Real-time Rumor Debunking on Twitter

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, Sameena Shah*
Research and Development
Thomson Reuters
3 Times Square, NYC, NY 10036
{xiaomo.liu,armineh.nourbakhsh,quanzhi.li,rui.fang,sameena.shah}@thomsonreuters.com

ABSTRACT
In this paper, we propose the first real time rumor debunking algorithm for Twitter. We use cues from 'wisdom of the crowds', that is, the aggregate 'common sense' and investigative journalism of Twitter users. We concentrate on identification of a rumor as an event that may comprise of one or more conflicting microblogs. We continue monitoring the rumor event and generate real time updates dynamically based on any additional information received. We show using real streaming data that it is possible, using our approach, to debunk rumors accurately and efficiently, often much faster than manual verification by professionals.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords
Rumor Debunking, Twitter, Social Computing

1. INTRODUCTION
A rumor may be defined as a statement whose truth value is unverified or deliberately false [7]. Rumors spread fear, hate, or even euphoria. They may lead to defamation (of people, brands, governments etc.), protests, destruction of properties, or other undesirable responses. With the permeation of social media in our lives, the amount of rumors has increased remarkably and in significance. Twitter has a plethora of rumors that are spread intentionally or unintentionally. Perhaps, one of the most infamous cases is of the hacked AP account tweeting a rumor that Barack Obama had been injured in an explosion at the White House [5]. The tweet caused the S&P to decline and wipe off $130 Billion in stock value in a matter of seconds.

Traditionally, rumors were debunked by common sense and investigative journalism. Recently, a new line of research has emerged that uses cues from 'wisdom of the crowds', that is, the aggregation of 'common sense' and investigative journalism of Twitter users. In this paper, we consider the problem of rumor debunking using only cues from social media data, specifically Twitter. We propose a systematic methodology to mine language features like people's opinion, find witness accounts, derive underlying belief from messages, use sourcing, network propagation, credibility and other user and meta features to debunk rumors. We show that it is possible, using our approach, to debunk rumors on Twitter accurately and efficiently, often much faster than manual verification by professionals.

Our contributions include: (1) method to automatically debunk rumors on social media using only social media data; (2) an authentic rumor data base constructed on real data, and the process of its creation; (3) real time algorithmic veracity prediction that is potentially faster than human verification. To the best of our knowledge, this is the first such solution based on Twitter platform. Primarily, most research relevant to debunking rumors on social media, has concentrated on 'fitting' data retrospectively and on a static set. Our method allows for a dynamic real time prediction. We show that we are able to beat baseline methods and even professionals by a significant margin even when as less as five tweets about an event are available. We study how performance and top features change when more information is known about rumors.

2. RELATED WORK
Separating newsworthy stories from misinformation on Twitter has been a popular research topic in recent years. Early exploration of this problem started from two special case studies on rumor propagation during natural disasters like earthquakes and hurricanes[6, 3]. So their results are not necessarily generalizable to other types of rumors. Many existing algorithms [11, 10] for debunking rumors followed the work of Castillo et al. in which they studied information credibility and proposed a set of features that are able to retrospectively predict if an event is credible [1]. However, credibility prediction and rumor debunking are related problems but not equivalent. Even trustable news agencies and celebrities can get involved in propagating rumors. Thus, there is a need to invent new features that can capture how journalists verify rumors.

Another distinction is that current research of rumor debunking on Twitter and its equivalent in China, Sina Weibo,

*Corresponding author

CIKM'15, October 19–23, 2015, Melbourne, Australia.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806651

mainly focused on identifying rumor microblogs [7, 10, 11]—a tweet level metric. While we concentrate on detection of a rumor as an event comprised of multiple conflicting microblogs. Individual microblog is short in nature and contains quite limited information. Thus, predicting on a single microblog may not be very reliable. In contrast, grouping tweets of the same event is like collecting wisdom from the crowd. Intuitively, the latter approach should be more robust. Also, it is closer to a real life scenario that journalists need go through a set of tweets before verifying a rumor. The work of Sun [9] is similar to our work in this respect, where they studied rumor events in Sina Weibo. It is worth noting that Twitter's API does not let us track the propagation of retweets in as much detail as Sina Weibo. Thus, recent research on rumor detection using propagation networks in Sina Weibo [10] cannot be applied to Twitter. However, understandings how rumor propagates on social media [2] can help inspire more creative rumor debunking algorithms.

3. PROBLEM STATEMENT

We assume our system ingests real time streaming tweets. Our event detector (see section 4 for a more detailed description) organizes the stream into a set of events and identifies fast-evolving news stories dynamically. Our goal is not to compute a static veracity score retrospectively, rather to update the veracity score in real time for an event as it grows with new tweets. Thus, we are constrained to use only the initial meta data associated with incoming tweets. Certain meta data like number of retweets and likes, dynamically updated, are in fact very discriminatory after a significant lapse in time. Since they take time to build, and are all zero valued initially, we cannot use them, although they have been shown to be very useful in previous retrospective studies.

Based on the above requirements, we can articulate our prediction task as follows. We define a set of news events $E = \{e_1, ..., e_n\}$, where each event e_i is associated with a set of tweets $P_i = \{p_{i,1}, ..., p_{i,m_i}\}$. As time elapses, m_i, the number of associated tweets will grow. Thus, we can define a Twitter event at time t as a set of event tweets at that moment $P(t) = \{(p_1, t_1), ..., (p_m, t_m)\}$ with the order $t_1 <, ..., < t_m \leq t$. So, a real-time rumor predictor $M(t)$ is a function $\mathbb{R}^{m \times f} \rightarrow \mathbb{R}^f \rightarrow \{1, 0\}$ that aggregates a feature matrix for all event tweets $P(t)$ to a f-dimensional feature vector of this event and, then, maps it to a binary veracity class: truth (1) or rumor (0) at time t.

The time component allows us to study the early and subsequent predictions as events propagate. While retweets and other evolving features become meaningful at subsequent stages' time, and it is also possible to issue new API calls to re-obtain those values for each tweet, but considering Twitter rate limitations, it is not a feasible approach. Hence, we continue not to use those even for later stages. Next, we describe how we created a suitable data set to train on.

4. DATA COLLECTION

A critical challenge in creating appropriate training data is access to *confirmed rumors*. Rumors are a tiny portion of the total social media content. Moreover, we need access to rumors that are confirmed to be true or false by trustworthy sources. Fortunately, there exist two rumor tracking websites, namely, `snopes.com` and `emergent.info`. We crawled both sites and collected 2,299 stories posted until March 2015. We eliminated stories that were either not 'newsworthy' or that did not have an explicit confirmation of veracity. The resulting dataset consists of 94 true and 446 false stories.

Table 1: Example of collected rumors and truths

Date	Events	Veracity
01-15	Musician Macklemore joins the ISIS.	rumor
02-14	Nestle recalls hot-pocket products.	truth

Since the prediction is based on event level, we also need to collect all relevant tweets for each story. Following the approaches used in [1, 11], we formulated keyword-based queries that can describe each story. The form of queries is $A \wedge B$, where A is a conjunction of objects and subjects in an event and B is a disjunction of possible action words. For example, query (macklemore $\wedge$ ISIS) $\wedge$ (joins $\vee$ joined $\vee$ participates ...) refers to a rumor as described in Table 1. Instead of using Twitter's search API which limits its results to one week, we submitted each query directly to Twitter's search interface[1] to get full history and used a web scraper to download all matched tweets automatically. Three researchers sampled and cross-checked the results to ensure that the queries were capturing relevant tweets.

To collect additional true stories, we sent Twitter's free data stream[2], i.e. 1% of firehose, to our event detection system which essentially is a clustering algorithm that groups tweets of the same stories or events[3], very close to the outcomes of *Twitter Monitor* [4]. Like rumor data collection, we applied the same approach to go beyond the 1% limit and assemble all relevant tweets of each event. To further ensure those events had truly occurred, we further picked out events containing links to news that could confirm them. We kept running our system with Twitter's streaming data in March 2015 until the same number of true stories as rumors were collected. Thus, the final dataset for evaluation consisted of 421 true and 421 false events. Table 1 shows a few examples of rumors and true events from our dataset.

5. METHODOLOGY

5.1 Feature Hypotheses

Numerous features have been proposed for classifying rumors in social media[9, 10, 11]. They are largely rooted in the seminal paper on credibility [1], and include features based on language, user, propagation, and other meta data. However, investigative journalists consider features like originating source, source diversity, source credibility to be incredibly important [8].

Another observation is that wisdom of the crowd is very helpful for debunking rumors. People often express mixed beliefs such as support, denial and neutrality in response to events[6, 7]. Intuitively, we can conceptualize the crowd's contradictory beliefs as their debates on veracity. For an

[1]https://twitter.com/search-home
[2]https://dev.twitter.com/streaming/public
[3]The details of this system are to be reported in upcoming publications

Table 2: Description of new features

Category	Feature Name
Source Credibility	Is trusted/satirical news account
	Has trusted/satirical news url
	Profile has url from top domains
	Client application name
Source Identity	Profile has person name
	Profile has location
	Profile includes profession information
Source Diversity	Has multiple news/non-news urls after dedup
	Deduped tweets' text is dissimilar
Source Location & Witness	If tweet location matches event location
	If profile location matches event location
	Has witness phrases, i.e. "I see" and "I hear"
Msg. Belief	Is support, negation, question or neutrality
Event Propagation	Event Topic
	Retweet, mention, hashtag h-index
	Max reply/retweet graph[4] size/depth

event, if information from negation/question side is more authentic (represented by other features) than the support side, it is more likely to be confirmed as a rumor later.

Thus, we hypothesize: H_1 - investigative journalism based verification features can improve veracity prediction; H_2 - features involving belief partitions can improve the veracity prediction.

5.2 Classification Features

Verification Features We propose six categories of verification features based on insights from journalists [8], as shown in Table 2. They largely overlap with the features of the baseline algorithms [1, 11]. Due to space limitations, we only summarize new proposed features in each category.

Belief Identification classifies Twitter users into who believe or support that event versus who deny or question the event. We implemented two belief identification algorithms: one is a replication of [7]'s model, the other is a rule-based method using an extensive list of positive (e.g., truth), negative (i.e., rumor) and negation (e.g., not) keywords with a set of language rules such as "negative words not preceded by a negation word imply that the user denies the event". Both algorithms were evaluated on [7]'s dataset and a labeled dataset we got annotated by Amazon Mechanical Turk. The results showed that our rule-based algorithm performed better in the cross dataset evaluation (80.5% in accuracy). Hence, we used it to build the belief features.

Feature Aggregation. A key characteristic of our system is to detect event-level rumors rather than individual tweets. This requires us to aggregate features of a set of event tweets to a single event feature vector. We convert *Boolean* features, for example, "if tweet contains a news url" to "fraction of event tweets that have a news url". *Numeric* features are aggregated to minimum, maximum, and average of values. For *categorical* features, whose values are strings like "Monday", we break them down into an array of *Boolean* features at tweet level. Again, the aggregation for *Boolean* feature is applied to further convert them to "fraction" features. In addition, we partitioned event tweets into different belief groups. All existing features were recomputed for each group and treated as new belief features. They were added to our feature set to test the hypothesis H_2.

[4]Retweet graph was constructed by linking RT @mention in tweet text

Table 3: Performance comparison of prediction models

Method	Castillo	Yang	+H1	+H1+H2
5-tweet-accuracy	0.655	0.666	0.736	**0.780**
100-tweet-accuracy	0.729	0.737	0.812	**0.857**
400-tweet-accuracy	0.745	0.747	0.843	**0.897**
1-hour-accuracy	0.724	0.657	0.789	**0.795**
12-hour-accuracy	0.679	0.673	0.821	**0.834**
72-hour-accuracy	0.781	0.729	0.854	**0.875**

6. EVALUATION

We perform three evaluations: (1) performance comparison with the two baseline algorithms on the dynamic prediction task; (2) timeliness comparison of our automatic method in contrast to human verification; (3) analysis of the most effective features for early and subsequent predictions.

6.1 Comparison of Prediction Models

We compared our proposed method with two state-of-the-art algorithms [1, 11]. Although [1] is for credibility task, the idea here is to compare the predictive power of those features for the verification task. Hence, following the experiment design of [10], we used J48 decision tree and 15 best features reported in [1] to construct the first baseline; Yang's SVM classifier [11] was implemented with all their 19 features except 4 features. The metadata of "gender of user", "avatar type" and "user name type" are not available in Twitter; Twitter stream cannot capture "number of comments" as explained in section 3. We also constructed two classifiers based on our hypotheses. The first classifier followed H_1 to combine the new proposed verification features with all features in [1, 11]. The second classifier included additional belief partition features as designed by H_2. We applied a 'best first' feature selection strategy and experimented with a multitude of algorithms including SVM, random forest and decision tree to select the best classifier for each classification model. Since SVM performed best across the board, we selected it for the comparison.

Table 3 shows the accuracy of 10-fold cross validation for all four classifiers. The result shows that our new verification features can outperform both baseline models. Also, including belief partition can improve the performance even more. This is further evidenced by the t-test, where all comparisons' $p < 0.05$. Thus, both our hypotheses are supported. With regards to early prediction, our method can achieve 78% and 79% accuracy when events initiate with the first 5 tweets and the first 1 hour respectively.

6.2 Timeliness Comparison with Humans Verification

In this subsection, we investigate a more interesting question: can our method effectively debunk rumors faster than

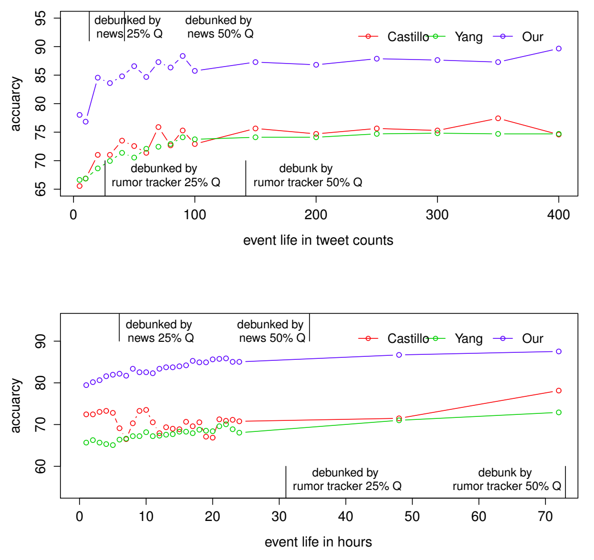

Figure 1: Comparison with human verification

human verification? To conduct this evaluation, we first compare the timeliness of our method with real-time rumor trackers, `snopes.com` and `emergent.info`, where debunking is performed by professionals. The delays between the first tweet in a rumor event and the debunking tweet published by rumor trackers are measured by tweet count and elapsed time. We then compute 25% and 50% quantiles of their delays and compare with our algorithm. Figure 1 shows that our algorithm can detect rumors with 85% accuracy when reaching 25% quantile lines (in the bottom of each subplot). In other words for other 75% of rumors, our method can already attain an 85% accuracy before rumors trackers can verify them. Next, we want to evaluate if our method can also verify rumors before news media. We adopt the same evaluation steps to measure the delays of rumor debunking by news agencies. In this case, we use the first event tweet linking to a debunking news article to measure the delay. However, there are some limitations here. We can only capture url links to news articles from tweets that appear in our data collection. So the earliest debunking time by news media is limited to the scope of our data. Also, we have to manually confirm the debunking time by reading each news article and making sure they are debunking the rumor. By the time of this paper's completion, 86 rumors were checked using this method. Thus, delays we measured for news media are an approximation. Based on the 25% quantile lines (in the top of each subplot) in Figure 1, we can again conclude that our method debunks 75% of rumors earlier than news media with at least 80% accuracy.

6.3 Feature Analysis

In total, 720 features were generated by our method. We examined the best features for early (5 tweets) vs. late (400 tweets) predictions. On comparing the top features in each case, we found that some features are more important for early detection while others begin dominating in later stages; some are consistently important. Actually, half of the consistently important features are based on user characteristics. We also observed that while belief features comprise 21% of best features in early prediction, they rise to 43% for later stage prediction. Thus, adding belief features to our model not only benefits the early prediction, but becomes increasingly important as more information becomes available.

7. CONCLUSIONS

Rumor debunking research has mostly concentrated on 'fitting' data retrospectively. In this paper, we propose the first real time rumor debunking algorithm for Twitter. We concentrate on identification of a rumor as an event that may comprise of one or more conflicting microblogs. We use beliefs of the crowd, along with traditional investigative journalism features on top of language, user, propagation and other meta features. We show that we are able to beat baseline methods and even professionals by a significant margin even when as little as five tweets about an event are available. Our system continues monitoring the rumor event and generates dynamic real time updates based on any additional information received.

8. REFERENCES

[1] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proc. International Conference on World Wide Web*, pages 675–684, 2011.

[2] A. Friggeri, L. A. Adamic, D. Eckles, and J. Cheng. Rumor cascades. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2014.

[3] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proc. International Conference on World Wide Web Companion*, pages 729–736, 2013.

[4] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1155–1158. ACM, 2010.

[5] C. Matthews. How does one fake tweet cause a stock market crash. *Wall Street & Markets: Time*, 2013.

[6] M. Mendoza, B. Poblete, and C. Castillo. Twitter under crisis: Can we trust what we rt? In *Proc. First Workshop on Social Media Analytics*, pages 71–79, 2010.

[7] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599. ACL, 2011.

[8] C. Silverman. *Verification handbook*. 2014.

[9] S. Sun, H. Liu, J. He, and X. Du. Detecting event rumors on sina weibo automatically. In *Web Technologies and Applications*, pages 120–131. Springer, 2013.

[10] K. Wu, S. Yang, and K. Q. Zhu. False rumors detection on sina weibo by propagation structures. In *IEEE International Conference of Data Engineering*, 2015.

[11] F. Yang, Y. Liu, X. Yu, and M. Yang. Automatic detection of rumor on sina weibo. In *Proc. of the ACM SIGKDD Workshop on Mining Data Semantics*, page 13, 2012.