Detect Rumors Using Time Series of Social Context Information on Microblogging Websites

Jing Ma[1,3] Wei Gao[2] Zhongyu Wei[4] Yueming Lu[1] Kam-Fai Wong[3,5]

[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar
[3]Dept. of SEEM, The Chinese University of Hong Kong, Hong Kong
[4]Computer Science Department, The University of Texas at Dallas, Texas 75080, USA
[5]MoE Key Laboratory of High Confidence Software Technologies, China
{majing,kfwong}@se.cuhk.edu.hk, wgao@qf.org.qa, zywei@hlt.utdallas.edu, ymlu@bupt.edu.cn

ABSTRACT

Automatically identifying rumors from online social media especially microblogging websites is an important research issue. Most of existing work for rumor detection focuses on modeling features related to microblog contents, users and propagation patterns, but ignore the importance of the variation of these social context features during the message propagation over time. In this study, we propose a novel approach to capture the temporal characteristics of these features based on the time series of rumor's lifecycle, for which time series modeling technique is applied to incorporate various social context information. Our experiments using the events in two microblog datasets confirm that the method outperforms state-of-the-art rumor detection approaches by large margins. Moreover, our model demonstrates strong performance on detecting rumors at early stage after their initial broadcast.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Rumor detection; temporal; time series; social context

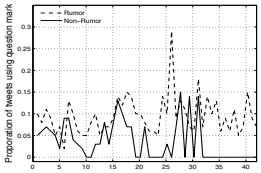
1. INTRODUCTION

A rumor is commonly defined as a statement whose truth value is unverifiable or deliberately false [3]. These rumors on microblogging websites, carrying unreal or even malicious information, can bring massive panic and social unrest to our community. For instance, on April 23, 2013, a rumor on Twitter about two explosions in the white house injuring Barack Obama caused the stock market crash in the US[1].

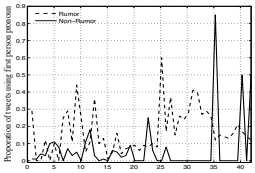
[1]http://www.bbc.com/news/world-us-canada-21508660

CIKM'15, October 19–23, 2015, Melbourne, Australia.

DOI: http://dx.doi.org/10.1145/2806416.2806607.



(a) Question mark



(b) The first-person pronoun

Figure 1: The two sample features changing over time (in hours) demonstrates different patterns in rumors and non-rumors

Therefore, automatic rumor detection technique that can quickly identify rumor messages and dynamically monitor the propagation of rumors become very useful.

Existing rumor detection methods typically exploit supervised machine learning models based on a wide range of features corresponding to users, contents of messages and their propagation patterns [2, 9, 7, 8, 10, 11]. An obvious limitation of these models is that they just consider the overall statistics on the social context information of messages as features, e.g., the total number of retweets, the time length of propagation, etc., and ignore the variation of these features over time.

To improve the accuracy of detection, we argue that it is of importance not only looking at the overall properties and the properties of individual messages, but also studying the changes or the trends of these properties along the lifecycle of the concerned hypothesis. For example, given two Twitter events, one is about "firing squad" (a rumor) and the other about "Hilton is arrested" (a non-rumor), Figure 1(a) and 1(b) show the variation of proportion of tweets using question marks and first person pronouns using time series, respectively, which are two of the typical features used in previous work. Figure 1(a) implies that the non-rumor tends to use less question marks than the rumor does at the later stage, but according to Figure 1(b), there might be more frequent use of the first person pronoun in the rumor at the early stage. Such variations reflect different characteristics of rumors and non-rumors over time during diffusion.

Kwon et al. [5, 4] recently introduced a time series fitting model that shows better detection result based on the temporal properties regarding a single feature – tweet volume. However, their temporal model focuses on converting

the continuous time series of tweet volume into only 3 fitting parameters for capturing the temporal fluctuations of features, which might result in significant information loss given complex time series. Also, it is difficult to extend the number of fitting parameters in their model for further improving the fitting effect.

To overcome these shortcomings, we propose a novel time series model called Dynamic Series-Time Structure (DSTS) to capture the variation of a wide spectrum of social context information over time far more than the tweet volume feature. We will study how well the time series of social context features can capture the variation of these features during the spread of event messages, which is supposed to benefit the differentiation between rumors and non-rumors. We utilize two datasets containing hundreds of events crawled from Twitter and Sina Weibo which are the most popular microblogging websites in English and Chinese, respectively. We build classifiers using the DSTS-based features and the annotated datasets. In our approach, we examine two basic settings: (1) given the complete lifecycle of an event about some specific topic, we decide it is a rumor or not; (2) given the event data at the early stage of propagation, we apply our model for early rumor detection. Experimental results under the two settings demonstrate that our DSTS-based model achieves promising improvements over the state-of-the-art approaches on both datasets.

2. TIME SERIES OF MICROBLOG EVENT

An event is considered as a set of microblogs related to some specific topic, e.g., "Hillary Clinton announces 2016 campaign for president", "MH370 landing in Nanning", etc.. The topics can be compiled manually from Twitter [2] or derived from Sina's community management center [10], which include verified rumors and non-rumor events (Section 4.1).

We model microblog data as a set of events $E = \{E_i\}$, and each event E_i consists of relevant microblogs $\{m_{ij}\}$. We represent each E_i as a D-dimensional vector $\mathbf{F}_i^D$ containing social context features regarding the contents, users and diffusion patterns of the relevant microblogs. To make the number of features tractable, we convert the continuous time stream of microblogs associated with each event into fixed time intervals. For learning our model, we extract a rich set of features sensitive to time, where not only the overall statistics of social context information but also the variation of individual features based on the time intervals can be captured.

In this section, we will first introduce an approach to discretize time stream for generating time stamps, then a method for capturing the variation of features.

2.1 Time Stamps Generation

For an event E_i, let $timeFirst_i$ and $timeLast_i$ be the time when the initial and the last microblog is posted, respectively. We convert the creation time of each microblog m_{ij} to a time interval falling into the range from 0 to N, serving as the time stamp of m_{ij}, where N is the tunable number of time intervals. We determine the length of time interval for E_i and the time stamp (TS) for m_{ij} created at time $t_{m_{ij}}$ as follows:

$$Interval(E_i) = \lceil \frac{timeLast_i - timeFirst_i}{N} \rceil \quad (1)$$

$$TS(m_{ij}) = \lfloor \frac{t_{m_{ij}} - timeFirst_i}{Interval(E_i)} \rfloor \quad (2)$$

where $Interval(.)$ is the length of each time interval in the number of time units like minutes, hours or days, and $TS(.)$ is the index of time stamp which m_{ij} falls into, taking the value of $0, 1, \cdots, N$. We use hour as time unit in this work.

2.2 Dynamic Series-Time Structure (DSTS)

With all the time stamps of E_i, a vector of its social context features $V(E_i)$ can be naturally generated given each time stamp. However, the temporal properties of such information is subject to continuous change over time, which cannot be captured effectively by just modeling features within individual time intervals. A better approach would be to identify the shapes of time series, which are formed by the relative change between the consecutive intervals, as a supplement of the absolute temporal properties.

For this purpose, we propose a Dynamic Series-Time Structure (DSTS), which is used to capture the variation of each feature. In this structure, we not only consider the absolute feature values from the initial time up to each interval, but also incorporate the slopes of features between two consecutive intervals. Therefore, the feature vector based on DSTS is represented as:

$$V(E_i) = (\mathbf{F}_{i,0}^D, \mathbf{F}_{i,1}^D, \ldots, \mathbf{F}_{i,N}^D; \mathbf{S}_{i,1}^D, \ldots \mathbf{S}_{i,N-1}^D) \quad (3)$$

$$\mathbf{F}_{i,t}^D = (\widetilde{f}_{i,t,1}, \widetilde{f}_{i,t,2}, \ldots, \widetilde{f}_{i,t,D}) \quad (4)$$

$$\mathbf{S}_{i,t}^D = \frac{\mathbf{F}_{i,t+1}^D - \mathbf{F}_{i,t}^D}{Interval(E_i)} \quad (5)$$

where $\mathbf{F}_{i,t}^D$ is the feature vector generated from social context features for the microblogs in E_i *from time 0 to the t-th interval*, and $\mathbf{S}_{i,t}^D$ is the slopes of features between the t-th and the $(t+1)$-th intervals.

We use Z-score to normalize feature values along the time series. The Z-score of a feature from 0 to the t-th interval $f_{t,k}$ is defined as[2]:

$$\widetilde{f}_{t,k} = \frac{f_{t,k} - \overline{f}_k}{\sigma(f_k)} \quad (6)$$

where $\overline{f}_k$ is the mean of the k-th feature and $\sigma(f_k)$ is the standard deviation of the k-th feature over all the time intervals, and $f_{t,k}$ is the k-th feature from time 0 to the t-th interval which is obtained by calculating the average or other statistics of the feature over the microblogs falling into that time span (Section 3).

3. FEATURE ENGINEERING

In this section, we will engineer each of the social context features corresponding to $f_{t,k}$ given in Equation 6. Note that $f_{t,k}$ is typically obtained by averaging the original social context feature f_k defined on individual microblogs, but chances are there that some features are defined directly on all microblogs from time 0 to the interval t. We present three types of features: content-based, user-based and propagation-based features, some of which are derived from prior work [2, 10, 11] and several others are newly proposed. Table 1 describes all these features. For clarity, we give more details on some of the following features.

[2]For the simplicity of presentation, we omit the notation of the event index i here

Table 1: Description of features $f_{t,k}$ on microblogs from time 0 to time interval t of an event

Content-based features
LDA-based topic distribution of microblogs with 18 topics [10]
Average length of microblogs [2]
of positive (negative) words in microblogs [2]
Average sentiment score of microblogs [2, 10]
% of microblogs with URL [2, 10, 11]
% of microblogs with smiling (frowning) emoticons [2]
% of positive (negative) microblogs [2]
% of microblogs with the first-person pronouns [2]
% of microblogs with hashtags [2, 11]
% of microblogs with @ mentions [2]
% of microblogs with question marks [2]
% of microblogs with exclamation marks [2]
% of microblogs with multiple question/exclamation marks [2]
User-based features
% of users that provide personal description [2, 10, 11]
% of users that provide personal picture in profile
% of verified users [2, 10, 11]
% of verified users of each type, e.g., celebrities [10, 11]
% of male (female) users [10, 11]
% of users located in large (small) cities
Average # of friends of users [2, 10, 11]
Average # of followers of users [2, 10, 11]
Average # of posts of users [2, 10, 11]
Average days users' accounts exist since registration [2, 10, 11]
Average reputation score of users (i.e., followers/followees ratio)
Diffusion-based features
Average # of retweets [2, 10, 11]
Average # of comments for Weibo posts [10, 11]
of microblogs [2]

LDA-based topic distribution: For all the microblogs in E, we use a Latent Dirichlet Allocation (LDA) model [1] to obtain a n-topic distribution for each post. For every post m_{ij} of every event E_i, we have $T(m_{ij}) = (p_{ij}^{(1)}, p_{ij}^{(2)}, \ldots, p_{ij}^{(n)})$ where $p_{ij}^{(z)}$ is the probability of m_{ij} belonging to topic z ($z = 1, \cdots, n$). Then the topic distributions of all microblogs in the concerned time span of an event are averaged to obtain the n LDA-based topic distribution features. We set n=18 following previous work [10].

Average sentiment score: Similar feature but not the same was used in [2, 10]. Given a sentiment lexicon and an emoticon lexicon, the average sentiment score of microblogs in a time span of event E_i is calculated as:

$$\frac{1}{|m_i|}\sum_{j=1}^{|m_i|}(|w_{pos}|_{ij} - |w_{neg}|_{ij} + |e_{pos}|_{ij} - |e_{neg}|_{ij})$$

where $|w_{pos}|_{ij}$ and $|w_{neg}|_{ij}$ is the number of positive and negative words, respectively, $|e_{pos}|_{ij}$ and $|e_{neg}|_{ij}$ is the number of smiling and frowning emoticons, respectively, in microblog m_{ij}, and $|m_i|$ is the number of microblogs in the concerned time span of event E_i. For tweets, we use MPQA[3] sentiment lexicon and a set of frequent emoticons collected by ourselves; for Weibo, we used the sentiment lexicon and emoticons described in [10].

4. EXPERIMENTAL EVALUATION

4.1 Datasets and Setup

For tweets, we used the public dataset released by Castillo et al. [2]. They extracted 288 events using Twitter Monitor [6] from tweet feeds in April-September 2010. We filtered out events with less than 10 tweets and left 207 popular events, in which 110 of them are labeled as rumors. We also collected an additional microblog dataset from Sina Weibo, where the verified rumors came from Sina's community management center [10] that accepts reports of various misinformation. We kept those rumor events with at least 100 posts. This left us 422 rumor events. We injected 500 normal (non-rumor) events, each with more than 100 posts. Sina Weibo API[4] provides interfaces to capture the information of original and retweeted posts. Table 2 shows the details for our datasets.

Table 2: Details of the datasets

Statistic	Twitter	Sina Weibo
Users	568,261	585,475
Tweets	1,207,767	473,698
Events	216	922
Rumors	101	500
Non-Rumors	115	422
AVG. time length	35.8 Hours	2,719 Hours

We resorted to *linear* SVM classifier for our model. We made comparison between our DSTS-based SVM model and several strong baselines: (1) DT: The Twitter credibility model using Decision Tree classifier proposed in [2] with the social context features in Table 1 *without considering time series*; (2) RF: The Random Forest classifier proposed in [5] using features consisting of the three parameters fitting tweet volume curve [5] plus all static features in Table 1; (3) RF-ext: Our extension of the RF model [5] with additional features by adding the fitting parameters of time series of all social context features in Table 1; (4) SVM-RBF: The SVM-based model with RBF kernel proposed in [11] using all features in Table 1 without considering time series; (5) SVM_c^{DSTS}, SVM_u^{DSTS}, SVM_d^{DSTS}: Our DSTS model using content-based, user-based and diffusion-based features, respectively; (6) SVM_{all}^{DSTS}: Our fully configured DSTS model. We did not compare with [10] which used specific propagation structure of Sina Weibo platform while our method is much more general.

We implemented DT, RF and RF-ext using Weka[5] and SVM models with LibSVM[6]. We conducted 10-fold cross-validation, and used accuracy, precision, recall and F-measure for evaluation. We fixed N=50 on a development set.

4.2 Experimental Results

Table 3(a) and 3(b) show the performance of different methods. Overall, our system SVM_{all}^{DSTS}, although a linear model, clearly outperforms the baselines that are all based on non-linear models. In terms of accuracy, it improves DT, RF, RF-ext and SVM-RBF by 9.5%, 3.3%, 4.4% and 9.1% respectively on Twitter data, and improves the same by 9.3%, 3.8%, 5.2% and 8.6% respectively on Weibo data. This is because DSTS model could reserve much variation of the rich social context information. DT only uses the static social context features over the entire lifecyle of posts, and RF uses as additional features the three parameters of a model that fits the time series of retweets volume, which may suffer from information loss. Surprisingly, RF-ext performs worse than RF, which implies that representing the variation of each feature with the three parameters cannot

[3] http://mpqa.cs.pitt.edu/lexicons/

[4] http://open.weibo.com/wiki/API

[5] http://www.cs.waikato.ac.nz/ml/weka/

[6] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

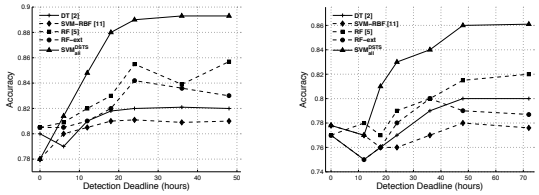
Table 3: Results of comparison with different methods (R: Rumor; N: Non-rumor)

(a) Twitter dataset

Method	Class	Accu.	Prec.	Recall	F_1
DT [2]	R	0.818	0.857	0.725	0.785
	N		0.793	0.897	0.842
RF [5]	R	0.867	0.896	0.805	0.848
	N		0.847	0.920	0.882
RF-ext	R	0.858	0.860	0.826	0.842
	N		0.856	0.886	0.871
SVM-RBF [11]	R	0.821	0.96	0.638	0.766
	N		0.760	0.977	0.855
SVM_c^{DSTS}	R	0.858	0.818	0.910	0.861
	N		0.905	0.810	0.855
SVM_u^{DSTS}	R	0.858	0.828	0.894	0.859
	N		0.892	0.825	0.857
SVM_d^{DSTS}	R	0.738	0.756	0.647	0.697
	N		0.725	0.816	0.768
SVM_{all}^{DSTS}	R	0.896	0.880	0.909	0.894
	N		0.912	0.883	0.897

(b) Sina Weibo dataset

Method	Class	Accu.	Prec.	Recall	F_1
DT [2]	R	0.774	0.771	0.830	0.800
	N		0.779	0.709	0.742
RF [5]	R	0.815	0.791	0.894	0.839
	N		0.852	0.720	0.780
RF-ext	R	0.804	0.775	0.898	0.832
	N		0.851	0.692	0.763
SVM-RBF [11]	R	0.779	0.771	0.842	0.805
	N		0.790	0.704	0.744
SVM_c^{DSTS}	R	0.817	0.833	0.828	0.830
	N		0.798	0.803	0.800
SVM_u^{DSTS}	R	0.811	0.815	0.844	0.829
	N		0.807	0.773	0.789
SVM_d^{DSTS}	R	0.760	0.755	0.826	0.789
	N		0.768	0.682	0.723
SVM_{all}^{DSTS}	R	0.846	0.861	0.854	0.857
	N		0.829	0.836	0.833



(a) Twitter (b) Sina Weibo

Figure 2: Results of rumor early detection

capture complex propagation patterns and fitting more features may accumulate even more information loss. Extending the three fitting parameters is limited by their original model. SVM-RBF is even worse than our DSTS models just using content-based and user-based subsets of features, indicating that our time series representation is very effective.

We find that using the subsets of features alone except diffusion-based features is already comparably good as the best baselines. Combining all features gives the best performance suggesting that they are complementary.

4.3 Rumor Early Detection

We examine the performance of our model on rumor early detection task that aims to identify rumors in the early stage of propagation. Given a detection deadline, we assume all messages of test events after the deadline are invisible when testing our model. When training the model, the complete lifecyle of training events is assumed observable.

Figure 2 shows the accuracy of our model SVM_{all}^{DSTS} in comparison with the baselines DT, RF, RF-ext and SVM-RBF at different deadlines. At the first few hours, our model does not have obvious advantage because it lacks of sufficient variation of social context. As time goes by, the performance of our model climbs much more rapidly after 5-10 hours while other models do not improve much because DSTS can capture rich variation patterns of features from the time series. Ours model achieves the similar accuracies of baselines at much earlier stage than they need. For example, on Twitter, it takes our model around 15 hours to get the highest accuracy of RF, the second best baseline, while RF needs more than 25 hours; on Weibo, our model needs about 20 hours but RF needs nearly 70 hours to achieve the similar performance. This suggests our model is very effective for early detection.

5. CONCLUSION AND FUTURE WORK

We proposed a novel approach to automatically identify rumors on microblogging websites. We develop a Dynamic Series-Time Structure model which explores the variation of various social context features over time. Experimental results show that our method with the time series of features achieves salient improvement on rumor detection given the complete lifecyle of events as well as at the early stage of diffusion. In future work, we plan to investigate unsupervised models using time series for identifying rumors online.

Acknowledgments

This work is partially supported by General Research Fund of Hong Kong (417112), Shenzhen Fundamental Research Program (JCYJ20130401172046450, JCYJ20120613152557576).

6. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on Twitter. In *Proceedings of WWW*, pages 675–684, 2011.

[3] N. DiFonzo and P. Bordia. *Rumor psychology: Social and organizational approaches.* American Psychological Association, 2007.

[4] S. Kwon and M. Cha. Modeling bursty temporal pattern of rumors. In *Proceedings of ICWSM*, 2014.

[5] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang. Prominent features of rumor propagation in online social media. In *Proceedings of ICDM*, pages 1103–1108, 2013.

[6] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the Twitter stream. In *Proceedings of SIGMOD*, pages 1155–1158, 2010.

[7] M. Mendoza, B. Poblete, and C. Castillo. Twitter under crisis: Can we trust what we RT? In *Proceedings of the first workshop on social media analytics*, pages 71–79. ACM, 2010.

[8] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of EMNLP*, pages 1589–1599, 2011.

[9] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: Mapping the spread of astroturf in microblog streams. In *Proceedings of WWW (demo)*, pages 249–252, 2011.

[10] K. Wu, S. Yang, and K. Q. Zhu. False rumors detection on Sina Weibo by propagation structures. In *Proceedings of ICDE*, 2015.

[11] F. Yang, Y. Liu, X. Yu, and M. Yang. Automatic detection of rumor on Sina Weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. ACM, 2012.