Rumor has it: Identifying Misinformation in Microblogs

Vahed Qazvinian Emily Rosengren Dragomir R. Radev Qiaozhu Mei
University of Michigan
Ann Arbor, MI
{vahed,emirose,radev,qmei}@umich.edu

Abstract

A rumor is commonly defined as a statement whose true value is unverifiable. Rumors may spread misinformation (false information) or disinformation (deliberately false information) on a network of people. Identifying rumors is crucial in online social media where large amounts of information are easily spread across a large network by sources with unverified authority. In this paper, we address the problem of rumor detection in microblogs and explore the effectiveness of 3 categories of features: content-based, network-based, and microblog-specific memes for correctly identifying rumors. Moreover, we show how these features are also effective in identifying disinformers, users who endorse a rumor and further help it to spread. We perform our experiments on more than 10,000 manually annotated tweets collected from Twitter and show how our retrieval model achieves more than 0.95 in Mean Average Precision (MAP). Finally, we believe that our dataset is the first large-scale dataset on rumor detection. It can open new dimensions in analyzing online misinformation and other aspects of microblog conversations.

1 Introduction

A rumor is an unverified and instrumentally relevant statement of information spread among people (DiFonzo and Bordia, 2007). Social psychologists argue that rumors arise in contexts of ambiguity, when the meaning of a situation is not readily apparent, or potential threat, when people feel an acute need for security. For instance rumors about 'office renovation in a company' is an example of an ambiguous context, and the rumor that 'underarm deodorants cause breast cancer' is an example of a context in which one's well-being is at risk (DiFonzo et al., 1994).

The rapid growth of online social media has made it possible for rumors to spread more quickly. Online social media enable unreliable sources to spread large amounts of unverified information among people (Herman and Chomsky, 2002). Therefore, it is crucial to design systems that automatically detect misinformation and disinformation (the former often seen as simply false and the latter as deliberately false information).

Our definition of a rumor is established based on social psychology, where a rumor is defined as a statement whose truth-value is unverifiable or deliberately false. In-depth rumor analysis such as determining the intent and impact behind the spread of a rumor is a very challenging task and is not possible without first retrieving the complete set of social conversations (e.g., tweets) that are actually about the rumor. In our work, we take this first step to retrieve a complete set of tweets that discuss a specific rumor. In our approach, we address two basic problems. The first problem concerns retrieving online microblogs that are rumor-related. In the second problem, we try to identify tweets in which the rumor is endorsed (the posters show that they believe the rumor).

2 Related Work

We review related work on 3 main areas: Analyzing rumors, mining microblogs, and sentiment analysis and subjectivity detection.

2.1 Rumor Identification and Analysis

Though understanding rumors has been the subject of research in psychology for some time (Allport and Lepkin, 1945), (Allport and Postman, 1947), (DiFonzo and Bordia, 2007), research has

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1589–1599,
Edinburgh, Scotland, UK, July 27–31, 2011.

only recently begun to investigate how rumors are manifested and spread differently online. Microblogging services, like Twitter, allow small pieces of information to spread quickly to large audiences, allowing rumors to be created and spread in new ways (Ratkiewicz et al., 2010).

Related research has used different methods to study the spread of memes and false information on the web. Leskovec et al. use the evolution of quotes reproduced online to identify memes and track their spread overtime (Leskovec et al., 2009). Ratkiewicz et al. (Ratkiewicz et al., 2010) created the "Truthy" system, identifying misleading political memes on Twitter using tweet features, including hashtags, links, and mentions. Other projects focus on highlighting disputed claims on the Internet using pattern matching techniques (Ennals et al., 2010). Though our project builds on previous work, our work differs in its general focus on identifying rumors from a corpus of relevant phrases and our attempts to further discriminate between phrases that confirm, refute, question, and simply talk about rumors of interest.

Mendoza et al. explore Twitter data to analyze the behavior of Twitter users under the emergency situation of 2010 earthquake in Chile (Mendoza et al.,). They analyze the re-tweet network topology and find that the patterns of propagation in rumors differ from news because rumors tend to be questioned more than news by the Twitter community.

2.2 Sentiment Analysis

The automated detection of rumors is similar to traditional NLP sentiment analysis tasks. Previous work has used machine learning techniques to identify positive and negative movie reviews (Pang et al., 2002). Hassan et al. use a supervised Markov model, part of speech, and dependency patterns to identify attitudinal polarities in threads posted to Usenet discussion posts (Hassan et al., 2010). Others have designated sentiment scores for news stories and blog posts based on algorithmically generated lexicons of positive and negative words (Godbole et al., 2007). Pang and Lee provide a detailed overview of current techniques and practices in sentiment analysis and opinion mining (Pang and Lee, 2008; Pang and Lee, 2004).

Though rumor classification is closely related to opinion mining and sentiment analysis, it presents a different class of problem because we are concerned not just with the opinion of the person posting a tweet, but with whether the statements they post appear controversial. The automatic identification of rumors from a corpus is most closely related to the identification of memes done in (Leskovec et al., 2009), but presents new challenges since we seek to highlight a certain type of recurring phrases. Our work presents one of the first attempts at automatic rumor analysis.

2.3 Mining Twitter Data

With its nearly constant update of new posts and public API, Twitter can be a useful source for collecting data to be used in exploring a number of problems related to natural language processing and information diffusion (Bifet and Frank, 2010). Pak and Paroubek demonstrated experimentally that despite frequent occurrences of irregular speech patterns in tweets, Twitter can provide a useful corpus for sentiment analysis (Pak and Paroubek, 2010). The diversity of Twitter users make this corpus especially valuable. Ratkiewicz et al also use Twitter to detect and track misleading political memes (Ratkiewicz et al., 2010).

Along with many advantages, using Twitter as a corpus for sentiment analysis does present unusual challenges. Because posts are limited to 140 characters, tweets often contain information in an unusually compressed form and, as a result, grammar used may be unconventional. Instances of sarcasm and humor are also prevalent (Bifet and Frank, 2010). The procedures we used for the collection and analysis of tweets are similar to those described in previous work. However, our goal of developing computational methods to identify rumors being transmitted through tweets differentiates our project.

3 Problem Definition

Assume we have a set of tweets that are about the same topic that has some controversial aspects. Our objective in this work is two-fold: (1) Extract tweets that are about the controversial aspects of the story and spread misinformation (**Rumor retrieval**). (2) Identify users who believe that misinformation versus users who refute or question the rumor (**Belief**

Name	Rumor	Regular Expression Query	Status	#tweets
obama	Is Barack Obama muslim?	`Obama & (muslim|islam)`	false	4975
airfrance	Air France mid-air crash photos?	`(air.france|air france) & (photo|pic|pix)`	false	505
cellphone	Cell phone numbers going public?	`(cell|cellphone|cell phone)`	mostly false	215
michelle	Michelle Obama hired too many staff?	`staff & (michelle obama|first lady|1st lady)`	partly true	299
palin	Sarah Palin getting divorced?	`palin & divorce`	false	4423

Table 1: List of rumor examples and their corresponding queries used to collect data from Twitter

classification).

The following two tweets are two instances of the tweets written about president Obama and the Muslim world. The first tweet below is about president Obama and Muslim world, where the second tweet spread misinformation that president Obama is Muslim.

(non-rumor) "As Obama bows to Muslim leaders Americans are less safe not only at home but also overseas. Note: The terror alert in Europe... "

(rumor) "RT @johnnyA99 Ann Coulter Tells Larry King Why People Think Obama Is A Muslim http://bit.ly/9rs6pa #Hussein via @NewsBusters #tcot .."

The goal of the retrieval task is to discriminate between such tweets. In the second task, we use the tweets that are flagged as rumorous, and identify users that endorse (believe) the rumor versus users who deny or question it. The following three tweets are about the same story. The first user is a believer and the second and third are not.

(confirm) "RT @moronwatch: Obama's a Muslim. Or if he's not, he sure looks like one #whyimvotingrepublican."

(deny) "Barack Obama is a Christian man who had a Christian wedding with 2 kids baptised in Jesus name. Tea Party clowns call that muslim #p2 #gop"

(doubtful) "President Barack Obama's Religion: Christian, Muslim, or Agnostic? - The News of Today (Google): Share With Friend... http://bit.ly/bk42ZQ"

The first task is substantially more challenging than a standard IR task because of the requirement of both high precision (every result should be actually discussing the rumor) and high recall (the set should be complete). To do this, we submit a handcrafted regexp (extracted from about.com) to Twitter and retrieve a large primitive set of tweets that is supposed to have a high recall. This set however, contains a lot of false positives, tweets that match the regexp but are not about the rumor (e.g., "Obama meets muslim leaders"). Moreover, a rumor is usually stated using various instances (e.g., "Barack HUSSEIN Obama" versus "Obama is muslim"). Our goal is then to design a learning framework that filters all such false positives and retrieves various instances of the same rumor

Although our second task, belief classification, can be viewed as an opinion mining task, it is substantially different from opinion mining in nature. The difference from a standard opinion mining task is that here we are looking for attitudes about a subtle statement (e.g., "Palin is getting divorce") instead of the overall sentiment of the text or the opinion towards an explicit object or person (e.g., "Sarah Palin").

4 Data

As September 2010, Twitter reports that its users publish nearly 95 million tweets per day[1]. This makes Twitter an excellent case to analyze misinformation in social media.

Our goal in this work was to collect and annotate a large dataset that includes all the tweets that are written about a rumor in a certain period of time. To collect such a complete and self-contained dataset about a rumor, we used the Twitter search API, and retrieved all the tweets that matched a given regular expression. This API is the only API that returns results from the entire public Twitter stream and not a small randomly selected sample. To overcome the rate limit enforced by Twitter, we collected matching tweets once per hour, and remove any duplicates.

To use the search API, we carefully designed regular expression queries to be broad enough to match

[1] http://twitter.com/about

all the tweets that are about a rumor. Each query represents a popular rumor that is listed as "false" or only "partly true" on About.com's Urban Legends reference site[2] between 2009 and 2010. Table 1 lists the rumor examples that we used to collect our dataset along with their corresponding regular expression queries and the number of tweets collected.

4.1 Annotation

We asked two annotators to go over all the tweets in the dataset and mark each tweet with a "1" if it is about any of the rumors from Table 1, and with a "0" otherwise. This annotation scheme will be used in our first task to detect false positives, tweets that match the broad regular expressions and are retrieved, but are not about the rumor. For instance, both of the following tweets match the regular expression for the `palin` example, but only the second one is rumorous.

(0) "McCain Divorces Palin over her 'untruths and out right lies' in the book written for her. McCain's team says Palin is a petty liar and phony"

(1) "Sarah and Todd Palin to divorce, according to local Alaska paper. http://ow.ly/iNxF"

We also asked the annotators to mark each previously annotated rumorous tweet with "11" if the tweet poster endorses the rumor and with "12" if the user refutes the rumor, questions its credibility, or is neutral.

(12) "Sarah Palin Divorce Rumor Debunked on Facebook http://ff.im/62Evd"

(11) "Todd and Sarah Palin to divorce http://bit.ly/15StNc"

Our annotation of more than 10,400 tweets shows that $\%35$ of all the instances that matched the regular expressions are false positives, tweets that are not rumor-related but match the initial queries. Moreover, among tweets that are about particular rumors, nearly $\%43$ show the poster believe the rumor, demonstrating the importance of identifying misinformation and those who are misinformed. Table 2 shows the basic statistics extracted from the annotations for each story.

[2]http://urbanlegends.about.com

Rumor	non-rumor (0)	believe (11)	deny/ (12) doubtful/neutral	total
obama	3,036	926	1,013	4975
airfrance	306	71	128	505
cellphone	132	74	9	215
michelle	83	191	25	299
palin	86	1,709	2,628	4,423
total	3,643	2,971	3,803	10,417

Table 2: Number of instances in each class from the annotated data

task	κ
rumor retrieval	0.954
belief classification	0.853

Table 3: Inter-judge agreement in two annotation tasks in terms of κ-statistic

4.2 Inter-Judge Agreement

To calculate the annotation accuracy, we annotated 500 instances twice. These annotations were compared with each other, and the Kappa coefficient (κ) was calculated. The κ statistic is formulated as

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

where $Pr(a)$ is the relative observed agreement among raters, and $Pr(e)$ is the probability that annotators agree by chance if each annotator is randomly assigning categories (Krippendorff, 1980; Carletta, 1996). Table 3 shows that annotators can reach a high agreement in both extracting rumors ($\kappa = 0.95$) and identifying believers ($\kappa = 0.85$).

5 Approach

In this section, we describe a general framework, which given a tweet, predicts (1) whether it is a rumor-related statement, and if so (2) whether the user believes the rumor or not. We describe 3 sets of features, and explain why these are intuitive to use for identification of rumors.

We process the tweets as they appear in the user timeline, and do not perform any pre-processing. Specially, we think that capitalization might be an important property. So, we do not lower-case the tweet texts either.

Our approach is based on building different Bayes classifiers as high level features and then learning a linear function of these classifiers for retrieval in the first task and classification in the second. Each

Bayes classifier, which corresponds to a feature f_i, calculates the likelihood ratio for a given tweet t, as shown in Equation 1.

$$\frac{P(\theta_i^+|t)}{P(\theta_i^-|t)} = \frac{P(\theta_i^+)}{P(\theta_i^-)} \frac{P(t|\theta_i^+)}{P(t|\theta_i^-)} \quad (1)$$

Here θ_i^+ and θ_i^- are two probabilistic models built based on feature f_i using a set of positive $(+)$ and negative $(-)$ training data. The likelihood ratio expresses how many times more likely the tweet t is under the positive model than the negative model with respect to f_i.

For computational reasons and to avoid dealing with very small numbers we use the log of the likelihood ratio to build each classifier.

$$LL_i = \log \frac{P(\theta_i^+|t)}{P(\theta_i^-|t)} = \log \frac{P(\theta_i^+)}{P(\theta_i^-)} + \log \frac{P(t|\theta_i^+)}{P(t|\theta_i^-)} \quad (2)$$

The first term $\frac{P(\theta_i^+)}{P(\theta_i^-)}$ can be easily calculated using the maximum likelihood estimates of the probabilities (i.e., the estimate of each probability is the corresponding relative frequency). The second term is calculated using various features that we explain below.

5.1 Content-based Features

The first set of features are extracted from the text of the tweets. We propose 4 content based features. We follow (Hassan et al., 2010) and present the tweet with 2 different patterns:

- Lexical patterns: All the words and segments in the tweet are represented as they appear and are tokenized using the space character.

- Part-of-speech patterns: All words are replaced with their part-of-speech tags. To find the part-of-speech of a hashtag we treat it as a word (since they could have semantic roles in the sentence), by omitting the tag sign, and then precede the tag with the label `TAG/`. We also introduce a new tag, `URL`, for URLs that appear in a tweet.

From each tweet we extract 4 (2×2) features, corresponding to unigrams and bigrams of each representation. Each feature is the log-likelihood ratio calculated using Equation 2. More formally, we represent each tweet t, of length n, lexically as $(w_1 w_2 \cdots w_n)$ and with part-of-speech tags as $(p_1 p_2 \cdots p_n)$. After building the positive and negative models (θ^+, θ^-) for each feature using the training data, we calculate the likelihood ratio as defined in Equation 2 where

$$\frac{P(t|\theta^+)}{P(t|\theta^-)} = \sum_{j=1}^{n} \log \frac{P(w_j|\theta^+)}{P(w_j|\theta^-)} \quad (3)$$

for unigram-lexical features (**`TXT1`**) and

$$\frac{P(t|\theta^+)}{P(t|\theta^-)} = \sum_{j=1}^{n-1} \log \frac{P(w_j w_{j+1}|\theta^+)}{P(w_j w_{j+1}|\theta^-)} \quad (4)$$

for bigram-based lexical features (**`TXT2`**). Similarly, we define the unigram and bigram-based part-of-speech features (**`POS1`** and **`POS2`**) as the log-likelihood ratio with respect to the positive and negative part-of-speech models.

5.2 Network-based Features

The features that we have proposed so far are all based on the content of individual tweets. In the second set of features we focus on user behavior on Twitter. We observe 4 types of network-based properties, and build 2 features that capture them.

Twitter enables users to re-tweet messages from other people. This interaction is usually easy to detect because the re-tweeted messages generally start with the specific pattern: '`RT @user`'. We use this property to infer about the re-tweeted message.

Let's suppose a user u_i re-tweets a message t from the user u_j (u_i: "`RT` @u_j t"). Intuitively, t is more likely to be a rumor if (1) u_j has a history of posting or re-tweeting rumors, or (2) u_i has posted or re-tweeted rumors in the past.

Given a set of training instances, we build a positive (θ^+) and a negative (θ^-) user models. The first model is a probability distribution over all users that have posted a positive instance or have been re-tweeted in a positive instance. Similarly, the second model is a probability distribution over users

that have posted (or been re-tweeted in) a negative instance. After building the models, for a given tweet we calculate two log-likelihood ratios as two network-based features.

The first feature is the log-likelihood ratio that u_i is under a positive user model (**USR1**) and the second feature is the log-likelihood ratio that the tweet is re-tweeted from a user (u_j) who is under a positive user model than a negative user model (**USR2**).

The distinction between the posting user and the re-tweeted user is important, since some times the users modify the re-tweeted message in a way that changes its meaning and intent. In the following example, the original user is quoting president Obama. The second user is re-tweeting the first user, but has added more content to the tweet and made it sound rumorous.

original message (non-rumor) "Obama says he's doing 'Christ's work'."

re-tweeted (rumor) "Obama says he's doing 'Christ's work.' Oh my God, CHRIST IS A MUSLIM."

5.3 Twitter Specific Memes

Our final set of features are extracted from memes that are specific to Twitter: hashtags and URLs. Previous work has shown the usefulness of these memes (Ratkiewicz et al., 2010).

5.3.1 Hashtags

One emergent phenomenon in the Twitter ecosystem is the use of hashtags: words or phrases prefixed with a hash symbol (#). These hashtags are created by users, and are widely used for a few days, then disappear when the topic is outdated (Huang et al., 2010).

In our approach, we investigate whether hashtags used in rumor-related tweets are different from other tweets. Moreover, we examine whether people who believe and spread rumors use hashtags that are different from those seen in tweets that deny or question a rumor.

Given a set of training tweets of positive and negative examples, we build two statistical models (θ^+, θ^-), each showing the usage probability distribution of various hashtags. For a given tweet, t, with a set of m hashtags ($\#h_1 \cdots \#h_m$), we calculate the log-likelihood ratio using Equation 2 where

	Feature	LL-ratio	model
Content	**TXT1**	content unigram	content unigram
	TXT2	content bigram	content unigram
	POS1	content pos	content pos unigram
	POS2	content pos	content pos bigram
Twitter	**URL1**	content unigram	target URL unigram
	URL2	content bigram	target URL bigram
	TAG	hashtag	hashtag
Network	**USR1**	tweeting user	all users in the data
	USR2	re-tweeted user	all users in the data

Table 4: List of features used in our optimization framework. Each feature is a log-likelihood ratio calculated against a a positive ($+$) and negative ($-$) training models.

$$\frac{P(t|\theta^+)}{P(t|\theta^-)} = \sum_{j=1}^{m} \log \frac{P(\#h_j|\theta^+)}{P(\#h_j|\theta^-)} \quad (5)$$

5.3.2 URLs

Previous work has discussed the role of URLs in information diffusion on Twitter (Honeycutt and Herring, 2009). Twitter users share URLs in their tweets to refer to external sources or overcome the length limit forced by Twitter. Intuitively, if a tweet is a positive instance, then it is likely to be similar to the content of URLs shared by other positive tweets. Using the same reasoning, if a tweet is a negative instance, then it should be more similar to the web pages shared by other negative instances.

Given a set of training tweets, we fetch all the URLs in these tweets and build θ^+ and θ^- once for unigrams and once for bigrams. These models are merely built on the content of the URLs and ignore the tweet content. Similar to previous features, we calculate the log-likelihood ratio of the content of each tweet with respect to θ^+ and θ^- for unigrams (**URL1**) and bigrams **URL2**).

Table 4 summarizes the set of features used in our proposed framework, where each feature is a log-likelihood ratio calculated against a positive ($+$) and negative ($-$) training models. To build these language models, we use the CMU Language Modeling toolkit (Clarkson and Rosenfeld, 1997).

5.4 Optimization

We build an L_1-regularized log-linear model (Andrew and Gao, 2007) on various features discussed before to predict each tweet. Suppose, a procedure generates a set of candidates for an input x. Also,

let's suppose $\Phi : X \times Y \rightarrow \mathbb{R}^D$ is a function that maps each (x, y) to a vector of feature values. Here, the feature vector is the vector of coefficients corresponding to different network, content, and twitter-based properties, and the parameter vector $\theta \in \mathbb{R}^D$ ($D \leq 9$ in our experiments) assigns a real-valued weight to each feature. This estimator chooses θ to minimize the sum of least squares and a regularization term R.

$$\hat{\theta} = arg\min_{\theta}\{\frac{1}{2}\sum_{i}||\langle\theta, x_i\rangle - y_i||_2^2 + R(\theta)\} \quad (6)$$

where the regularizer term $R(\theta)$ is the weighted L_1 norm of the parameters.

$$R(\theta) = \alpha\sum_{j}|\theta_j| \quad (7)$$

Here, α is a parameter that controls the amount of regularization (set to 0.1 in our experiments).

Gao et. al (Gao et al., 2007) argue that optimizing L_1-regularized objective function is challenging since its gradient is discontinuous whenever some parameters equal zero. In this work, we use the *orthant-wise limited-memory quasi-Newton* algorithm (OWL-QN), which is a modification of L-BFGS that allows it to effectively handle the discontinuity of the gradient (Andrew and Gao, 2007). OWL-QN is based on the fact that when restricted to a single orthant, the L_1 regularizer is differentiable, and is in fact a linear function of θ. Thus, as long as each coordinate of any two consecutive search points does not pass through zero $R(\theta)$ does not contribute at all to the curvature of the function on the segment joining them. Therefore, we can use L-BFGS to approximate the Hessian of $L(\theta)$ alone and use it to build an approximation to the full regularized objective that is valid on a given orthant. This algorithm works quite well in practice, and typically reaches convergence in even fewer iterations than standard L-BFGS (Gao et al., 2007).

6 Experiments

We design 2 sets of experiments to evaluate our approach. In the first experiment we assess the effectiveness of the proposed method when employed in an Information Retrieval (IR) framework for rumor retrieval and in the second experiment we employ various features to detect users' beliefs in rumors.

6.1 Rumor Retrieval

In this experiment, we view different stories as queries, and build a relevance set for each query. Each relevance set is an annotation of the entire 10,417 tweets, where each tweet is marked as *relevant* if it matches the regular expression query and is marked as a rumor-related tweet by the annotators. For instance, according to Table 2 the `cellphone` dataset has only 83 relevant documents out of the entire 10,417 documents.

For each query we use 5-fold cross-validation, and predict the relevance of tweets as a function of their features. We use these predictions and rank all the tweets with respect to the query. To evaluate the performance of our ranking model for a single query (Q) with the set of relevant documents $\{d_1, \cdots, d_m\}$, we calculate Average Precision as

$$AP(Q) = \frac{1}{m}\sum_{k=1}^{m}\text{Precision}(R_k) \quad (8)$$

where R_k is the set of ranked retrieval results from the top result to the k^{th} relevant document, d_k (Manning et al., 2008).

6.1.1 Baselines

We compare our proposed ranking model with a number of other retrieval models. The first two simple baselines that indicate a difficulty lower-bound for the problem are **Random** and **Uniform** methods. In the Random baseline, documents are ranked based on a random number assignment to them. In the Uniform model, we use a 5-fold cross validation, and in each fold the label of the test documents is determined by the majority vote from the training set.

The main baseline that we use in this work, is the regular expression that was submitted to Twitter to collect data (**regexp**). Using the same regular expression to mark the relevance of the documents will cause a recall value of 1.00 (since it will retrieve all the relevant documents), but will also retrieve false positives, tweets that match the regular expression but are not rumor-related. We would like to investigate whether using training data will help us decrease the rate of false positives in retrieval.

Finally, using the Lemur Toolkit software[3], we employ a KL divergence retrieval model with

[3]http://www.lemurproject.org/

Dirichlet smoothing (**KL**). In this model, documents are ranked according to the negation of the divergence of query and document language models. More formally, given the query language model θ_Q, and the document language model θ_D, the documents are ranked by $-D(\theta_Q||\theta_D)$, where D is the KL-divergence between the two models.

$$D(\theta_Q||\theta_D) = \sum_w p(w|\theta_Q) \log \frac{p(w|\theta_Q)}{p(w|\theta_D)} \quad (9)$$

To estimate $p(w|\theta_D)$, we use Bayesian smoothing with Dirichlet priors (Berger, 1985).

$$p_s(w|\theta_D) = \frac{C(w,D) + \mu.p(w|\theta_S)}{\mu + \sum_w C(w,D)} \quad (10)$$

where, μ is a parameter, C is the count function, and $theta_S$ is the collection language model. Higher values of μ put more emphasis on the collection model. Here, we try two variants of the model, one using the default parameter value in Lemur ($\mu = 2000$), and one in which μ is tuned based on the the data ($\mu = 10$). Using the test data to tune the parameter value, μ, will help us find an upper-bound estimate of the effectiveness of this method.

Table 5 shows the Mean Average Precision (MAP) and $F_{\beta=1}$ for each method in the rumor retrieval task. This table shows that a method that employs training data to re-rank documents with respect to rumors makes significant improvements over the baselines and outperforms other strong retrieval systems.

6.1.2 Feature Analysis

To investigate the effectiveness of using individual features in retrieving rumors, we perform 5-fold cross validations for each query, using different feature sets each time. Figure 1 shows the average precision and recall for our proposed optimization system when content-based (**TXT1+TXT2+POS1+POS2**), network-based (**USR1+USR2**), and twitter specific memes (**TAG+URL1+URL2**) are employed individually.

Figure 1 shows that features that are calculated using the content language models are very effective in achieving high precision and recall. Twitter specific features, especially hashtags, can result in high precisions but lead to a low recall value because many tweets do not share hashtags or are not written based on the contents of external URLs.

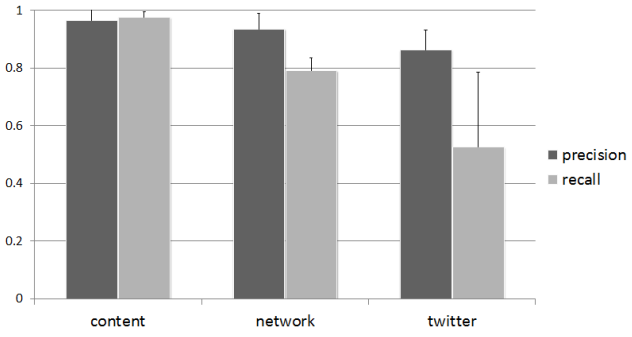

Figure 1: Average precision and recall of the proposed method employing each set of features: content-based, network-based, and twitter specific.

Finally, we find that user history can be a good indicator of rumors. However, we believe that this feature could be more helpful with a complete user set and a more comprehensive history of their activities.

6.1.3 Domain Training Data

As our last experiment with rumor retrieval we investigate how much new labeled data from an emergent rumor is required to effectively retrieve instances of that particular rumor. This experiment helps us understand how our proposed framework could be generalized to other stories.

To do this experiment, we use the `obama` story, which is a large dataset with a significant number of false positive instances. We extract 400 randomly selected tweets from this dataset and keep them for testing. We also build an initial training dataset of the other 4 rumors, and label them as *not relevant*. We assess the performance of the retrieval model as we gradually add the rest of the `obama` tweets. Figure 2 shows both Average Precision and labeling accuracy versus the size of the labeled data used from the `obama` dataset. This plot shows that both measures exhibit a fast growth and reach 80% when the number of labeled data reaches 2000.

6.2 Belief Classification

In previous experiments we showed that maximizing a linear function of log-likelihood ratios is an effective method in retrieving rumors. Here, we in-

Method	MAP	95% C.I.	$F_{\beta=1}$	95% C.I.
Random	0.129	[-0.065, 0.323]	0.164	[-0.051, 0.379]
Uniform	0.129	[-0.066, 0.324]	0.198	[-0.080, 0.476]
regexp	0.587	[0.305, 0.869]	0.702	[0.479, 0.925]
KL ($\mu = 2000$)	0.678	[0.458, 0.898]	0.538	[0.248, 0.828]
KL ($\mu = 10$)	0.803	[0.641, 0.965]	0.681	[0.614, 0.748]
LL (all 9 features)	**0.965**	[0.936, 0.994]	**0.897**	[0.828, 0.966]

Table 5: Mean Average Precision (MAP) and $F_{\beta=1}$ of each method in the rumor retrieval task. (C.I.: Confidence Interval)

Method	Accuracy	Precision	Recall	$F_{\beta=1}$	Win/Loss Ratio
random	0.501	0.441	0.513	0.474	1.004
uniform	0.439	0.439	1.000	0.610	0.781
TXT	0.934	0.925	0.924	0.924	14.087
POS	0.742	0.706	0.706	0.706	2.873
content (**TXT+POS**)	**0.941**	0.934	0.930	**0.932**	15.892
network (**USR**)	0.848	0.873	0.765	0.815	5.583
TAG	0.589	0.734	0.099	0.175	1.434
URL	0.664	0.630	0.570	0.598	1.978
twitter (**TAG+URL**)	0.683	0.658	0.579	0.616	2.155
all	0.935	**0.944**	0.906	0.925	14.395

Table 6: Accuracy, precision, recall, $F_{\beta=1}$, and win/loss ratio of belief classification using different features.

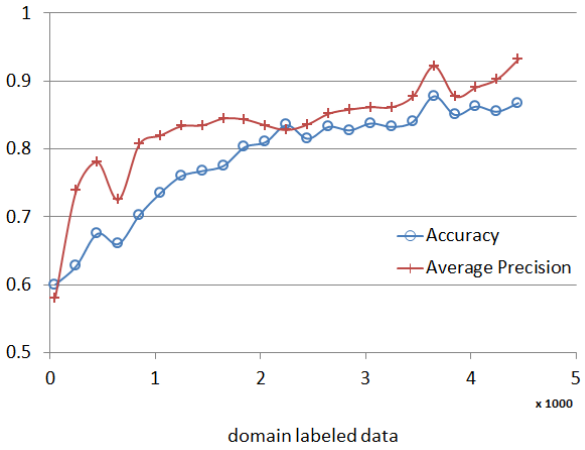

Figure 2: Average Precision and Accuracy learning curve for the proposed method employing all 9 features.

vestigate whether this method, and in particular, the proposed features are useful in detecting users' beliefs in a rumor that they post about. Unlike retrieval, detecting whether a user endorses a rumor or refutes it may be possible using similar methods regardless of the rumor. Intuitively, linguistic features such as negation (e.g., "obama is not a muslim"), or capitalization (e.g., "barack HUSSEIN obama ..."), user history (e.g., liberal tweeter vs. conservative tweeter), hashtags (e.g., #tcot vs. #tdot), and URLs (e.g., links to fake airfrance crash photos) should help to identify endorsements.

We perform this experiment by making a pool of all the tweets that are marked as "rumorous" in the annotation task. Table 2 shows that there are 6,774 such tweets, from which 2,971 show belief and 3,803 tweets show that the user is doubtful, denies, or questions it.

Using various feature settings, we perform 5-fold cross-validation on these 6,774 rumorous tweets. Table 6 shows the results of this experiment in terms of F-score, classification accuracy, and win/loss ratio, the ratio of correct classification to an incorrect

classification.

7 Conclusion

In this paper we tackle the fairly unaddressed problem of identifying misinformation and disinformers in Microblogs. Our contributions in this paper are two-fold: (1) We propose a general framework that employs statistical models and maximizes a linear function of log-likelihood ratios to retrieve rumorous tweets that match a more general query. (2) We show the effectiveness of the proposed feature in capturing tweets that show user endorsement. This will help us identify disinformers or users that spread false information in online social media.

Our work has resulted in a manually annotated dataset of 10,000 tweets from 5 different controversial topics. To the knowledge of authors this is the first large-scale publicly available rumor dataset, and can open many new dimensions in studying the effects of misinformation or other aspects of information diffusion in online social media.

In this paper we effectively retrieve instances of rumors that are already identified and evaluated by an external source such as About.com's Urban Legends reference. Identifying new emergent rumors directly from the Twitter data is a more challenging task. As our future work, we aim to build a system that employs our findings in this paper and the emergent patterns in the re-tweet network topology to identify whether a new trending topic is a rumor or not.

8 Acknowledgments

The authors would like to thank Paul Resnick, Rahul Sami, and Brendan Nyhan for helpful discussions. This work is supported by the National Science Foundation grant "SoCS: Assessing Information Credibility Without Authoritative Sources" as IIS-0968489. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the supporters.

References

Floyd H. Allport and Milton Lepkin. 1945. Wartime rumors of waste and special privilege: why some people believe them. *Journal of Abnormal and Social Psychology*, 40(1):3 – 36.

Gordon Allport and Leo Postman. 1947. *The psychology of rumor*. Holt, Rinehart, and Winston, New York.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of l1-regularized log-linear models. In *ICML '07*, pages 33–40.

James Berger. 1985. *Statistical decision theory and Bayesian Analysis (2nd ed.)*. New York: Springer-Verlag.

Albert Bifet and Eibe Frank. 2010. Sentiment knowledge discovery in twitter streaming data. In Bernhard Pfahringer, Geoff Holmes, and Achim Hoffmann, editors, *Discovery Science*, volume 6332 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin / Heidelberg.

Jean Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.*, 22(2):249–254.

Philip Clarkson and Roni Rosenfeld. 1997. Statistical language modeling using the cmu-cambridge toolkit. *Proceedings ESCA Eurospeech*, 47:45–148.

Nicholas DiFonzo and Prashant Bordia. 2007. Rumor, gossip, and urban legend. *Diogenes*, 54:19–35, February.

Nicholas DiFonzo, P. Prashant Bordia, and Ralph L. Rosnow. 1994. Reining in rumors. *Organizational Dynamics*, 23(1):47–62.

Rob Ennals, Dan Byler, John Mark Agosta, and Barbara Rosario. 2010. What is disputed on the web? In *Proceedings of the 4th workshop on Information Credibility*, WICOW '10, pages 67–74.

Jianfeng Gao, Galen Andrew, Mark Johnson, and Kristina Toutanova. 2007. A comparative study of parameter estimation methods for statistical natural language processing. In *ACL '07*.

Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, Boulder, CO, USA.

Ahmed Hassan, Vahed Qazvinian, and Dragomir Radev. 2010. What's with the attitude? identifying sentences with attitude in online discussions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1245–1255, Cambridge, MA, October. Association for Computational Linguistics.

Edward S Herman and Noam Chomsky. 2002. *Manufacturing Consent: The Political Economy of the Mass Media*. Pantheon.

Courtenay Honeycutt and Susan C. Herring. 2009. Beyond microblogging: Conversation and collaboration

via twitter. *Hawaii International Conference on System Sciences*, 0:1–10.

Jeff Huang, Katherine M. Thornton, and Efthimis N. Efthimiadis. 2010. Conversational tagging in twitter. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, HT '10, pages 173–178.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to its Methodology*. Beverly Hills: Sage Publications.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: Can we trust what we rt?

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL'04*, Morristown, NJ, USA.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of conference on Empirical methods in natural language processing, EMNLP'02*, pages 79–86.

Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. 2010. Detecting and tracking the spread of astroturf memes in microblog streams. *CoRR*, abs/1011.3768.