Rumor Detection with Hierarchical Social Attention Network

Han Guo[1,2], Juan Cao[1,2], Yazi Zhang[1,2], Junbo Guo[1] and Jintao Li[1]
[1]Key Laboratory of Intelligent Information Processing & Center for Advanced Computing Research, Institute of Computing Technology, CAS, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
{guohan,caojuan,zhangyazi,guojunbo,jtli}@ict.ac.cn

ABSTRACT

Microblogs have become one of the most popular communication tools for news sharing. However, due to its openness and lack of supervision, rumors could also be easily posted and propagated in microblogs, which could have serious consequences. Therefore, tools for automatic detection and verification of rumors in microblogs are very valuable. In this paper, we propose a novel hierarchical neural network combined with social information (HSA-BLSTM) for rumor detection. At first a hierarchical bi-directional long short-term memory model is built for representation learning. Then, the social contexts are incorporated into the network via attention mechanism. Test this model on two real-world datasets from Weibo and Twitter demonstrate outstanding performance in both rumor detection and early detection scenarios.

CCS CONCEPTS

• **Information systems → Multimedia information systems**; *Social networks*;

KEYWORDS

Rumor Detection; Recurrent Neural Network; Attention Mechanism

ACM Reference Format:
Han Guo[1,2], Juan Cao[1,2], Yazi Zhang[1,2], Junbo Guo[1] and Jintao Li[1]. 2018. Rumor Detection with Hierarchical Social Attention Network. In *2018 ACM Conference on Information and Knowledge Management (CIKM'18), October 22–26, 2018, Torino, Italy,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Social media has become a major source for news. It, however, also fostered rapid spread of rumors carrying unverified or even fake information could cause panic or other serious consequences. For instance, during the 2016 U.S. presidential election, Twitter became a social medium for candidates and voters to share news and opinions. Meantime, it also created a fertile soil for the emergence and propagation of rumors due to its openness. According to a recent work [13], 529 rumors about Donald Trump and Hillary Clinton were released during the election, which defamed the candidates and misled the voters with fake stories or malicious comments.

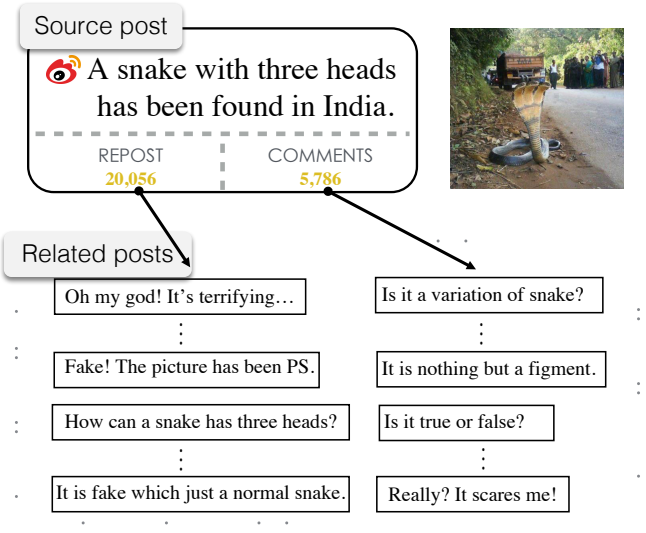

Figure 1: An example of rumor event on Weibo.

Recently, rumors on social media have become a major concern. Several organizations, such as Snopes[1] and the Community Management Center of Weibo[2], aim to debunk rumors emerging on the social media. Their judgment are mainly made by editor staffs. However, manually collecting and investigating rumors are quite time-consuming and automatic rumor detection are very useful when dealing with massive real-time rumor posts.

A rumor is defined as a story or statement in general circulation without confirmation or certainty to facts [8]. For example, Figure 1 shows a rumor about a snake with three heads from Weibo, with a source post and a set of related posts including reposts and comments (bottom-left) will also emerge on the social network. Besides the source post, crucial opinions and sentiments conveyed from the related posts are also helpful for rumor detection. Therefore, we follow the convention and define the source post and all its related posts as a rumor event. Early studies detect rumors based on the source post and its related comments and reposts, but ignore its hierarchical structure, where an event can be represented at sub-event, post, and word level.

Recently, deep learning models are proposed to predict the credibility of an event [6, 18]. By modeling the posts in an event as time-series, recurrent neural network (RNN) can learn the latent

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

[1]www.snopes.com
[2]http://service.account.weibo.com

textual representation automatically. Nonetheless, these methods only focus on the semantic information and ignore the social characteristics in the rumor events as not all users (or posts) contribute equally under certain social context. Hierarchical characteristic of an event is also neglected in these works.

This paper proposes a novel hierarchical network with social attention for rumor detection(HSA-BLSTM). We first model the rumor event as hierarchical time-series containing different semantic levels of information. More specifically, an event is divided into several sub-events containing several posts. Each post is further segmented into several words. A structured event is then fed into our hierarchical Bi-LSTM network. Social features are utilized as another clue to identify the prominent part of rumor. We implement these features via the attention mechanism in Bi-LSTM to obtain an accurate representation for rumor detection. Extensive experiments demonstrate that our model outperforms several state-of-the-arts models. Meanwhile, outstanding performance is also obtained in early detection of rumors, which is quite crucial in preventing rumor propagation in real world applications. The contributions of this paper can be summarized as follows:

1) We propose a hierarchical social attention network for rumor detection.

2) We leverage social features and textual information to address the challenges in rumor detection. Our proposed model incorporates multi-modal information via the attention mechanism.

3) Extensive experiments validate the effectiveness in early rumor detection. By using only the posts released at the very early stage of propagation, our model outperforms all other state-of-the-arts models by a large margin.

2 RELATED WORK

Automatic rumor detection can be regarded as a binary classification problem, which need to apply feature extraction and machine learning algorithms.

2.1 Method for Rumor Detection

Castillo *et al.* focus on information credibility on Twitter by extracting four kinds of hand-crafted features including message-based features, user-based features, topic-based features and propagation features [4]. Subsequently, similar studies [19, 24] are performed to detect rumors based on a wide range of social features. For example, in the feature selection process, Yang *et al.* extract the location of event and client program as new features to classify a post on Weibo [30]; Jin *et al.* pay more attention on images attached to the posts and propose visual and statistical features to detect fake news [15]. Rather than using different sets of features for rumor detection, Zhao *et al.* select some regular expressions as rumor patterns to identify rumor tweets [32]. Cao *et al.* consider that making prediction based on a short message is unreliable[3], observing that each tweet contains videos/images in the dataset, and tweets containing the same videos/images are rather independent but have strong relations with each other. In other words, these tweets attend to have same topics and credibilities, so they cluster these tweets and build a topic-level classifier. Zhou *et al.* build a real-time news verification system on Sina Weibo [33]. They extract some keywords of rumor events and gather related microblogs through a distributed data acquisition system.

In addition to applying supervised algorithms, Jin *et al.* propose a novel method that discovers conflicting viewpoints [14]. Based on identified the viewpoints in tweets, they build a credibility propagation network of messages linked with supporting or opposing relations. The credibility propagation on the network generates the final evaluation result for news with iterative deduction.

Compared with traditional hand-crafted features, deep neural networks can learn accurate representations for visual and textual content. Specifically, recurrent neural networks have shown their power on modeling variable-length sequential information such as time series and sentences [12, 27]. Wan *et al.* introduce an RNN model to learn the similarity score of two sentences [29]. Notably, in the task of detecting rumors, Ma *et al.* propose a GRU network that regards the social context information of a rumor event as a variable-length time series, which achieves a good performance on two real-world microblog datasets [18]. Chen *et al.* treat a rumor message as an exception in users' social behavior [7]. By combining users' historical messages, the rumor messages will be classified through detecting 'noise points'.

2.2 Attention Mechanism

In 2014, Mnih et al. [22] introduced the "attention" mechanism in recurrent neural network to classify images. This work simulates the process of people observing images. In most cases, people are not concerned with every pixel in a given image, but focus on some specific parts according to the requirements. Moreover, people will also learn which part of the image should be paid more attention from the experience of the previous observation. In the same year, Bahdanau et al. [1] introduced the attention mechanism to the natural language processing field for the first time, and conducted translation and alignment procedure at the same time on the Machine Translation task. The work uses two recurrent neural network models, one that encodes the to the hidden vector, and then decodes it with another recurrent neural network. The key technique is to learn the "attention" matrix. By giving different weights to the words in the source language to predict the target language, the source language can closely connect to the target language. At the same time, by visualizing the attention mechanism matrix, we can see which part of the data that the neural networks pay close attention to when performing tasks. The core of the attention mechanism is a process of automatic learning weight matrix. By learning the probability distribution of weights through the softmax function, two kinds of modules can be connected by weighting. The core method is calculated as follow:

$$\alpha_{x,y} = \mathcal{F}(m_x, m_y) = \frac{exp(f(m_x, m_y))}{\sum_Y exp(f(m_x, m_Y))} \tag{1}$$

where m_x represents one module and m_y represents another one, $\alpha_{x,y}$ represents the weight parameter, $\mathcal{F}$ is the connection function and f represents the function that calculated the relationship between two modules, it could be dot product or a perceptron.

Attention mechanisms are able to learn important semantic components in some natural language processing tasks [27, 31]. Chen *et al.* propose an attention-based neural network with user and

product embedded for sentiment classification [5]. For rumor detection, Chen *et al.* apply attention mechanism to each word in a post, aiming to learn valuable semantics in sequential inputs [6].

Motivated by the great success in applications of attention-based neural networks, we propose an efficient rumor detection model with hand-crafted social features serving as attention in different semantic levels. As social features contain important clues, our proposed model is expected to improve the performance of rumor detection.

3 METHODOLOGY

In this section, we first briefly revisit bidirectional LSTM for completeness. Then, we present the formulations of event-level rumor detection. Finally, we show how to obtain the representation of each semantic level with social features via attention mechanism.

3.1 Long Short-Term Memory (LSTM) Networks

The standard LSTM are able to process input sequences with variable-length via recursive operation. Some commonly-used variants of the basic LSTM are proposed in literatures [10, 28]. In this paper, we leverage bidirectional LSTM (Bi-LSTM) as an encoder to learn the latent representation of sequential data. For completeness, we present a quick review of generic LSTM and bidirectional LSTM in this section.

3.1.1 Generic LSTM. Given an input sequence $\{x_1, x_2, \ldots, x_t\}$ with length T, a basic recurrent neural network (RNN) predicts the output sequence $\{y_1, y_2, \ldots, y_t\}$, also calculates the hidden layer $\{h_1, h_2, \ldots, h_t\}$ with a recurrent unit as follow:

$$h_t = \mathcal{H}(h_{t-1}, x_t) \tag{2}$$

where h_{t-1} is the last hidden state and x_t is the current input, $\mathcal{H}(\cdot)$ can be an activation function or other hidden layer function which takes h_{t-1} and x_t as inputs to produce current hidden state h_t. LSTM can solve the gradient vanishing problem [2, 23] in learning long-term dependencies. Specifically, the LSTM cell c is controlled by a set of gates: an input gate i, an output gate o and a forget gate f. Let W_* represent the parameters for the corresponding gates and b_* are the bias terms. $\odot$ denotes the element-wise multiplication between two vectors. ϕ is the hyperbolic tangent function and σ is the logistic sigmoid function. The precise form of LSTM cell is described as follows [9]:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{3}$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{4}$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{5}$$
$$g_t = \phi(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{6}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{7}$$
$$h_t = o_t \odot \phi(c_t) \tag{8}$$

3.1.2 Bidirectional LSTM. A obvious limitation of basic LSTM is that it is only able to make use of previous context [25]. Bidirectional LSTM (Bi-LSTM) overcomes the issue by processing the data in both directions with forward hidden layer and backward hidden layer, also obtains accurate representations of the input data in each unit. The output of the memory cell is computed as follows [11]:

$$\overrightarrow{h}_t = \mathcal{LSTM}(\overrightarrow{h}_{t-1}, x_t) \tag{9}$$
$$\overleftarrow{h}_t = \mathcal{LSTM}(\overleftarrow{h}_{t+1}, x_t) \tag{10}$$

where $\mathcal{LSTM}(\cdot)$ represents the generic LSTM cell. By iterating the forward layer from $t = 1$ to T, the backward layer from $t = T$ to 1, Bi-LSTM generates the forward hidden sequence $\overrightarrow{h}$ and the backward hidden sequence $\overleftarrow{h}$. Then the output layer will be updated with the linear combination or concatenation of $\overrightarrow{h}$ and $\overleftarrow{h}$.

3.2 Formulation

In this paper, we aim to classify a group of posts which constitute an event. In other words, an event contains a source post attached with a number of related posts (reposts and comments). It is difficult to predict the credibility of a single post, because the source post only contains very limited information in a few sentences. We focus on detecting rumor on event level rather than post level.

Formally, an event e contains the source post and its related posts. However, a popular event may include a huge amount of posts in a long time interval. For computational efficiency, we follow previous work [6] and divide the posts in e into different time intervals and each interval is considered as a subevent. Therefore, $e = [u_1, u_2, \ldots, u_m]$ contains m subevents; the i-th subevent u_i consists of n posts $u_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,n}]$; and each post $p_{i,j}$ consists of l_{ij} words $p_{i,j} = [w^1_{i,j}, w^2_{i,j}, \ldots, w^{l_{ij}}_{i,j}]$. Besides the hierarchical textual information, we also extract 22 social features represented as $s \in \mathbb{R}^{22}$ (see Section 3.5 for details) representing the social context of e. Formally, rumor detection on event-level aims to learn a projection $F(e, s) \rightarrow \{0, 1\}$, where 0 and 1 indicate labels for non-rumor and rumor, respectively.

3.3 Hierarchical Network with Social Attention

To obtain the representation of an event, we model e via a hierarchical structure. From the prospective of different semantic levels, the proposed model consists of several parts: word level part containing a Bi-LSTM layer and attention layer, post level part containing a Bi-LSTM layer and a social feature attention layer, subevent level part containing a Bi-LSTM layer and a social feature attention layer. The overall framework of the proposed model is shown in Figure 2. The reason we choose bi-directional structure is that it encodes rumor data in both forward and backward directions. Due to different writing styles, some posts are better processed in backward direction. The concatenated hidden states are more robust on generating the representation for posts.

Word level: We first embed every word into a low-dimensional semantic space. That is, each word is embedded as a word vector $w^k_{i,j} \in \mathbb{R}^d$. At each time step, by feeding a word vector to Bi-LSTM, the forward LSTM cell generates a hidden state $\overrightarrow{h^k_{i,j}}$ and the backward LSTM cell generates a hidden state $\overleftarrow{h^k_{i,j}}$. We obtain the final hidden state by concatenating two hidden states:

$$h^k_{i,j} = \overrightarrow{h^k_{i,j}} \oplus \overleftarrow{h^k_{i,j}}, \tag{11}$$

where $\oplus$ denotes the concatenation operation.

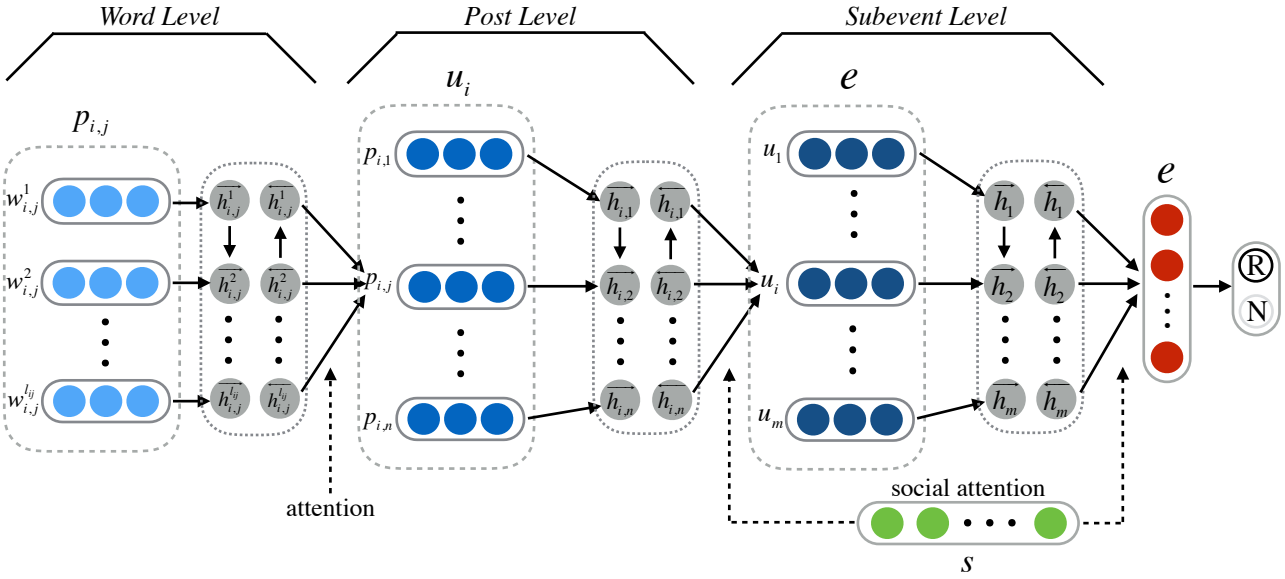

Figure 2: The framework of Hierarchical Networks with Social Attention (HSA-BLSTM) for rumor detection.

Obviously, not all words contribute equally to the representation of a post, some words are more important than others. Rather than feeding the hidden sequence to the average pooling layer, we highlight those valuable words via attention mechanism and the enhanced post representation is computed as follows:

$$\mathcal{F}_w(h^k_{i,j}) = {v_w}^T \phi(W_{hw} h^k_{i,j} + b_w), \tag{12}$$

$$\alpha^k_{i,j} = \frac{exp(\mathcal{F}_w(h^k_{i,j}))}{\sum_k exp(\mathcal{F}_w(h^k_{i,j}))}, \tag{13}$$

$$p_{i,j} = \sum_k \alpha^k_{i,j} h^k_{i,j}, \tag{14}$$

where W_{hw} denotes the weight matrices and b_w is the bias term, ${v_w}^T$ denotes a transposed weight vector and $\mathcal{F}_w$ is the score function in word-level, which measures the significance of each word. Afterwards, we obtain the normalized weight of k-th word $\alpha^k_{i,j}$ via a softmax function and compute the representation of j-th post in i-th subevent $p_{i,j}$ as a weighted sum of hidden states. The matrices W_{hw} and the vector v_w are randomly initialized and jointly learned during the training process.

Post level: Given the post vector $p_{i,j}$, we can obtain the representation of subevent u_i. After encoding the post vectors $p_{i,j}$ by Bi-LSTM, we obtain the hidden state $h_{i,j}$. Then we compute the subevent vector u_i by incorporating social feature vector s to attention mechanism as follows:

$$\mathcal{F}_p(h_{i,j}) = {v_p}^T \phi(W_{hp} h_{i,j} + W_{sp} s + b_p), \tag{15}$$

$$\beta_{i,j} = \frac{exp(\mathcal{F}_p(h_{i,j}))}{\sum_j exp(\mathcal{F}_p(h_{i,j}))}, \tag{16}$$

$$u_i = \sum_j \beta_{i,j} h_{i,j}, \tag{17}$$

where W_{hp} and W_{sp} are the weight matrices, $h_{i,j}$ is the hidden state in post level which is computed same as that in word level, the score function in post level $\mathcal{F}_p$ measures the importance of each post and we obtain the weight $\beta_{i,j}$. Note that we start fusing social features to the network in post level, especially in the operation of generating weight parameters. We assume that global social information could help the attention mechanism find several significant posts in detection task.

Table 1: Summary of social features adopted in this work.

Category	Features
User Profile	U1: Fraction of users with a self description
	U2: Fraction of users with an avatar
	U3: Fraction of verified users
	U4: Average reputation score
	U5: Average number of followers
	U6: Average number of friends
	U7: Average time length of registration
	U8: Average number of post per user
Propagation	P1: Fraction of tweets being reposted
	P2: Average comments
	P3: Average reposts
Post Texts	T1: Total number of posts
	T2: Average length of posts
	T3: Average sentiment score
	T4: Fraction of posts containing ?
	T5: Fraction of posts containing !
	T6: Fraction of posts containing ?!
	T7: Fraction of posts containing @
	T8: Fraction of posts containing URL
	T9: Fraction of posts containing #
	T10: Fraction of posts with positive sentiment
	T11: Fraction of posts with negative sentiment

Table 2: Statistics on the datasets used in our experiments.

	Weibo	Twitter
Users	1,422,140	125,602
Posts	1,803,891	245,799
Rumor Events	2,313	498
Non-Rumors Events	2,351	493

Subevent level: After obtaining the subevent vector u_i, we again use the attention mechanism with social vector s in post level to select significant subevents to form a event representation. Let W_{hu} and W_{hu} denote the weight matrices, b_u denotes the bias term and ${v_u}^T$ is a transposed vector. The event vector e is computed as follows:

$$\mathcal{F}_u(h_i) = {v_u}^T \phi(W_{hu} h_i + W_{su} s + b_u) \tag{18}$$

$$\gamma_i = \frac{exp(\mathcal{F}_u(h_i))}{\sum_i exp(\mathcal{F}_u(h_i))} \tag{19}$$

$$e = \sum_i \gamma_i h_i \tag{20}$$

where γ_i is the weight of hidden state in subevent level which can be obtained similar to the weight in post level.

3.4 Rumor Classification

The event vector e, which is extracted via hierarchical attention based network, can be regard as high level representation of an given event. We project the vector e into the target space of two classes, which are rumor and non-rumor, via a fully connected layer:

$$\hat{e} = \phi(W_c e + b_c) \tag{21}$$

Afterwards, we use a softmax layer to compute the rumor distribution:

$$p = softmax(\hat{e}) \tag{22}$$

In the proposed model, the cross-entropy error between the probability distribution of the prediction and the ground truth is defined as loss function. Given the training samples, the training minimizes the loss:

$$L = -\sum_{e \in E} [y^e log(p_r^e) + (1 - y^e) log(1 - p_r^e)] \tag{23}$$

where y^e is the ground truth with 1 representing rumor and 0 representing non-rumor, p_r^e is the predicted probability of class rumor and E represents the training events.

3.5 Social Features

Social information has been frequently used in many literatures [4, 19, 30].These features capture important social characteristics, and incorporating social features could enhance the performance by providing extra social contexts of the rumor event. For example, given the sentiment related features in the social features, the model will pay more attention on the words or posts with negative emotion under the social attention mechanism if those features represents negative sentiment, or the model will pay more attention on the words or posts with positive emotion. We extract three types of social features including account-based features, propagation-based features and post-based features, as summarized in Table 1.

User Profile considers users' credibility, reliability, and reputation, which are extracted from user profile and behavior, such as the fraction of verified users in the rumor event, the average number of followers (or friends). As we all know, users' credibility is crucial in detecting rumors. For example, users without avatar are more likely to be zombie users, who play an important role in spreading rumors.

Propagation considers attributes related to the propagation pattern of an event, such as the average number of comments or reposts. In most cases, rumors are propagated in several fixed patterns.

Post Texts refers to statistic characteristics of posts in an event such as the average length of posts, the fraction of posts with positive sentiment. For example, T4 , T5 and T6 are related to the attitude of users involved in the event; rumors are prone to express negative sentiments to attract more users. These features can present sentiment-related attributes of posts.

4 EXPERIMENTAL RESULTS

In this section, we extensively evaluate our proposed model on two datasets collected from real-world microblogs.

4.1 Datasets

Twitter dataset [18]. All rumor events are crawled from Twitter by searching keywords extracted from fake news on Snopes. Part of non-rumor events are also from Snopes, and others are from two public datasets [4, 17].

Weibo dataset [18]. This dataset includes 2,313 rumors events and 2,351 non-rumors events. The rumors events are verified by Sina community management center, and the non-rumor events are gathered by crawling posts in general threads.

In our experiments, we split each dataset into training set (80%) and testing set (20%). Table 2 summarizes the statistics of these two datasets.

4.2 Experimental Setup

For word representation, we choose the distributed word vector instead of one-hot representation. We pre-train a 32-dimensional word embedding on each dataset with SkipGram [21] after standard word segmentation [20, 26]. We optimize the models by Adam [16] with default settings. We set the dimension of the hidden states of Bi-LSTM as 32 and use tangent activation function. We use 64 events for each training batch and iterate all training events in each epoch. Training stops upon 50 epochs.

We compare our model with the following state-of-the-arts models:

- **DTC** [4] uses Decision Tree Classifier to predict the credibility on twitter. We implement this method with features based on the statistics of posts.
- **SVM-TS** [19] utilizes a linear SVM to classify rumors on twitter and uses time-series structure to model the social feature variations.

Table 3: Performance comparison on two datasets: Weibo (top) and Twitter (bottom).

Datasets	Methods	Accuracy	Rumor			Non-rumor		
			Precision	Recall	F-1	Precision	Recall	F-1
Weibo	DTC	0.854	0.890	0.805	0.845	0.825	0.902	0.862
	SVM-TS	0.859	0.925	0.779	0.846	0.812	0.938	0.871
	ML-GRU	0.862	0.858	0.864	0.861	0.865	0.860	0.863
	CallAtRumor	0.887	0.918	0.847	0.881	0.860	0.926	0.892
	HSA-BLSTM	**0.943**	**0.946**	**0.940**	**0.943**	**0.941**	**0.947**	**0.944**
Twitter	DTC	0.711	0.744	0.646	0.692	0.685	0.776	0.727
	SVM-TS	0.751	0.805	0.667	0.729	0.713	0.837	0.770
	ML-GRU	0.784	0.870	0.670	0.757	0.730	0.899	0.805
	CallAtRumor	0.804	0.821	**0.780**	0.800	**0.789**	0.828	0.808
	HSA-BLSTM	**0.844**	**0.948**	0.730	**0.825**	0.779	**0.960**	**0.863**

- **ML-GRU** [18] uses a multilayer generic GRU network to model the microblog event as a variable-length time series, which is effective for early detection of rumors.
- **CallAtRumor** [6] presents an LSTM model to automatically identify rumors. By using the standard attention mechanism at word level, this method could detect rumors effectively.
- **HSA-BLSTM** denotes our proposed hierarchical model with social attention.

For evaluation metrics, we follow the convention and adopt *Accuracy*, *Precision*, *Recall* and *F1* scores for a comprehensive evaluation.

4.3 Performance Comparison

Table 3 shows the performance of all compared methods. Our model outperforms all other baselines on both datasets. Specifically, HSA-BLSTM achieves a accuracy of 94.3% on Weibo dataset and 84.4% on Twitter dataset, which indicates the flexibility of our model on different types of datasets. Moreover, the outstanding results indicate that the hierarchy based models can effectively learn the representation of each semantic level.

As expected, ML-GRU and CallAtRumor outperform methods using traditional classifiers on manually crafted features. This observation indicates that generic RNN model can learn deep latent features automatically and standard attention mechanism in word level can further improve the accuracy.

The proposed model perform strongly on detecting rumors. Specifically, on Weibo dataset, our model increases the accuracy of SVM based method from 85.9% to 94.3% and outperforms the second-best method by 6%. On Twitter dataset, the accuracy is boosted from 75.1% to 84.4%, and HSA-BLSTM outperforms the second-best method by 4.4%. This demonstrates the importance of incorporating social features into the model.

To better analyze the behavior of the attention mechanism, we rank detected rumors by the predicted scores and obtain the weight parameters. Figure 3 shows a visualization of weights at word-level and post-level. The color indicates the attention degree paid to each word in a post, and related posts are listed in descending order according to their weights. We observe that posts questioning the credibility of the event or seeking for verification are given more attention by our model. Interestingly, words highly related to the event are paid less attention than those words expressing questioning, negative sentiment. These observations demonstrate that our attention mechanism can discover the crucial components while ignore unrelated ones in detecting rumors.

切除，消融

4.4 Ablation Study

We also evaluate several internal models, which are simplified variations of HSA-BLSTM by removing some components, to further investigate the impact of attention mechanism and social features in the proposed model:

- **H-BLSTM:** We remove the attention mechanism and the social features in each semantic level.
- **HA-network:** The Bi-LSTM layer and the social features in each semantic level are removed.
- **HA-BLSTM:** Social features are removed.
- **HUA-BLSTM:** Only the user-profile features are used.
- **HDA-BLSTM:** Only the propagation features are used.
- **HPA-BLSTM:** Only the post-texts features are adopted.
- **HA-BLSM+S:** We simply concatenate the social features with the event representation in this model.

Network Components From Table 4, we have the following observations: (1) Standard attention mechanism applied in each semantic level improves the performance, compared with direct average pooling. (2) Bi-LSTM improves the performance, by encoding the input vector and learning discriminative representations. (3) Models with social features perform better, indicating the importance of social context. (4) Attention mechanism incorporating social features performs better than standard attention and direct social feature concatenation. Fused social features provide extra social contexts and extract important components via attention mechanism. Moreover, it also demonstrates that the characteristics of social contexts is reflected on multiple semantic levels.

Social Features To study the importance of each social feature, we further compare models that only adopt one type of social features. Based on our study, post features contribute the most among three types of social features. Features related to sentiments (e.g., “average sentiment score", “fraction of posts with positive/negative sentiment") work best, since rumor and non-rumor event always have different sentiment polarities. The combination of three types of features works best, which indicates that all features related to social context are helpful for rumor detection.

Table 4: Ablation study of our method.

Datasets	Methods	Accuracy	Rumor			Non-rumor		
			Precision	Recall	F-1	Precision	Recall	F-1
Weibo	H-BLSTM	0.895	0.912	0.873	0.892	0.880	0.917	0.898
	HA-Network	0.908	0.902	0.914	0.908	0.914	0.902	0.908
	HA-BLSTM	0.926	0.950	0.898	0.923	0.905	0.953	0.929
	HUA-BLSTM	0.928	0.952	0.901	0.926	0.907	0.955	0.930
	HDA-BLSTM	0.928	0.954	0.898	0.925	0.906	0.958	0.931
	HPA-BLSTM	0.936	0.968	0.901	0.933	0.910	0.970	0.939
	HS-BLSTM+S	0.934	0.963	0.901	0.931	0.908	0.966	0.936
Twitter	H-BLSTM	0.774	0.923	0.600	0.727	0.701	0.949	0.807
	HA-Network	0.819	0.856	0.770	0.811	0.789	0.869	0827
	HA-BLSTM	0.824	0.911	0.720	0.804	0.767	0.929	0.840
	HUA-BLSTM	0.824	0.911	0.720	0.804	0.767	0.929	0.840
	HDA-BLSTM	0.829	0.913	0.730	0.811	0.773	0.929	0.844
	HPA-BLSTM	0.834	0.914	0.740	0.818	0.780	0.929	0.848
	HA-BLSTM+S	0.834	0.924	0.730	0.816	0.775	0.939	0.849

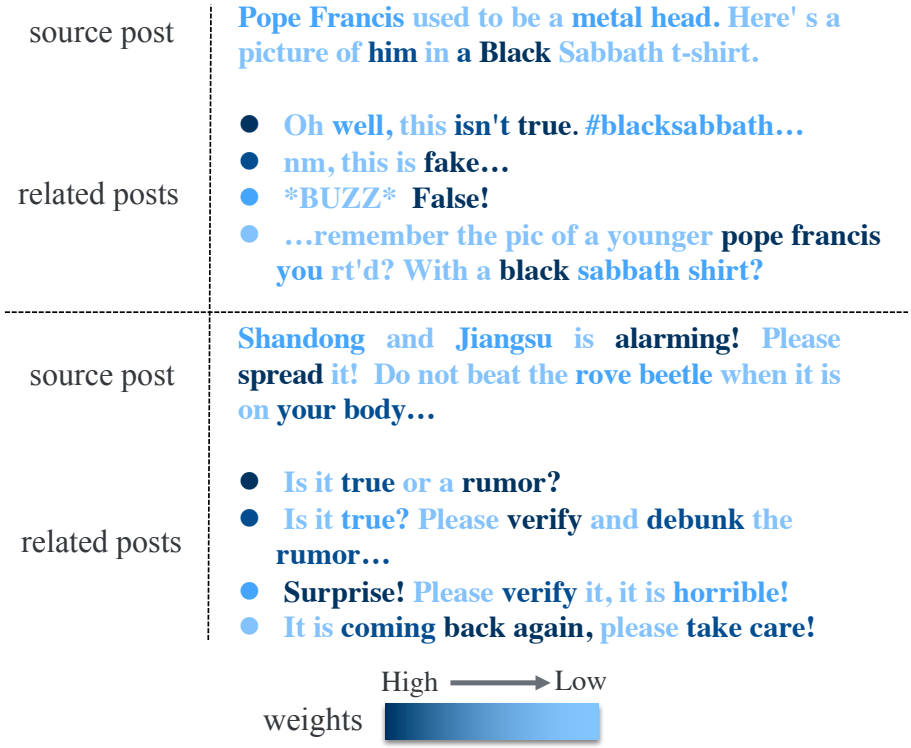

Figure 3: Visualization of attention weights on example rumors (Top: from Twitter, bottom: from Weibo). Word-level weights are shown in different colors, and posts are listed according to their weights in descending order. [This figure is best viewed in color.]

4.5 Early Detection

In rumor detection, one of the most crucial goals is to detect rumors as early as possible so that interventions can be made in time [32]. We set the maximum number for posts in an event as a deadline and assume that all related posts published after this deadline are invisible. For example, if the maximum number is 200, we only use the first 200 posts in the given event for classification.

By varying the maximum number of posts, the accuracy of several competitive models are shown in Figure 4 and Figure5. Note that the performance of our proposed model is superior to other methods when only a few microblogs are visible. With more data

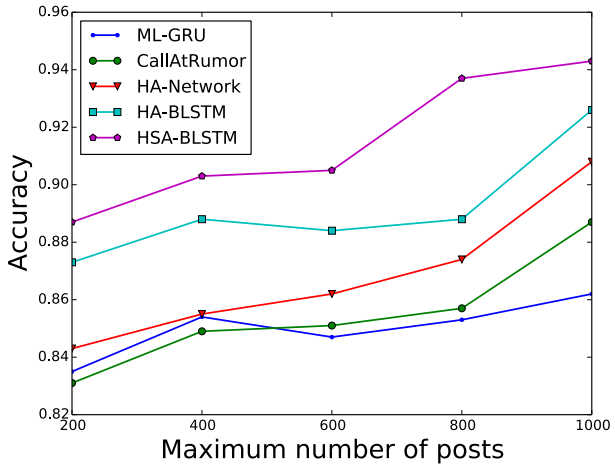

Figure 4: Early detection on Weibo dataset

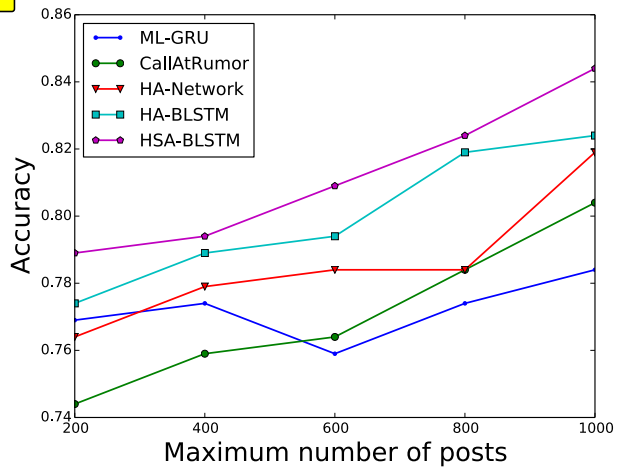

Figure 5: Early detection on Twitter dataset

involved in training, all methods perform better. In particular, unlike other generic RNN-based methods whose performances are unstable around 400 maximal posts, the ability of making early detection of our model increases not only rapidly but also steady as we access more data.

5 CONCLUSIONS

In this paper, we propose a hierarchical LSTM network with social attention for detecting rumors. Unlike most existing works extracting hand-crafted social features or feeding sequential posts to network directly, we model a given event as a hierarchical structure in three semantic levels. To obtain a more accurate representation, we fuse the social information via attention mechanism to capture important components in post and subevent level. Extensive experiments conducted on Weibo and Twitter datasets show that the proposed model (HSA-BLSTM) can significantly outperform other state-of-the-arts models. Moreover, during the early stage of rumor propagation, our proposed model can effectively and stably detect rumor events based on a small quantity of posts.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (U1703261), Beijing Municipal Science and Technology Project (Z181100008918006) and the National Natural Science Foundation of China (61571424).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.
[3] Juan Cao, Zhiwei Jin, Yazi Zhang, and Yongdong Zhang. 2016. MCG-ICT at MediaEval 2016 Verifying Tweets from both Text and Visual Content.. In *MediaEval*.
[4] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*. ACM, 675–684.
[5] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural Sentiment Classification with User and Product Attention.. In *EMNLP*. 1650–1659.
[6] Tong Chen, Lin Wu, Xue Li, Jun Zhang, Hongzhi Yin, and Yang Wang. 2017. Call Attention to Rumors: Deep Attention Based Recurrent Neural Networks for Early Rumor Detection. *arXiv preprint arXiv:1704.05973* (2017).
[7] Weiling Chen, Yan Zhang, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee. 2017. Unsupervised rumor detection based on usersâĂŹ behaviors using neural networks. *Pattern Recognition Letters* (2017).
[8] Nicholas DiFonzo and Prashant Bordia. 2007. *Rumor psychology: Social and organizational approaches*. American Psychological Association.
[9] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. 2002. Learning precise timing with LSTM recurrent networks. *Journal of machine learning research* 3, Aug (2002), 115–143.
[10] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 273–278.
[11] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 6645–6649.
[12] Linmei Hu, Juanzi Li, Liqiang Nie, Xiao-Li Li, and Chao Shao. 2017. What Happens Next? Future Subevent Prediction Using Contextual Hierarchical LSTM.. In *AAAI*. 3450–3456.
[13] Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, Yu Wang, and Jiebo Luo. 2017. Detection and Analysis of 2016 US Presidential Election Related Rumors on Twitter. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, 14–24.
[14] Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. 2016. News Verification by Exploiting Conflicting Social Viewpoints in Microblogs.. In *AAAI*. 2972–2978.
[15] Zhiwei Jin, Juan Cao, Yongdong Zhang, Jianshe Zhou, and Qi Tian. 2017. Novel visual and statistical image features for microblogs news verification. *IEEE Transactions on Multimedia* 19, 3 (2017), 598–608.
[16] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[17] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 1103–1108.
[18] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks.. In *IJCAI*. 3818–3824.
[19] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1751–1754.
[20] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010
[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[22] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*. 2204–2212.

[23] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*. 1310–1318.

[24] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1589–1599.

[25] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[26] Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016. *Thulac: An efficient lexical analyzer for chinese.* Technical Report. Technical Report.

[27] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[28] Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* (2015).

[29] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations.. In *AAAI*. 2835–2841.

[30] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. ACM, 13.

[31] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J Smola, and Eduard H Hovy. 2016. Hierarchical Attention Networks for Document Classification.. In *HLT-NAACL*. 1480–1489.

[32] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1395–1405.

[33] Xing Zhou, Juan Cao, Zhiwei Jin, Fei Xie, Yu Su, Dafeng Chu, Xuehui Cao, and Junqiang Zhang. 2015. Real-Time News Cer tification System on Sina Weibo. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 983–988.