2-2010

Estimating the Quality of Postings in the Real-time Web

Hady W. LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Alexandros NTOULAS
Microsoft Research

Krishnaram KENTHAPADI
Microsoft Research

Citation

LAUW, Hady W.; NTOULAS, Alexandros; and KENTHAPADI, Krishnaram. Estimating the Quality of Postings in the Real-time Web. (2010). *Workshop on Search in Social Media 3rd SSM 2010 co-located with WSDM, February 3*. 1-4. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/1556

Estimating the Quality of Postings in the Real-time Web

Hady W. Lauw
Microsoft Research
1065 La Avenida
Mountain View, CA 94043
hadylauw@microsoft.com

Alexandros Ntoulas
Microsoft Research
1065 La Avenida
Mountain View, CA 94043
antoulas@microsoft.com

Krishnaram Kenthapadi
Microsoft Research
1065 La Avenida
Mountain View, CA 94043
krisken@microsoft.com

ABSTRACT

Millions of users are posting their status updates, interesting findings, news, ideas and observations in real-time on microblogging services such as Twitter, Jaiku and Plurk. This real-time Web can be a great resource of valuable timely information. Since the real-time Web is completely open and decentralized and anyone may post information at whim, distinguishing interesting and popular postings from the mundane ones is a challenging task. In this paper we study the problem of estimating the quality (or "interestingness") of postings in the real-time Web. We identify several important factors that are indicative of the quality of postings, and present metrics that capture these factors. To showcase the promise of our approach, we present early experimental results on Twitter.

1. INTRODUCTION

Millions of users on the Web today are using services such as `Twitter`, `Jaiku` and `Plurk` to post their status updates, findings, news, ideas and observations in a real-time fashion. Such postings represent a great variety of experiences, and can serve as a means of getting timely information on various events or other happenings in the world. To exemplify the power of the real-time Web, consider the recent breaking-news event,[1] when an aircraft with both engines failing ended up with a safe landing on the Hudson river. One of the first pictures of the event appeared on Twitter[2].

However, not all the information in the real-time Web is important or meaningful. While some users do post information that is of interest to many people, many others post mundane details that are of interest only to a very limited set of people, if not only to the authors, e.g., "I had eggs for breakfast". In order to realize the full potential of the real-time Web, we need a way to distinguish the important (e.g., the Hudson river landing) postings from the mundane ones.

In this paper, we look at the problem of estimating the quality (or "interestingness") of a posting in the real-time Web. Our high-level intuition is that interesting postings will appear quite often (possibly with variations) within the real-time Web, and will be seen and re-posted by several different users within a short period of time. We will describe the desirable properties of the interesting postings and present metrics for capturing these properties. We showcase their potential and performance with experimental results on the most popular of the microblogging services, i.e., `Twitter.com`.

2. QUALITY OF POSTINGS

Our goal is to determine the quality or "interestingness" of a posting in the real-time Web. The quality of a posting depends on a number of factors, which we will study in this section. In the following discussion, we denote a posting (or tweet) as a tuple $\tau = \langle u, m, t \rangle$, where u is the user who posted the text message m at time t. A message is typically short, under 140 characters.

Size of the story. Intuitively, an important posting is one that appears as part of a larger news story. The more postings that we can identify as belonging to the same event, the more important is the event, and correspondingly the more important are the individual postings arising from the event. Hence, we can use the number of postings belonging to the same event as one indication of importance.

The main challenge is to identify which postings are about the same event in a robust way. Services like Twitter allow users to specify tags, which are words preceded by '#' (e.g., #hudson) within their postings. Tags are generally used to indicate that a posting is about specific topics, and thus they could be a good proxy for grouping related postings. However, we cannot rely on tags alone, because there may be several different tags used for the same event (e.g., #hudson and #hudsonaccident), which we may not be able to enumerate completely.

In order to group related postings in a more robust way we employ a state-of-the-art single-link clustering algorithm. In addition to the tags, we also measure the similarity between the text message m of the postings. To this end, we use a vector-space representation for each posting, where each element in a posting's vector corresponds to a TF.IDF weighted value for word w in a given message m. We use

[1] `http://www.msnbc.msn.com/id/28678669/`

[2] `http://www.bbc.co.uk/blogs/technology/2009/01/twitter_and_a_classic_picture.html`

the cosine similarity of the vectors as the similarity metric of two postings.

The clustering proceeds as follows. For each posting, and in order of appearance, we compute its similarity to the existing clusters and we assign it to the cluster c with the highest similarity above a threshold. If no such cluster is found, we create a new singleton cluster with only one posting. Here, we use the centroid of the postings within a cluster as its vector representation.

We can now define the $clusterSize$ metric that captures the importance of a posting in terms of the size of the cluster it belongs to. $clusterSize_t(c)$ is the actual number of postings within the cluster c at time t. Note that since a cluster contains very similar postings, the metric assigned to a posting is the same as the metric assigned to a cluster. Hereonafter, we use *posting* and *cluster* interchangeably with respect to the metrics.

Re-postings (re-tweets). Certain micro-blogging services, such as Twitter, allow users to re-post a posting that they have seen. Re-posting (or re-tweeting) is essentially a simple way for a given user to endorse a posting that she has read and liked by re-posting it for her friends to see. In a sense, a re-posting is a vote of confidence. Intuitively, the more re-postings that a posting receives, the more interesting is the posting.

This information can be used either to filter postings (e.g., operate only on postings having at least 2 re-posts), or to be incorporated directly in determining the posting quality. In order to find the most interesting postings, we take it one step further, and use this in conjunction with the clustering. Hence, we only consider clusters containing only re-postings by at least two different users, while allowing for at most one original posting (not a re-post) in a cluster.

Size of the audience. The importance of a posting also depends on the number of users that see it. It may seem at first glance that the greater the number of users who have seen a given posting, the more important it is. However, this may unduly favor users that have a huge number of followers. For instance, a mundane post by a celebrity with a huge following may overshadow an important post by less known authors.

Instead, our intuition is that the audience size has to be seen with respect to the number of re-postings. A posting that generates a larger number of re-postings out of a smaller audience size is likely to be important, reflecting the degree of excitement of the audience who then go on to re-post the message. In order to capture this notion of importance, we use a metric called $audienceSize$ that is expressed by the expected number of users that have seen a given posting or its variations in a cluster.

We formally define $audienceSize$ metric of a cluster c at time t as:

$$audienceSize_t(c) = \sum_{\tau \in c} seen_t(\tau) \times followers_t(\tau.u)$$

where $seen_t(\tau)$ is the fraction of users that have seen the posting τ by time t, and $followers_t(\tau.u)$ is the number of followers at time t of the user u who is the author of τ.

Given the fast-paced nature of the real-time Web, estimating $seen_t(\tau)$ is not straightforward. If a user re-posts a posting, we can be fairly certain that she has seen the posting. Otherwise, there is no reliable way of knowing when a user reads a posting (and decides not to act on it). As an approximation of how many users have seen a given posting, we use the logging-in rates of a set of 200 thousand users in Twitter, by measuring the average interval between two consecutive postings by the same user. This assumes that the average user generally logs in and out to post each message.

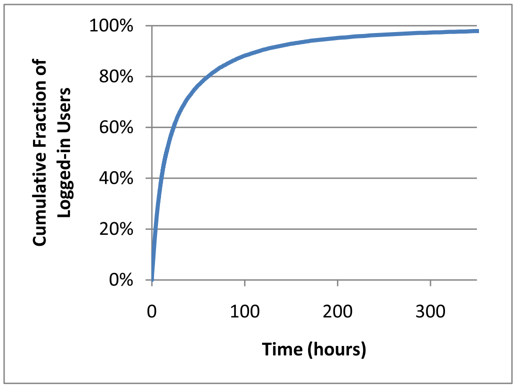

Figure 1: Cumulative Fraction of Logged-in Users

In Figure 1, we plot a cumulative frequency graph of the fraction of users in our dataset (to be described in Section 3) who have logged in after a certain period of time. For example, after 100 hours, about 85% of the users have posted at least once in Twitter. We can combine the numbers in Figure 1 with the followers of a given user in order to estimate how many users are expected to have seen a posting after a given time period. For example, if a user has 100 followers, we expect that, roughly 85 of them will have seen her posting after 100 hours.

Time interval. An additional factor to the importance of a posting is its recency relative to other postings in the same cluster. Intuitively, if there are many postings around the same topic within a narrow time period, then these postings are more important as they indicate a high level of interest and activity. To this end, we use a $interval$ metric, defined as $interval_t(c) = \tau_k.t - \tau_0.t$, where $\tau_k.t$ is the timestamp (in minutes) of the newest posting within a cluster, and $\tau_0.t$ is the timestamp of the oldest posting in the cluster. The smaller the $interval$ metric, the more concentrated in time the postings are expected to be.

Quality. Estimating the final quality of a posting (or a cluster) can be performed in a number of different ways. For example, based on our discussion above, one straightforward way (that can potentially serve as a baseline) would be to consider the postings that belong to larger clusters as the more important ones. This approach aims at directly capturing the popularity of a given topic, and thus makes the assumption that the most popular clusters are the most interesting ones.

In addition to this baseline approach, we consider a posting quality estimation which combines the metrics that we have discussed so far. More specifically, we will use the following equation as our quality estimation function for a given posting in a cluster c.

$$quality_t(c) = \frac{clusterSize_t(c)}{log(audienceSize_t(c))} \times \frac{1}{log(interval_t(c))} \quad (1)$$

The first term $\frac{clusterSize_t(c)}{log(audienceSize_t(c))}$ approximates the rate

Window	User	Time	Message
13:00-16:00	aaronmbaer	15:22	IT’S SOOO ON: @Twitter/@TMZ vs @CNN/Cable News - Who can break the MJ REAL story first
14:00-17:00	SCMcCarthy	16:55	RT @anildash: Probably the single best Michael Jackson clip you’ve never seen. http://bit.ly/illbethere (via@susanorlean)
15:00-18:00	Kels_bels	17:56	RT @marclamonthill: Come on y’all. Chill with the negative Michael Jackson tweets. Please allow his passing to occur with grace and dignity.
16:00-19:00	devonbowers	18:07	RT @OMG_Ponies: @ iPhone developers: We need an ”Abe Vigoda is still alive” app.
17:00-20:00	paulstenis	19:57	RT @levjoy Upon finding dearth of MJ videos on MTV, wife of @levjoy says ”shame on you MTV.”

Table 1: Top Cluster in each Window with Quality Scoring

Window	User	Time	Message
13:00-16:00	7sexysecrets	13:00	Top SEO Services - How Can You Tell?
14:00-17:00	KSATGMSA	14:44	Breaking: TMZ.com is reporting Michael Jackson is dead at the age of 50. http://is.gd/1dtF3 - story still developing.
15:00-18:00	EricBowling	15:00	@melissaanelli ”Georgia . . . on my mind. . . . And Melissa and Leaky!”
16:00-19:00	AndrewCMiller	16:00	I guess the girl really is mine now. - @paulmccartney
17:00-20:00	ASOS_Natalie	17:00	@princepelayo yes and so are the press

Table 2: Top Cluster in each Window Based on Cluster Size

of re-postings by people who see the posting or its variations. The higher its value, the larger the number of re-postings ($clusterSize_t(c)$ is larger) out of a smaller audience size ($audienceSize_t(c)$ is smaller). The second term denotes the preference for clusters with smaller time interval between the oldest and newest postings.

3. EXPERIMENTS

Dataset. Each user page in Twitter displays the user’s most recent twenty postings, as well as up to thirty six other users whom she follows. Our dataset was obtained by crawling the Twitter site for such user pages, starting from the home page (twitter.com) and following outgoing links. The crawl ran in July 2009 for about a month, and a vast majority (around 90%) of the postings in the crawl were from the preceding three-month period, i.e. May to July 2009. This dataset contains 3 million tweets by 200 thousand users, and 6.5 million following relationships.

Prior to experiments, we pre-processed the data as follows. We removed duplicate postings due to the same user being crawled more than once at different times. We also retained postings with up-to-the-minute timestamps (e.g., “7:47 AM Jul 22nd”), and removed postings with imprecise timestamps (e.g., “about 3 hours ago”). These resulted in the loss of 10% of the postings. In addition, we also removed punctuation marks, and Porter-stemmed each word in the postings.

Event. To showcase the result of our proposed method, we focus on a news-breaking event on June 25, 2009: the death of Michael Jackson. We considered five overlapping time windows on June 25, 2009 spanning 13:00 to 20:00 hours, which was around the time when the story first broke. Each window is of three hours long.

For each window, we read the postings in the chronological order, clustered the postings, and finally assigned them quality scores as of the time when the window ended. Postings in a given cluster are similar content-wise, and we pick a random posting to represent a cluster. We compare two scoring functions, our proposed quality scoring function in Equation 1 and a baseline function that takes into account only the cluster size $clusterSize$.

The top postings/clusters at the end of each window based on our quality scoring function are shown in Table 1. Notably, the top postings are meaningful; they are not merely mundane reports. For four of the five windows, the top postings concerned Michael Jackson’s death. For instance, the top posting in the first window (*13:00-16:00*) by user *aaronmbaer* questioned who could break the Michael Jackson story first. The top postings also tend to be the most recent in their respective windows, e.g., at time *15:22* for the window *13:00-16:00*.

As a baseline, we consider another scoring function that simply relies on the cluster size, i.e., the largest cluster is ranked first, and does not restrict it to re-postings. The top postings for the same windows are shown in Table 2. Unlike the previous case, the posting about Michael Jackson’s death only came in top in the second window. The top postings here are also older, closer to the beginning of the window. The reason is that older postings have more time to accumulate larger clusters. Compared to Table 1, the top postings tend to be less interesting (e.g., “Top SEO Services - How Can You Tell?”), and less relevant (only one window has Michael Jackson as the top posting).

4. RELATED WORK

There have been a few recent studies on analyzing data from micro-blogging services. The geographical and topological properties of the Twitter network were explored in [4, 5, 1]. Huberman et al. [2] observed that a sparse network of real friends (defined as those to whom a user has directed a post at least twice) is a better predictor of user activity in Twitter than the network of declared friends and followers. Zhao et al. [7] conducted a study of 11 users to understand the motivations for tweeting and how Twitter provided relational and personal benefits. Jansen et al. [3] used Twitter to track consumer sentiments towards certain brands. Weng et al. [6] proposed TwitterRank, a PageRank-like algorithm to measure the topic-sensitive influence of the users in Twitter.

5. CONCLUSION

In this paper, we looked at the problem of estimating the quality of postings in the real-time Web. We identified several factors that are likely good indicators of the quality of a posting, namely the size of the story, re-postings, audience size, as well as time interval, and proposed a quality estimation function combining these factors. Our preliminary experiments on a dataset collected from Twitter looked promising in identifying the most interesting stories in a stream of postings. This is still an ongoing work, and we look forward to developing this work further, and conducting more comprehensive experiments.

6. REFERENCES

[1] A. Cheng and M. Evans. Inside Twitter, an in-depth look inside the Twitter world, 2009. http://www.sysomos.com/insidetwitter/.

[2] B. A. Huberman, D. M. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. *First Monday*, 14(1), 2009.

[3] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 2009.

[4] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: Understanding microblogging usage and communities. In *Joint 9th WEBKDD and 1st SNA-KDD Workshop*, 2007.

[5] B. Krishnamurthy, P. Gill, and M. Arlitt. A few chirps about Twitter. In *WOSP'08: Proceedings of the 1st Workshop on Online Social Networks*, 2008.

[6] J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank: Finding topic-sensitive influential twitterers. In *WSDM*, 2010.

[7] D. Zhao and M. B. Rosson. How and why people twitter: The role that micro-blogging plays in informal communication at work. In *GROUP'09: Proceedings of the ACM 2009 International Conference on Supporting Group Work*, 2009.