

Rumor detection based on graph attention network

Yuwei Lv¹, Xuemei Sun^{1,*}, Yonggang Wen³, and Wanru Wang²

¹School of Computer Science and Technology, Tiangong University, Tianjin, China

²School of Software, Tiangong University, Tianjin, China

³Boustead College, Tianjin University of Commerce, Tianjin, China

Abstract. At present, most of the existing rumor detection methods focus on the learning and fusion of various features, but due to the complexity of language, these models often rarely consider the relationship between parts of speech. This paper uses graph attention neural network model to learn text features and syntactic relations to solve this problem. It uses node attention collection text feature and edge attention collection relationship feature for syntactic dependency tree, and node attention and edge attention to enhance each other. Finally, the proposed method is verified on Twitter and Weibo data sets. The experimental results show that the proposed method has greatly improved the early detection and accuracy of rumors.

Keywords: Rumor detection, Graph attention neural network, Syntactic relations, Deep learning.

1 Introduction

Rumor detection is to identify whether the information on the social media platform is a rumor. It has the characteristics of fast propagation, difficult identification and great harm. It has a wide application prospect in social media and other fields. Because rumors usually spread quickly, they may reach a large number of users before experts or authoritative organizations draw conclusions. Therefore, there is an urgent need to automatically detect rumors in the early stage in order to minimize its negative impact

The earliest research on rumor detection was based on traditional machine learning[1-2], that is, rumor detection is transformed into a binary classification problem. Firstly, the rumor features are extracted, then the machine learning model is used for modeling, and finally the rumor features are input into the model for training to realize rumor detection. Literature [3] proposed a model based on decision tree classifier, which uses the phrases of query and correction classes extracted from the message to cluster, and then constructs a decision tree according to the statistical characteristics to realize rumor detection.

Most of the existing rumor detection methods focus on learning and integrating various features for detection[4]. These methods are easy to lose time and other information. Some scholars use cyclic neural network to realize rumor detection, considering the forwarding

* Corresponding author: Sunxuemei@tiangong.edu.cn

comment information and timing characteristics of the message in the transmission process, but these methods do not save the hierarchical information of the transmission structure. Other scholars[5-6] use bidirectional graph convolution neural network to obtain the propagation structure. This method considers a top-down rumor propagation and bottom-up rumor diffusion to capture and identify rumors. This method does not consider the dynamic change process of propagation structure.

In order to improve the accuracy of rumor detection, this paper proposes a graph neural network model, which uses the dependency parsing package to obtain the syntactic dependency tree, and uses the node attention and edge attention of the graph attention model to collect text features and syntactic dependencies respectively. It is tested on the public standard Twitter and Weibo data sets, which greatly improves the early detection and accuracy of rumors.

2 Proposed method

2.1 Graph neural network summary

With the vigorous development of deep learning technology, GNN has achieved great success in the representation and learning of graph structure data [7]. Generally speaking, most existing GNN models follow the neighborhood aggregation strategy. A GNN layer can be defined as shown in Formula 1:

$$h_i^l = \text{AGGR}(h_i^{l-1}, \{h_j^{l-1} | \forall j \in N_i\}) \tag{1}$$

Where h_i^l is the node representation of node i in layer L , and N_i is the local domain set of node i . AGGR is an aggregate function of GNN and has many implementation methods. GNN has the ability to capture the long-distance interaction between entities, and has good performance in text classification. Most of the current methods are to establish a corpus level document word map and try to classify documents through semi supervised node classification. Although they have achieved success, most of the existing methods have computational defects. At the same time, the simplicity of using graphs to model text is greatly limited. Therefore, how to improve the computing power of model expression is an important and must be solved task.

2.2 Graph attention network

The input of this model is the syntactic dependency tree generated by the syntactic dependency parsing package. As shown in Figure 1, its node is the text of the news, and the edge is the dependency. After encoding it, it passes through an LSTM layer, and then uses the graph attention network model to learn it. The node attention learns its text features, the edge attention learns the dependency, enhances each other, and finally classifies the news. The overall framework is shown in Figure 2.



Fig. 1. Syntactic dependency tree.

The graph shows that the neural network aggregates the representation of neighborhood nodes along the dependent path. However, this process does not take into account

dependencies, which may lose some important dependency information. Intuitively, neighborhood nodes with different dependencies should have different effects. This model uses additional relationship headers to expand the original graph attention network, and uses these relationship headers as relationship level gates to control the information flow from neighborhood nodes.. In order to support the representation learning of the constructed dependency tree, this paper uses GAT model, which uses node attention aggregation, neighborhood node representation and edge attention aggregation domain dependency respectively, and iteratively updates each node representation, such as formula (2), (4).

$$f_{node_i}^{l+1} = \parallel_{k=1}^K \sum_{j \in N_i} a_{ij}^{lk} W_k^l h_j^l \quad (2)$$

$\parallel_{k=1}^K$ represents the splicing of vectors from X_1 to X_K . W_k^l is a trainable weight matrix. a_{ij} indicates the attention coefficient of node j , and the calculation is as shown in formula (3):

$$a_{ij} = \text{attention}(\text{node}_i, \text{node}_j) \quad (3)$$

This model uses edge attention to highlight the next layer representation of learning node i . This process can be expressed as:

$$h_{edge_i}^{l+1} = \parallel_{m=1}^M \sum_{j \in N_i} B_{ij}^{lk} W_k^l f_j^l \quad (4)$$

W_k^l is a trainable weight matrix. B_{ij}^{lk} indicates the attention coefficient of the upper edge j of node i , which can be calculated by formula (5) (6).

$$B_{ij}^{lm} = \frac{\exp(a^T v_j^{lm})}{\sum_{j \in N_i} \exp(a^T v_j^{lm})} \quad (5)$$

$$v_j = \text{LeakRelu}([w_2 f_j^{l+1} \parallel w_1 h_j^l]) \quad (6)$$

Where a^T is a weight vector for calculating dependencies, and \parallel is a join operation. This model can not only capture high-order word interaction, but also learn the dependency of dependency tree.

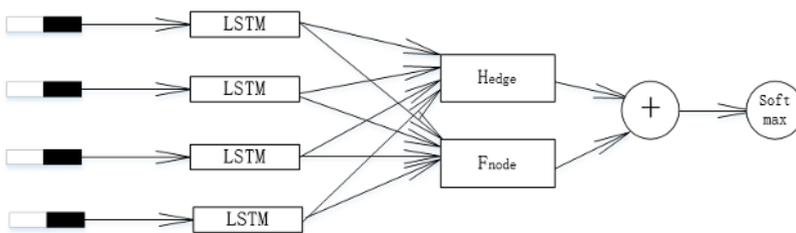


Fig. 2. Overall framework of graph attention neural network.

3 Experiment and analysis

3.1 Datasets

The three data sets of Weibo [8], twitter15 [9] and twitter 16 [9] are all from real social media. The Weibo data set contains two binary Tags: false rumors (f) and true rumors (T).

The twitter15 and twitter16 data sets contain four Tags: non rumors (n), false rumors (F), true rumors (T) and unverified rumors (U). The tags of each event on Lang Weibo are marked according to the Sina community management center to record various error messages. Each event tag of twitter 15 and twitter 16 is marked according to the authenticity tag of rumor website articles. The statistical data of these three data sets are shown in Table 1.

Table 1. The statistical data of three data sets.

Statistics	Twitter15	Twitter16	weibo
User#	331,612	204,820	3,805,656
Posts#	26,663	173,487	2,746,818
Events#	1490	818	4664
True Rumors#	374	205	2351
False Rumors#	370	205	2313
Unverified Rumors#	374	203	0
Non-rumors#	372	205	0
Avg.hours per event	1,337hours	848hours	2460.7hours
Avg.# of posts per event	223	851	816
Max # of posts per event	1,768	2,765	223
Min # of post per event	55	81	251

3.2 Experimental environment

The syntax dependency tree is processed by the method in [10] to form a. The random gradient descent method is used to update the parameters of the model, and the Adam algorithm is used to optimize the model. The hyperparameter of the balance loss function is = 0.1. This experiment is carried out in Python 3 6, pytorch1. 2.0 GPU. The learning rate is set to 0.0005 and the batch processing is set to 128. The data set is randomly divided into five parts, and 5x cross validation is carried out to obtain more generalized results. The training process is carried out for 200 iterations. When the verification loss stops for at least 10 times, the application ends the training.

3.3 Experimental result

We compare some traditional machine learning and deep learning models with the model proposed in this paper. The experimental results of these three data sets using different methods are shown in Table 2.

Table 2. The experimental results of these three data sets.

Method	Weibo			Twitter15					Twitter16				
	ACC	F	T	ACC	N	F	T	U	ACC	N	F	T	U
		F1	F1		F1	F1	F1	F1		F1	F1	F1	F1
SCM-RBF	87.9	70.8	65.1	31.8	22.5	8.2	45.5	21.8	55.3	67.0	8.5	11.7	36.1
SVM-TS	88.5	93.8	5.9	45.4	79.6	47.2	40.4	48.3	57.4	75.5	42.0	57.1	52.6
Tanh-RNN	87.3	86.1	88.4	82.7	82.5	83.0	81.2	82.9	82.7	82.1	83.4	81.6	82.3
RvNN	90.8	91.2	90.4	72.3	68.2	75.8	82.1	65.4	73.7	66.2	74.3	83.5	70.8
GRU-1	90.8	90.4	91.3	86.4	84.5	80.2	87.8	80.3	86.4	84.3	80.5	87.6	80.5
our method	94.1	93.1	93	85.6	87.2	83.4	90.1	83.7	85.4	81.8	83.4	90.4	83.2

For the Weibo dataset, the accuracy and F1 measurements in two categories were compared. For two Twitter datasets, the accuracy and F1 in four categories were compared. Through comparison, it is found that the performance of deep learning method is

significantly better than that of manual feature extraction method. Because the deep learning method can learn the high-level representation of rumors to capture effective features. This shows the importance and necessity of studying deep learning in rumor detection. The proposed method is superior to GRU-1 method in all performance aspects, which shows the effectiveness of rumor detection combined with syntactic information.

3.4 Early rumor detection

Early rumor detection is to detect rumors in the early stage of transmission, which is another important index to evaluate the quality of this method. First, we build an early detection task. We set the deadline for rumor detection, and only use the posts published before the deadline to evaluate the accuracy of the proposed method and mainstream methods.

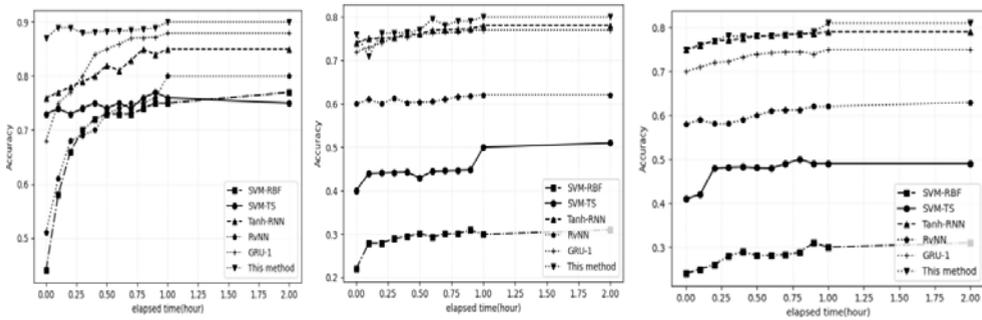


Fig. 3. Early detection results of rumors, from left to right are Weibo, Twitter 15 and Twitter 16.

The early performance of all methods fluctuates more or less. This is because with the spread of posts, there is more semantic and structural information, and noise information increases. The method of this model can get good results after being published on twitter data set for about one hour, which shows that the performance of this model in the early detection of rumors is very superior.

4 Conclusion

In order to improve the accuracy and timeliness of rumor detection, the author proposes a rumor detection method based on a graph attention network model, which uses two attention mechanisms to aggregate text features and syntactic dependencies respectively, and finally enhance each other. The experimental results show that this method does not urgently improve the accuracy of rumor detection, and the effect is very good in early rumor detection. In the next work, we will pay more attention to the external factors affecting rumor detection. At the same time, applying graph neural network to rumor detection is also a promising direction.

References

1. Yang F,Liu Y,Yu X,et al.Automatic detection of rumor on sina weibo[C].Processdings of the ACM SIGKDD workshop on Mining Data Semantics.2012:1-7.
2. Kwon S,Cha M,Jung K,et al;Prominent featurea of rumor propagation in online social media[C].2013 IEEE 12th international conference on Data Mining.2013:1103-1108.

3. Zhao Z, Resnick P, Mei Q. Enquiring minds: Early detection of rumors in social media from enquiry posts[C]. *Proceedings of the 24th international conference on world wide web*.2015: 1395-1405.
4. Guo c, Cao j, Zhang x .Exploiting Emotions for Fake News Detection on Social Media[j], *Arxiv preprint arxiv: 1903. 01728*,2019.
5. Yu F, Liu Q, Wu s, et al. A convolutional approach for misinformation identification[J].2017
6. Chen T, Wu L, Li X. Call attention to rumors: Deep attention based Recurrent Neural networks for early rumor detection [J]. *Arxiv preprint arxiv: 1704. 0597352017*.
7. Zhang Z, Zhuang F, Zhu H, et al. Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion[J]. *Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(05): 9612–9619*.
8. Wu, K, Yang, S, Zhu. False rumors detection on sina weibo by propagation structures[C]. In *2015 IEEE 31st international conference on data engineering*, 651–662. IEEE.
9. Ma j, Gao W, Mitra P, et al. Detection Rumors from Microblogs with Recurrent Neural Networks[C]*Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York: IJCAI, 2016: 3818–3824*
10. Wang K, Shen W, Yang Y. Relational Graph Attention Network for Aspect-based Sentiment Analysis[C], *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics(ACL)2020:3229--3238*.
11. Ma j,Gao W, Mitra Pl. Detection Rumors from Microblogs with Recurrent Neural Networks[C]*Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York: IJCAI, 2016: 3818 - 3824*
12. Chen T,Wu L,Li X,et al. Call attention to rumors: Deep attention based Recurrent Neural networks for early rumor detection[J]. *Arxiv preprint arxiv: 1704. 0597352017*.
13. Wu K, Yang S. False rumors detection on sina weibo by propagation structures[C]. In *2015 IEEE 31st international conference on data engineering*, 651–662. IEEE.
14. Ma J, Gao W, Wei Z. Detect rumors using time series of social context information on microblogging websites[C]. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1751– 1754.
15. Ma J, Gao W. Rumor detection on twitter with tree-structured recursive neural networks[C]. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 1980–1989.