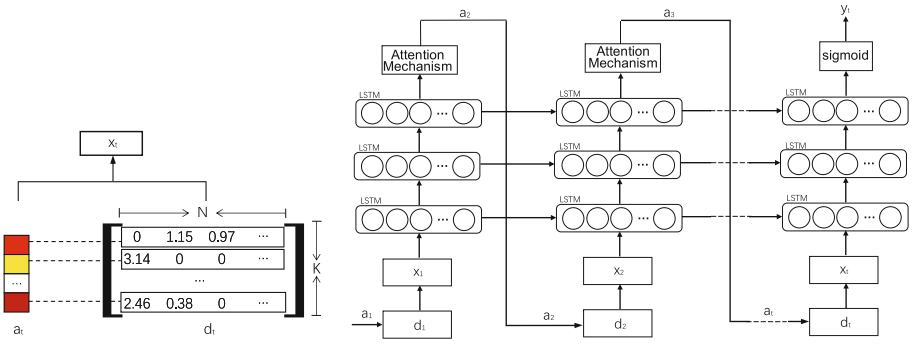
$$\begin{aligned} i_t &= \sigma(U_i h_{t-1} + W_i x_t + V_i c_{t-1} + b_i), \\ f_t &= \sigma(U_f h_{t-1} + W_f x_t + V_f c_{t-1} + b_f), \\ c_t &= f_t c_{t-1} + i_t \tanh(U_c h_{t-1} + W_c x_t + b_c), \\ o_t &= \sigma(U_o h_{t-1} + W_o x_t + V_o c_t + b_o), \\ h_t &= o_t \tanh(c_t), \end{aligned} \tag{1}$$

where $\sigma(\cdot)$ is the logistic sigmoid function, and i_t, f_t, o_t, c_t are the input gate, forget gate, output gate and cell input activation vector, respectively. In each of them, there are corresponding input-to-hidden, hidden-to-output, and hidden-to-hidden matrices: $U_\bullet$, $V_\bullet$, $W_\bullet$ and the bias vector $b_\bullet$.

In Eq. (1), the context vector x_t is a dynamic representation of the relevant part of the social post input at time t. To calculate x_t, we introduce an attention weight $a_t[i], i = 1, \ldots, K$, corresponding to the feature extracted at different element positions in a tf-idf matrix d_t. Specifically, at each time stamp t, our model predicts a_{t+1}, a softmax over K positions, and y_t, a softmax over the binary class of rumors and non-rumors with an additional hidden layer with $sigmoid(\cdot)$ activations (see Fig. 3(c)). The location softmax [21] is thus, applied over the hidden states of the last LSTM layer to calculate a_{t+1}, the attention weight for the next input matrix d_{t+1}:

$$a_{t+1}[i] = P(L_{t+1} = i|h_t) = \frac{e^{W_i{}^\top h_t}}{\sum_{j=1}^{K} e^{W_j^\top h_t}} \qquad i \in 1, \ldots, K, \tag{2}$$

where $a_{t+1}[i]$ is the attention weight for the i-th element (word index) at time step $t+1$, W_i is the weight allocated to the i-th element in the feature space, and L_{t+1} represents the word index and takes 1-of-K values.



(a) The soft attention mechanism (b) Unfolded model structure

Fig. 3. (a) The attention module computes the current input x_t as an average of the tf-idf features weighted according to the attention softmax a_t. (b) At each time stamp, the proposed model takes the feature slice x_t as input and propagates x_t through stacked layers of LSTM and predicts next location weight a_{t+1}. The class label y_t is calculated at the last time step t.

The attention vector a_{t+1} consists of K weight scalars for each feature dimension, representing the importance attached to each word in the input matrix d_{t+1}. Our model is optimized to assign higher focus to words that are believed to be distinct in learning rumor/non-rumor representations. After calculating these weights, the **soft deterministic attention** mechanism [19] computes the

expected value of the input at the next time step x_{t+1} by taking weighted sums over the word matrix at different positions:

$$x_{t+1} = \mathbb{E}_{P(L_{t+1}|h_t)}[d_{t+1}] = \sum_{i=1}^{K} a_{t+1}[i] d_{t+1}[i], \tag{3}$$

where d_{t+1} is the input matrix at time step $t+1$ and $d_{t+1}[i]$ is the feature vector of the i-th position in the matrix d_{t+1}. Thus, Eq.(3) formulates a deterministic attention model by computing a soft attention weighted word vector $\sum_i a_{t+1}[i] d_{t+1}[i]$. This corresponds to feeding a soft-a-weighted context into the system, whilst the whole model is smooth and differential under the deterministic attention, and thus learning end-to-end is trivial by using standard back-propagation.

3.4 Loss Function and Model Training

In model training, we employ cross-entropy loss coupled with $l2$ regularization. The loss function is defined as follows:

$$\mathcal{L} = -\sum_{c=1}^{C} y_{t,c} \log \hat{y}_{t,c} + \gamma \phi^2, \tag{4}$$

where y_t is the one hot label represented by 0 and 1, $\hat{y}_t$ is the predicted binary class probabilities at the last time step t, $C = 2$ is the number of output classes (rumors or non-rumors), γ is the weight decay coefficient, and ϕ represents all the model parameters.

The cell state and the hidden state for LSTM are initialized using the input tf-idf matrices for faster convergence:

$$\begin{aligned} c_0 &= f_c\left(\frac{1}{\tau}\sum_{t=1}^{\tau}\left(\frac{1}{K}\sum_{i=1}^{K} d_t[i]\right)\right), \\ h_0 &= f_h\left(\frac{1}{\tau}\sum_{t=1}^{\tau}\left(\frac{1}{K}\sum_{i=1}^{K} d_t[i]\right)\right), \end{aligned} \tag{5}$$

where f_c and f_h are two multi-layer perceptrons, and τ is the number of time steps for each event sequence. These values are used to compute the first location softmax a_1 which determines the initial input x_1.

4 Experiments

In this section, we evaluate the performance of our proposed methodology in early rumor detection using real-world data collected from two different social media platforms.

4.1 Datasets

We use two public datasets published by [9]. The datasets are collected from Twitter[2] and Sina Weibo[3] respectively. Both of the datasets are organised at event-level with the ground truth verified via Snopes[4] and Sina Community Management Center[5]. In addition, we follow the criteria from [9] to manually gather 4 non-rumors from Twitter and 38 rumors from Weibo for comprehensive class balancing. Note for Tweet datasets, some posts are no longer available when we crawled those tweets, causing a 10% shrink on the scale of data compared with the original Twitter dataset and this is a main cause for a slight performance fluctuation compared with the results in other papers.

Table 1 gives statistical details of the two datasets. We observe that more than 76% of the users tend to repost the original news with very short comments to reflect their attitudes towards those news. As a consequence, the contents of the posts related to one event are mostly duplicate, which can be rather challenging for early rumor detection tasks.

Table 1. Statistical details of datasets. PPE stands for posts per event.

Dataset	Total users	Total posts	Events	Rumors	Non-rumors	Avg. PPE	Min. PPE	Max. PPE
Twitter	466,577	1,046,886	996	498	498	1,051	8	44,316
Weibo	2,755,491	3,814,329	4,702	2,351	2,351	811	10	59,318

4.2 Settings and Baselines

The model is implemented using Tensorflow[6]. All parameters are set using cross-validation. To generate the input variable-length post series, we set the amount of posts N for each time step as 5 and the minimum post series length Min as 2. We selected $K = 10{,}000$ top words for the construction tf-idf matrices. We randomly split our datasets with the ratio of 70%, 10% and 20% for training, validation and test respectively. We apply a three-layer LSTM model with descending amount of hidden states (specifically 1,024, 512 and 128). The learning rate is set as 0.001 and the γ is set to be 0.005. Our model is trained through back-propagation [27] algorithm, namely Adam [28]. We iterate the whole training process until the loss value converges.

We evaluate the effectiveness and efficiency of CallAtRumors by comparing with the following state-of-the-art approaches in terms of precision and recall:

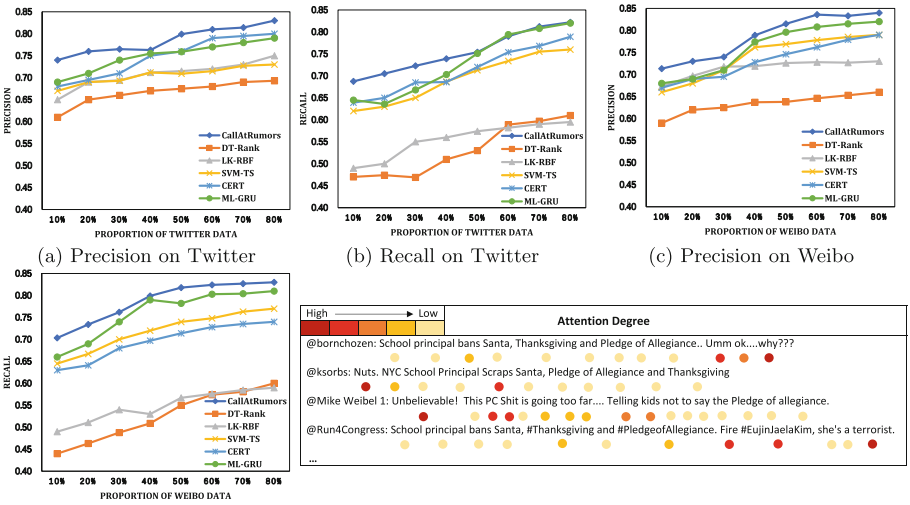
[2] www.twitter.com.
[3] www.weibo.com.
[4] www.snopes.com.
[5] http://service.account.weibo.com.
[6] https://www.tensorflow.org.

- DT-Rank [8]: This is a decision-tree based ranking model using enquiry phrases which is able to identify trending rumors by recasting the problem as finding entire clusters of posts whose topic is a disputed factual claim.
- SVM-TS [4]: SVM-TS can capture the temporal characteristics of from contents, users and propagation patterns based on the time series of rumors' lifecycle with time series modelling technique applied to incorporate carious social context information.
- LK-RBF [12]: We choose this link-based approach and combine it with the RBF (Radial Basis Function) kernel as a supervised classifier because it achieved the best performance in their experiments.
- ML-GRU [9]: This method utilizes basic recurrent neural networks for early rumor detection. Following the settings in their work, we choose the multi-layer GRU (gated recurrent unit) as it performs the best in the experiment.
- CERT [13]: This is a cross-topic emerging rumor detection model which can jointly cluster data, select features and train classifiers by using the abundant labeled data from prior rumors to facilitate the detection of an emerging rumor.

4.3 Effectiveness and Earliness Analysis

In this experiment, we take different ratios of the posts starting from the first post within all events for model training, ranging from 10% to 80% in order to test how early CallAtRumors can detect rumors successfully when there are limited amount of posts available. Through incrementally adding training data in the chronological order, we are able to estimate the time that our method can detect emerging rumors. The results on earliness are shown in Fig. 4. At the early stage with 10% to 60% training data, CallAtRumors outperforms four comparative methods by a noticeable margin. In particular, compared with the most relevant method of ML-GRU, as the data proportion ranging from 10% to 20%, CallAtRumors outperforms ML-GRU by 5% on precision and 4% on recall on both Twitter and Weibo datasets. The result shows that attention mechanism is more effective in early stage detection by focusing on the most distinct features in advance. With more data applied into test, all methods are approaching their best performance. For Twitter dataset and Weibo Dataset with highly noticable duplicate contents in each event, our method starts with 74.02% and 71.73% in precision while 68.75% and 70.34% in recall, which means an average time lag of 20.47 h after the emerge of one event. This result is promising because the average report time over the rumors given by Snopes and Sina Community Management Center is 54 h and 72 h respectively [9], and we can save much manual effort with the help of our deep attention based early rumor detection technique.

Apart from numerical results, Fig. 4(e) visualises the varied attention effects on a detected rumor. Different color degrees reflect various attention degrees paid to each word in a post. In the rumor "School Principal Eujin Jaela Kim banned the Pledge of Allegiance, Santa and Thanksgiving", most of the vocabularies closely connected with the event itself are given less attention weight than words



(a) Precision on Twitter (b) Recall on Twitter (c) Precision on Weibo

(d) Recall on Weibo (e)Visualization of Attention on an Early Detected Rumor

Fig. 4. The charts in (a)–(d) reveal the performance for all methods with accumulative training data size. The effect of attention mechanism is visualized via (e). (Color figure online)

expressing users' doubting, esquiring and anger caused by the rumor. Despite the massive duplication from users' comments, by implementing textual attention mechanism, CallAtRumors is able to lay more emphasis on discriminative words, thus guaranteeing high performance in such case.

5 Conclusion

Rumor detection on social media is time-sensitive because it is hard to eliminate the vicious impact in its late period of diffusion as rumors can spread quickly and broadly. In this paper, we introduce CallAtRumors, a novel recurrent neural network model based on soft attention mechanism to automatically carry out early rumor detection by learning latent representations from the sequential social posts. We conducted experiments with five state-of-the-art rumor detection methods to illustrate that CallAtRumors is sensitive to distinguishable words, thus outperforming the competitors even when textual feature is sparse at the beginning stage of a rumor. In our future work, it would be appealing to investigate more complexed feature from opinion clustering results [29] and user behavior patterns [30] with our deep attention model to further improve the early detection performance.

References

1. Zimbra, D., Ghiassi, M., Lee, S.: Brand-related Twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks. In: 2016 49th Hawaii International Conference on System Sciences (HICSS), pp. 1930–1938. IEEE (2016)
2. Zafarani, R., Liu, H.: 10 bits of surprise: detecting malicious users with minimum information. In: CIKM, pp. 423–431. ACM (2015)
3. Wu, K., Yang, S., Zhu, K.Q.: False rumors detection on Sina Weibo by propagation structures. In: ICDE, pp. 651–662. IEEE (2015)
4. Ma, J., Gao, W., Wei, Z., Lu, Y., Wong, K.F.: Detect rumors using time series of social context information on microblogging websites. In: CIKM, pp. 1751–1754. ACM (2015)
5. Liu, X., Nourbakhsh, A., Li, Q., Fang, R., Shah, S.: Real-time rumor debunking on Twitter. In: CIKM, pp. 1867–1870. ACM (2015)
6. Rayana, S., Akoglu, L.: Collective opinion spam detection using active inference. In: Proceedings of the 2016 SIAM International Conference on Data Mining, pp. 630–638. SIAM (2016)
7. Rayana, S., Akoglu, L.: Collective opinion spam detection: bridging review networks and metadata. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985–994. ACM (2015)
8. Zhao, Z., Resnick, P., Mei, Q.: Enquiring minds: early detection of rumors in social media from enquiry posts. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1395–1405. ACM (2015)
9. Ma, J., et al.: Detecting rumors from microblogs with recurrent neural networks. In: Proceedings of IJCAI (2016)
10. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on Twitter. In: Proceedings of the 20th International Conference on World Wide Web, pp. 675–684. ACM (2011)
11. Hu, X., Tang, J., Gao, H., Liu, H.: Social spammer detection with sentiment information. In: ICDM, pp. 180–189. IEEE (2014)
12. Sampson, J., Morstatter, F., Wu, L., Liu, H.: Leveraging the implicit structure within social media for emergent rumor detection. In: CIKM, pp. 2377–2382. ACM (2016)
13. Wu, L., Li, J., Hu, X., Liu, H.: Gleaning wisdom from the past: early detection of emerging rumors in social media. In: SDM (2016)
14. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of NAACL-HLT, pp. 1480–1489 (2016)
15. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
16. Wu, L., Wang, Y., Li, X., Gao, J.: What-and-where to match: deep spatially multiplicative integration networks for person re-identification. Pattern Recognit. **76**, 727–738 (2018)
17. Wu, L., Wang, Y.: Where to focus: deep attention-based spatially recurrent bilinear networks for fine-grained visual recognition. arXiv preprint arXiv:1709.05769 (2017)
18. Wu, L., Wang, Y., Gao, J., Li, X.: Deep adaptive feature embedding with local sample distributions for person re-identification. Pattern Recognit. **73**, 275–288 (2018)

19. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
20. Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., Hinton, G.: Grammar as a foreign language. In: Advances in Neural Information Processing Systems, pp. 2773–2781 (2015)
21. Sharma, S., Kiros, R., Salakhutdinov, R.: Action recognition using visual attention. arXiv preprint arXiv:1511.04119 (2015)
22. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
23. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)
24. Wu, L., Haynes, M., Smith, A., Chen, T., Li, X.: Generating life course trajectory sequences with recurrent neural networks and application to early detection of social disadvantage. In: Cong, G., Peng, W.-C., Zhang, W.E., Li, C., Sun, A. (eds.) ADMA 2017. LNCS (LNAI), vol. 10604, pp. 225–242. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69179-4_16
25. Chen, W., et al.: EEG-based motion intention recognition via multi-task RNNs. In: Proceedings of the 2018 SIAM International Conference on Data Mining. SIAM (2018)
26. Zhang, D., Yao, L., Zhang, X., Wang, S., Chen, W., Boots, R.: EEG-based intention recognition from spatio-temporal representations via cascade and parallel convolutional recurrent neural networks. arXiv preprint arXiv:1708.06578 (2017)
27. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**(Aug), 2493–2537 (2011)
28. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
29. Chen, H., Yin, H., Li, X., Wang, M., Chen, W., Chen, T.: People opinion topic model: opinion based user clustering in social networks. In: WWW Companion, pp. 1353–1359. International World Wide Web Conferences Steering Committee (2017)
30. Yin, H., Chen, H., Sun, X., Wang, H., Wang, Y., Nguyen, Q.V.H.: SPTF: a scalable probabilistic tensor factorization model for semantic-aware behavior prediction. In: ICDM. IEEE (2017)