RumorLens: A System for Analyzing the Impact of Rumors and Corrections in Social Media

Paul Resnick[1], Samuel Carton[1], Souneil Park[2], Yuncheng Shen[1], Nicole Zeffer[1]
[1] University of Michigan School of Information
[2] Telefonica Research
{presnick, scarton, yuncheng, nzeffer}@umich.edu, souneil.park@telefonica.com

ABSTRACT
Some rumors spread quickly and widely through social media. Journalists write about them, both to help the public understand whether they are true, and to help the public understand how widely misinformation and corrections have spread, and how they did. We describe RumorLens, a suite of interactive tools that are designed to help journalists identify new rumors on Twitter and assess the audiences that rumor and correction tweets have reached. The tools make efficient use of human labor to assess whether a rumor's content is interesting enough to warrant further exploration, to label tweets as spreading, correcting, or unrelated to the rumor, and to analyze the rumor visually. Behind the scenes, automated learning and computation amplifies the effectiveness of that labor, making it feasible to engage journalists and the broader public to run a continuous rumor-monitoring service.

Author Keywords
Social Media; rumor; visualization; data mining; machine-learning; active-learning; information retrieval

ACM Classification Keywords
H.1.2. User/Machine Systems: Human Information Processing. H.3.3. Information Search and Retrieval. H.4.3. Communications Applications: Information browsers.

INTRODUCTION
Analyzing the diffusion of rumors (i.e., disputed factual claims) is often as important as checking their truthfulness. For critical rumors that involve serious economic or social harm, diffusion analysis allows an assessment of the potential damage. More generally, it provides a means to understand an interesting aspect of public communication from various perspectives; for example, the underlying psychological mechanisms [1] and the influence of social networks [2], and political beliefs [3].

Today, journalists sometimes write stories about specific rumors. They would benefit from tools that help them do several things. First, tools should help them discover new rumors early, minutes or hours after they start spreading rather than days. Second, tools should help journalists assess whether a rumor has reached a big enough audience to be worth reporting on. Third, tools should help them begin an investigation of the rumor's truthfulness, by providing easy access to tweets that argue for and against its veracity. Fourth, it should provide tools for analyzing the diffusion of the rumor, including the size and overlap of audiences for tweets spreading the rumor or corrections, and the identity of individual tweeters.

In this paper, we present RumorLens, a data mining pipeline and visual analysis tool that will meet those needs. We ultimately intend for the RumorLens pipeline to become the nucleus of a web community dedicated to the discovery and analysis of online rumors. We hope for the RumorLens community to serve the needs of journalists, social scientists, and anyone else with an abiding interest in the spread of misinformation online.

RELATED WORK: APPROACHES TO RUMOR TRACKING, ANALYSIS, AND CORRECTION
There have been some organized efforts to identify, analyze, and investigate rumors, relying on human effort with little computational support. For example, the sites politifact.com and factcheck.org investigate the veracity of claims made in political ads, speeches, and debates. Snopes.com tracks rumors more generally, with a special focus on urban legends. These sites welcome the general public to suggest rumors that are worth investigating: staff decide which to investigate and conduct the investigations. Many other sites and user communities share rumors about specific topics (e.g., macrumors.com, nfltraderumors.com, TMZ.com). In addition to using the crowd to identify candidate rumors, one site, NewsTrust, now defunct, experimented with crowdsourcing the process of rumor investigation [4].

Several systems offer new channels for disseminating the results of fact-checking investigations and automatically integrating them into users' existing reading activities. For example, Truth Teller[1] is a prototype that aims to incorporate fact-checking into any political speech video. It tries to detect claims and link them to entries in a knowledge base. Lazy Truth[2] is a similar tool that targets emails; it tries to detect unverified claims in an email and forward corrections from existing resources such as politifact.com. Truth Goggles [5] implements a similar approach in a browser plug-in, aiming to foster critical thinking in web browsing sessions.

Many automated data-mining tools help with identifying trending topics (some of which may be rumors), retrieving related tweets, and classifying them as either spreading or correcting the rumor. These are discussed and cited in the papers describing the identification and retrieval components of our system [6, 7]. Though not specific to rumors, Truthy (Ratkiewicz, 2011) is a system for detection of political astroturfing campaigns. The Guardian developed an interactive visualization of the tweets about several rumors related to the London Riots of 2011 [8].

MAJOR CHALLENGES IN RUMOR ANALYSIS

1. Systematic, early detection of rumors

The massive quantity of posts in social media make it challenging to detect circulating rumors. The rumors that are covered in the news media are usually the ones that reach a wide audience, or those related to major events. It would also be beneficial, however, to detect both less widespread rumors and rumors in their infancy, when they might be more easily counteracted.

With current tools, searching for rumors in social media is a laborious task that requires multiple iterations of content exploration and evaluation. At the exploration stage, a journalist may start by monitoring his or her news feed or querying on the latest controversial issues with generic terms. Then the journalist will have to go through a mix of various content such as text and video clips, and check if any of them represents a distinct rumor. Such an approach takes a lot of time, and success depends on chance and the heuristic techniques of individual journalists.

2. Collection of rumor and correction posts

Once a target rumor is identified, it is valuable to retrieve as many related posts as possible (tweets, in the case of Twitter). Doing so can present a challenge. For example, a journalist tracking rumors related to this year's unrest in Ferguson, Missouri through the "#IfTheyGunnedMeDown" Twitter hashtag might never even discover the whole segment of Twitter users expressing roughly the opposite sentiment with the "#WeAreDarrenWilson" hashtag. It is easy to get some example tweets through the Twitter search engine, but comprehensive retrieval is a very difficult task, requiring both high recall and high precision where traditional retrieval mostly emphasizes precision alone. High recall, meaning collecting *almost all* posts related to the rumor, requires generating a set of queries to existing search interfaces that, taken together, cover all the posts. However, such a set of queries is likely to match many tweets that are not related to the rumor. High precision, avoiding these false positives, requires making this set of queries as small and specific as possible so they don't match unrelated tweets. In addition, among the related tweets, it will be valuable to automatically categorize them as to whether they are spreading or correcting the rumor.

3. Making Sense of the Collected Tweets

After collecting the tweets, the journalist may want to explore various questions about the data to understand the [9] diffusion and impact of the rumor. Such questions might include whether the rumors or the corrections had larger audiences, to what extent the audiences overlapped, whether corrections were effective, and who were the important players in the spread of rumor and correction. Each of these questions requires a different way of aggregating the collected data and some of them require additional data about the followers of the authors of the posts.

RUMORLENS ARCHITECTURE

RumorLens is a software pipeline of three components, each of which addresses one of the challenges above. Figure 1 shows the overall architecture of the system. The three core components are the following: Rumor Detector; ReQ-ReC Interactive Retriever; and Interactive Visualizer.

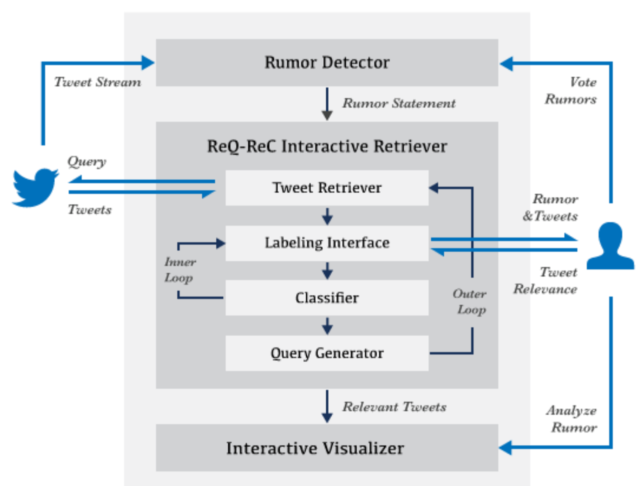

Figure 1: RumorLens system architecture

1. Rumor Detector

The Rumor Detector component mines rumors from tweets obtained through the Twitter Garden Hose API (a 10% sample of the overall tweet stream), and produces a continuously-updated, ranked list of tweet clusters that seem to be rumors.

[1]http://www.washingtonpost.com/blogs/ask-the-post/wp/2013/09/25/announcing-truth-teller-beta-a-better-way-to-watch-political-speech/

[2] http://www.lazytruth.com/

The technique used by the Rumor Detector is to search for a set of expressions commonly employed in the flagging of controversial claims but rarely otherwise (e.g., “Is this true?”) The technique used by the detector produced higher recall than baselines based on overall trending topics or trending hashtags [7]. In addition, the mean time to first detection of a candidate rumor was less than 10 minutes after the first post about it.

The RumorLens community website will include an interface, a prototype of which is shown in Figure 2, with an upvote/downvote mechanism that allows users to refine the output of the Rumor Detector component. Upvotes indicate candidate rumors that users think are worth investigating further. As more feedback accrues, the detector will become more and more accurate in identifying rumors.

2. ReQ-ReC Retriever and Classifier

We have a developed a process called ReQuery-ReClassify (ReQ-ReC) for retrieving and classifying tweets related to a particular rumor. People provide judgments about particular tweets, which leads to updating a classifier (the ReClassify part). Occasionally, the system generates additional queries (the ReQuery part). By carefully choosing a small subset of the tweets for labeling and queries to run, the system achieves good precision and recall, with a feasible amount of human labeling (approximately 200 tweets for each rumor).

The ReQ-ReC system yielded a 20%-30% improvement over iterative relevance feedback, the baseline state-of-the-art, on standard TREC retrieval tasks [6].

3. Interactive Visualization

The RumorLens interactive visualization allows users to explore the data collected by the ReQ-ReC retrieval system. It enables an accurate estimation of the impact of a rumor through a state-transition-based visualization that is specific to rumor diffusion. We treat user states—such as exposure to the rumor—as nodes, and flows of people between states as links. Link width represents the number of people flowing between the two states.

The visualization includes the following states for passive exposure: not yet exposed; exposed only to the rumor; exposed only to a correction; exposed to both. For active involvement, it includes states for having tweeted the rumor, having tweeted the correction, and having tweeted both. By downloading the list of follower IDs for each active user from the Twitter API, we can process the posts chronologically in order to efficiently determine which people transition between which states as a result of each tweet.

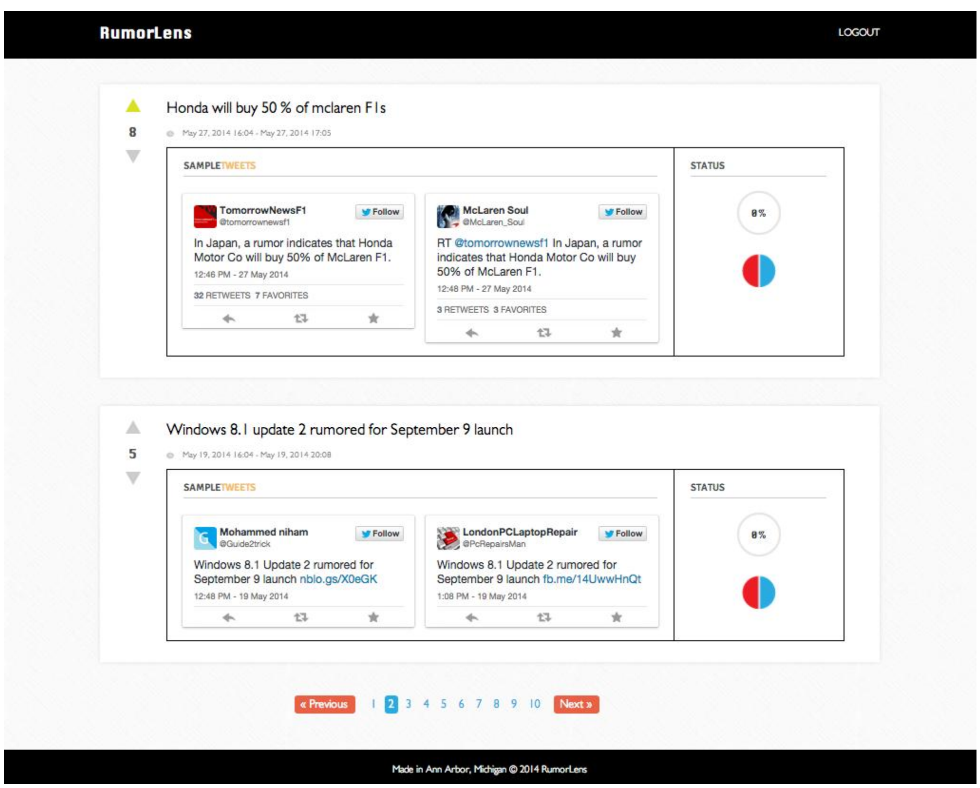

Figure 2: RumorLens community site prototype

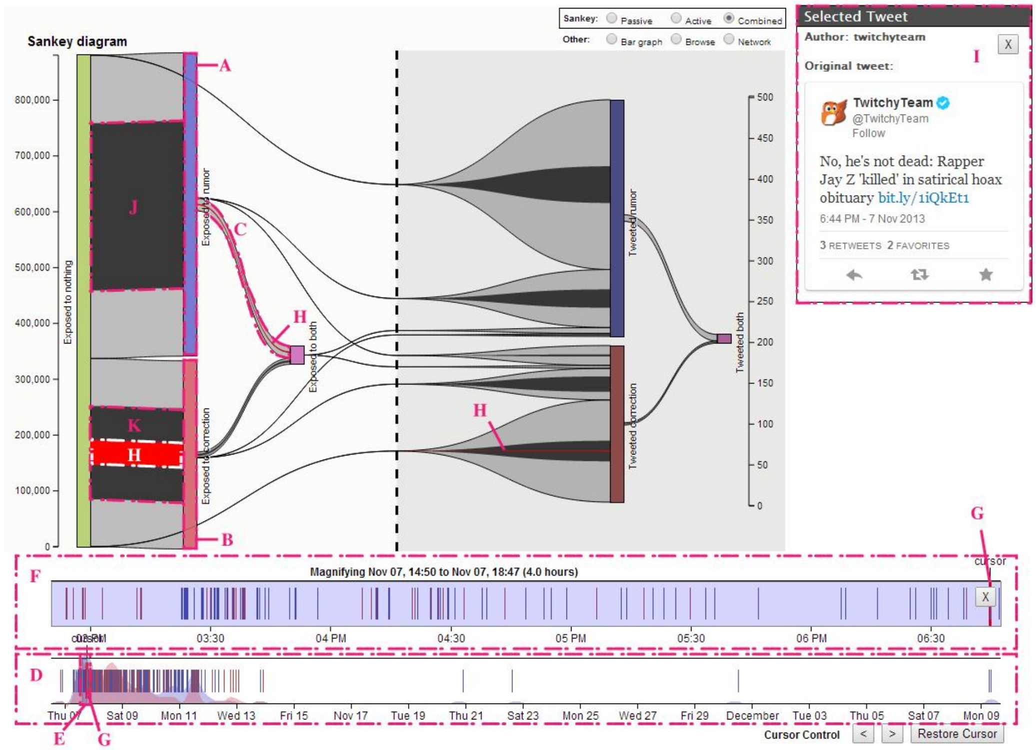

Figure 3: Sankey state transition diagram of the rumor's diffusion

SAMPLE RUMOR ANALYSIS

To illustrate how the visual tool works, we collected 616 tweets related to a rumor that the rapper Jay-Z had died. The source of the rumor was a satirical news article claiming he had "died inside," with respect to his musical integrity[3]. Some of the people who tweeted the claim later recanted, and many others tweeted or retweeted corrections saying that it was a hoax or mocking those taken in.

Revealing the audience. Nearly 900,000 people were followers of someone who tweeted about this rumor. The Sankey diagram (Figure 3) illustrates the movement of these people between different states of interaction with the rumor. Comparing the vertical width of the flows and state nodes allows several inferences to be drawn about the diffusion. For example, over half again as many people were only exposed to the rumor as were only exposed to the correction (compare the size of the "Exposed to Rumor" node labeled A to that of the "Exposed to correction" node labeled B). Second, people exposed to the rumor are rarely exposed to the correction. The flow from "Exposed to Rumor" to "Exposed to both", labeled C, is relatively thin, showing that this transition did not occur many times, and that therefore the audiences of the rumor and the correction remained largely disjoint.

The visualization also includes a timeline of the whole lifetime of the rumor, labeled D, with vertical bars for

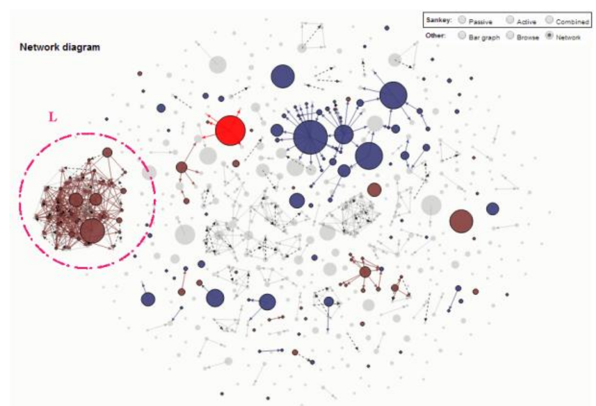

Figure 4: Network diagram of the same rumor

List of tweets

Tweet #	Author	Date	Tweet type	Selected	# of followers of tweet author	# followed by tweet author	Exposed to nothing → Exposed to rumor	Exposed to nothing → Exposed to correction	Exposed to correction → Exposed to both
150	mcfuture95	Nov 07, 18:24	Rumor	false	679	726	612	0	25
151	raycistsarrcasm	Nov 07, 18:26	Rumor	false	363	244	350	0	0
152	thebqe718	Nov 07, 18:29	Rumor	false	362	1,156	228	0	1
153	masayasport	Nov 07, 18:30	Rumor	false	4,629	1,447	4,598	0	1
154	leythug	Nov 07, 18:31	Rumor	false	-1	-1	0	0	0
155	lv3miaaj_xx	Nov 07, 18:34	Rumor	false	599	473	570	0	5
156	hyfr_dan	Nov 07, 18:38	Rumor	false	55	54	58	0	0
157	ticketradio	Nov 07, 18:38	Correction	false	26,017	285	0	26,002	0
158	twitchyteam	Nov 07, 18:44	Correction	true	132,923	3,466	0	44,287	0
159	roeyhilliard	Nov 07, 18:44	Correction	false	828	296	0	754	0
160	goldwatergal	Nov 07, 18:44	Correction	false	6,252	6,176	0	5,504	0
161	kevinwhipp	Nov 07, 18:44	Correction	false	1,918	1,916	0	1,189	0
162	rightofcenternc	Nov 07, 18:44	Correction	false	62	247	0	44	0
163	amagdalina	Nov 07, 18:46	Rumor	false	836	946	781	0	8
Sum:	N/A	N/A	N/A	1	1,370,864	513,799	299,270	167,137	4,244

Figure 5: Browsable list of tweets related to the rumor

[3] http://therapinsider.com/2013/11/06/rapper-jay-z-found-dead-inside-at-43/

individual tweets, colored blue for rumors and red for corrections. A log-scale histogram in the background indicates tweet density.

Effects in different time intervals. Selecting a time interval, labeled E, on the lower timeline causes the selected time interval to appear, magnified, in a second upper timeline, labeled F. Clicking on a vertical bar on either timeline highlights that tweet on the timeline (G), on the diagram (H), and brings the content of the tweet up in the rightmost pane (I).

The impact of the tweets in the selected interval is shown as a black overlay on the Sankey diagram. The interval can be dragged across the timeline and the black overlay updates instantaneously. This allows for visual exploration. For example, comparing the overlay on the link from "Exposed to nothing" to "Exposed to rumor" (J) with the overlay on the link from "Exposed to nothing" to "Exposed to correction" (K) we see that more than half of all exposure to these two viewpoints occurred during the tiny selected time interval (E) of 4 hours, and that the ratio of exposure to the rumor and correction during this time period was roughly the same as for the overall diffusion, about 3:2.

Prominent individuals and tweets. RumorLens offers alternatives to the Sankey diagram that let an analyst discover important individuals and tweets. The network diagram (Figure 4) displays tweets as circular nodes, colored according to whether they propagated the rumor or the correction, and sized by how many people they exposed to that information for the first time. Manipulating the timebar causes nodes to fade to grey as they pass out of the selected time interval, while selecting a tweet causes that tweet to light up in a brighter color. A solid arrow between two tweet nodes indicates that the target tweet was made by a follower of the author of the source tweet, later in time. Thus the solid arrows indicate potential lines of influence between tweets. Dashed arrows connect consecutive tweets by the same author.

The network diagram shows that the highlighted time interval contained a small number of highly prominent tweets, which account for the large portion of overall exposure depicted in the Sankey diagram. It also shows that a community (labeled L) of like-minded twitter accounts all tweeted the correction in rapid succession, which might bear further investigation.

Finally, the tweet list (figure 5), shows all the raw data of the other two diagrams in browsable list form. Clicking on column headers sorts the list by that column, allowing for efficient discovery of outliers.

CONCLUSION

RumorLens is a promising tool for combining human effort with computation behind the scenes to systematically detect new rumors in Twitter, retrieve almost all the tweets related to them, and interactively analyze how many people tweeted about them or were exposed to the rumor or a correction. It depends on human labor. The amount required, however, is small enough that it seems plausible that journalists and enthusiasts about particular topics might voluntary provide the labor.

ACKNOWLEDGMENTS

This work is partially supported by the National Science Foundation under grant numbers IIS-0968489. This work is also partially supported by Google.

REFERENCES

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

[2] Doerr, B., Fouz, M. and Friedrich, T. Why rumors spread so quickly in social networks. *Communications of the ACM*, 55, 6 2012), 70-75.

[3] Garrett, R. K. Troubling consequences of online political rumoring. *Human Communication Research*, 37, 2 2011), 255-274.

[4] Florin, F. *Crowdsourced Fact-Checking? What We Learned from Truthsquad*. 2010. Retrieved Sep 5, 2014 from http://www.pbs.org/mediashift/2010/11/crowdsourced-fact-checking-what-we-learned-from-truthsquad320/

[5] Schultz, D. D. E. P. *Truth Goggles: Automatic incorporation of context and primary source for a critical media experience*. Massachusetts Institute of Technology, 2012.

[6] Li, C., Wang, Y., Resnick, P. and Mei, Q. ReQ-ReC: high recall retrieval with query pooling and interactive classification. In *Proceedings of the 37th international ACM SIGIR conference on Research &; development in information retrieval* (Gold Coast, Queensland, Australia, 2014). ACM.

[7] Zhao, Z., Resnick, P. and Mei, Q. *Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts*. in submission.

[8] Dant, A. and Richards, J. *Behind the rumours: how we built our Twitter riots interactive*. 2011. Retrieved Sep 5, 2014 from http://www.theguardian.com/news/datablog/2011/dec/08/twitter-riots-interactive

[9] Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Patil, S., Flammini, A. and Menczer, F. *Truthy: mapping the spread of astroturf in microblog streams*. ACM, City, 2011.