Information Propagation on Twitter

Eldar Sadikov
Stanford University
eldar@cs.stanford.edu

Maria Montserrat Medina Martinez
Stanford University
mmmedina@stanford.edu

ABSTRACT

We analyze URL and tag propagation on Twitter social network with 54 million nodes and 1.5 billion edges, one of the largest social networks studied in academia. We specifically focus on the interplay of external and network influences. We attribute propagation to the network whenever a user mentions a tag or URL after one of its neighbors has previously mentioned it, and to external influence otherwise. We develop a new metric to measure external influence and an efficient algorithm to calculate it. The insight we obtain from the external influence metric paired with the analysis of cascade dynamics of the network influence, not only validates some of the previously observed phenomena in other social networks but also provides new insight into the interplay of network and external influences over the lifetime of memes in the network.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications—*Data mining*

General Terms

Experimentation, Measurement, Algorithms

Keywords

Social networks, information propagation

1. INTRODUCTION

Twitter is a social networking and micro-blogging service which has become popular over the last couple of years. Twitter users send and read messages known as *tweets*. Tweets, as shown in Figure 1, are text-based posts up to 140 characters long and are delivered to the author's subscribers, known as *followers*. Such subscriptions form directed not always symmetric connections, where user A may follow user B and, hence, receive A's tweets, but user B may not follow A and may not receive B's tweets. These connections naturally form a directed *social network* where nodes are the users of the service and edges represent the subscriptions.

> Secretary of state Clinton announces 2012 Int'l AIDS conference to be held in U.S.: http://bit.ly/8Bc #WorldsAIDSDay

Figure 1: An example of a tweet

Tweets, as shown in Figure 1, often contain *URLs* and *tags*. Tags are tokens prefixed with a #, e.g. "WorldsAIDSDay" in Figure 1, and often used to indicate the tweet's topic or event. On the other hand, URLs are typically HTTP links to news articles, pictures, or videos.

In this paper, we study the propagation of URLs and tags over the Twitter social network. Specifically, we look at the users (i.e. nodes in the network) that mention a particular URL or tag in chronological order, their distribution over the network, and the presence/absence of edges between them. We will generally refer to such chronological distribution as *information propagation*, where information, in our context, will refer to the URLs and tags.

There are a number of factors that make Twitter interesting for studying information propagation and make our work different from the previous work in the area. First of all, Twitter, unlike much of the previously studied media such as news feeds, blogs, emails, has an explicit social (i.e. subscription) network. Our work is one of the first ones to look at information propagation on a real social network which is, on top of all, one of the largest ones known to date. Second of all, because information on Twitter often spreads in the form of tags and URLs, there is no burden of data cleaning and thus no potential bias in results. Thirdly, Twitter URLs and tags, in themselves, are interesting to study as people often now rely on them to find trends and news on the web.

While Twitter users can discover information directly from the network, i.e. by reading the tweets of the users they follow, they can equally discover information from outside of the network, e.g. from TV, newspapers, online news aggregators (e.g., Google News), etc. In this work, we attempt to understand how to differentiate between the two cases and propose new metrics to quantitatively analyze them. To our knowledge, this is one of the first attempts to study the contribution of external sources to information spread in social networks. Moreover, most of the work up to date has focused on propagation directly via the network, i.e., in our setting, the case when users find information directly from the people they follow. However, even to this respect, our findings not just confirm the already observed phenom-

ena, but also unveil new interesting dynamics. Last but not least, the findings we report in this paper have applications in news and trend detection and tracking.

In summary, our contributions are as follows:

- Basic analysis of Twitter social network properties (Section 3)
- Metric to measure the contribution of external sources to information propagation in a social network: 1) its formal definition, 2) efficient algorithm to compute it, 3) its evaluation on real data and 4) comparison of its behavior on real data to two simple synthetic information propagation models (Section 4)
- Comprehensive analysis of the network contribution to the spread of URLs/tags on Twitter (Section 5)

2. PRELIMINARIES

We begin by introducing our terminology and notation.

Network: Given subscription (i.e. follow) relationships between Twitter users, we define the Twitter *network* as a directed graph $G = (V, E)$, where each node in V is a Twitter user. For any two users $v_1 \in V, v_2 \in V$ there is an edge $(v_1, v_2) \in E$ if and only if v_2 follows v_1. In other words, we introduce an edge from v_1 to v_2 as long as v_2 subscribes to tweets of v_1, and, hence, information in the form of URLs or tags posted by v_1 is observed by (i.e. flows to) v_2.

DEFINITION 1. *For any two nodes $v_i \in V$ and $v_j \in V$, the distance $d(v_i, v_j)$ from v_i to v_j is defined as the shortest directed path in the network from v_i to v_j. If such path does not exist, $d(v_i, v_j) = \infty$.*

Stream:

DEFINITION 2. *Tweet is a triple $\langle v, m, t\rangle$, where $v \in V$ is the user who posted the tweet, m is the text of the tweet (message), and t is the time when the tweet was posted.*

DEFINITION 3. *Stream is a sequence of tweets $\langle\langle v_1, m_1, t_1\rangle, \langle v_2, m_2, t_2\rangle, \ldots\rangle$, where any tweet $\langle v_i, m_i, t_i\rangle$ precedes another tweet $\langle v_j, m_j, t_j\rangle$ iff $t_i \leq t_j$.*

For simplicity, we will assume that we have access to only one stream. This stream may either contain all publicly available tweets or a sample of all the tweets.

DEFINITION 4. *For a given URL or tag x, user v mentions x if there is a tweet $\langle v, m_i, t_i\rangle$ in the stream, such that x is contained within m_i.*

DEFINITION 5. *For a given URL or tag x, the first mention of x by user v is the first tweet $\langle v, m_i, t_i\rangle$ in the stream, such that x is contained within m_i. If v does not mention x, the first mention of x by v does not exist.*

DEFINITION 6. *An infection is a pair of some tag or URL x and a sequence of nodes $\langle v_1, v_2, \ldots, v_n\rangle$ (infection sequence) of some length n, such that a node v belongs to the sequence iff it mentions x and for any i, j, $1 \leq i < j \leq n$, the first mention of x by v_i is before the first mention of x by v_j.*

Note here that according to this definition of infection, a node can appear in the infection sequence only once. In other words, we use Susceptible-Infected-Recovered (SIR) disease propagation model [1], where once infected, a node cannot be reinfected (all mentions of some URL or tag x after the first mention are ignored).

Cascades:

DEFINITION 7. *For a given infection $\langle x, \langle v_1, v_2, \ldots, v_n\rangle\rangle$, we define its infection graph at step k, $1 \leq k \leq n$, as a subgraph $C_k(V_k, E_k)$ of $G(V, E)$, where $V_k = \cup_{1\leq i\leq k} v_i$ and there is an edge $(v_i, v_j) \in E_k$ iff $(v_i, v_j) \in E$ and $i < j$.*

DEFINITION 8. *In a given infection graph $C_k(V_k, E_k)$, we refer to its connected components as cascades.*

DEFINITION 9. *A cascade is said to be non-trivial if its size is greater than 1.*

DEFINITION 10. *A node v_k is said to be cascaded if for a given infection $\langle x, \langle v_1, v_2, \ldots, v_n\rangle\rangle$ and its graph $C_n(V_n, E_n)$, v_k is in a non-trivial cascade.*

DEFINITION 11. *For infection $\langle x, \langle v_1, v_2, \ldots, v_n\rangle\rangle$ and its graph $C_{k+1}(V_{k+1}, E_{k+1})$, $1 \leq (k+1) \leq n$, let V' be the set of incoming neighbors of v_{k+1}. Node v_{k+1} is said to merge cascades if the nodes in V' belong to more than one cascade in the graph $C_k(V_k, E_k)$ at step k of the same infection.*

3. DATA SET

Network: We have been continuously monitoring a sample of all publicly available tweets over the last six month and, from each observed user ID among the collected tweets, we explored the Twitter network structure in a breadth-first-search manner. We collected this way follow relationships (i.e. IDs of people a user follows) for almost 54,310,622 users which is 99.9% of all user IDs known to us. The graph we have obtained has 1,491,979,651 edges with average number of followees (in-degree of a node in the network, per our definition) per user at 27.47.

Here we provide some other basic stats on the collected network.

- Figure 2 shows the out-degree distribution of the network plotted on the log-log scale, i.e. the distribution of the number of followers per user. One can observe that the distribution follows the power law. The power law exponent is 1.95, estimated by the complementary cumulative distribution function (CCDF) method, and 1.84, as estimated by the maximum likelihood estimate (MLE) method.

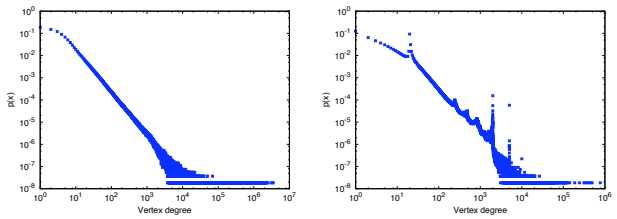

Figure 2: Degree distribution of Twitter network: a) outgoing (followers) b) incoming (followees).

- Figure 2 shows the in-degree distribution of the network plotted on the log-log scale, i.e. the distribution of the number of followees per user. Although there are

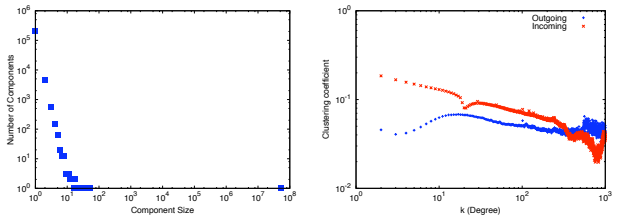

Figure 3: a) Weakly connected components size distribution b) Clustering Coefficient as a function of vertex degree.

a few spikes, overall, the distribution also follows the power law. The power law exponent is 2.13, estimated by the CCDF method, and 1.90, as estimated by the MLE method. The spikes can be explained by the existing follow limits, enforced by Twitter. Specifically, one cannot currently follow more than 2000 people unless she is followed by at least 2000 people. Hence, we observe in Figure 2 a spike at the vertex degree of 2000. The subsequent spikes may also be due to the follow limits: beyond 2000 friends, Twitter enforces follow limits proportionally to the current number of followers. The small spikes before 2000 could be possibly due to other similar constraints enforced by Twitter.

- Figure 3 shows the distribution of weakly connected components by size. As observed with many other social networks, there is a giant connected component that includes 99.9995% of the graph. The second largest component has only 43 nodes.

- Although the clustering coefficient was originally defined for undirected graphs [4], we extend its definition to directed graph as follows. For a vertex v_i, we define its *clustering coefficient* to be

$$C_i = \frac{\sum_{v_j \in N_i, v_k \in N_i, v_j \neq v_k} 1\{(v_j, v_k) \in E\}}{|N_i||N_i - 1|}$$

If N_i is the set of outgoing neighbors, i.e. followers, of v_i, then C_i is defined as the *outgoing clustering coefficient* of v_i. Equivalently, if N_i is the set of incoming neighbors, i.e. followees of v_i, then C_i is defined as the *incoming clustering coefficient* of v_i.

Now, using these definitions, in Figure 3 we plot incoming and outgoing clustering coefficients as a function of vertex in-degree and out-degree, respectively, up to 1000. The average incoming clustering coefficient is 0.12 and the average outgoing clustering coefficient is 0.05. The incoming coefficient curve has a number of "dips" at the same exact points that incoming degree distribution has spikes. Granted that the spikes in in-degree distribution are due to the follow limits, it is natural that as people force their number of followees to reach the limit, fewer of these people will follow each other among themselves. On the other hand, observe that the outgoing clustering coefficient has a sine wave shape in the beginning. We believe this is due to spam on Twitter: users with low out-degree are typically users with low activity or new users, thus they have few friends who follow them and the percentage of spam users that follow them is higher than average.

Stream: We have collected 33% sample of all publicly available tweets from mid June through the end of October of this year with up to 5 million tweets per day. To simplify processing, we focused only on the 10-day period from August 1 until August 10. From this 10-day period with the total of 45 million tweets, we extracted all of the URL and tag mentions. We have found 315,000 distinct tags and 9.5 million distinct URLs after translating short URLs to long URLs. We proceeded by picking a sample of 20 most popular URLs and 15 most popular tags with a distinct spike in mentions in the middle of Aug. 1 - Aug. 10 and no or little activity in the beginning and end of this time interval. All of the analysis in what follows is based on these 20 URLs and 15 tags with 1000 mentions per each, on average.

Synthetic Models: Our main objective is to study the interplay between the external and network influences in information propagation on Twitter. Hence, to put our results in perspective, we would like to have some basis for comparison. For this, we developed two primitive models of information propagation that represent two extremes: one where only external sources contribute to information propagation and one where all the contribution is due to the network.

- We will call the first one `Random` , where we produce an infection, by picking at each step a node uniformly at random from the network, as long as it is different from the previously picked ones.

- We will call the second one `RC` (for "Random Cascade"), where we pick the first node uniformly at random from the network, and generate the subsequent ones by picking at random a neighbor of already infected nodes (at step 2, there is one such node, at step 3, there are two such nodes, etc.), again as long as it is different from the previously picked nodes.

For each of these two models, we generate 10 infections with 1200 nodes each and, when reporting results, use the average of these 10 generated infections.

4. EXTERNAL INFLUENCE

4.1 External Influence Metric

If there is a major event that happens externally to the network, e.g. death of Michael Jackson or Barack Obama winning the Noble prize, memes (URLs and tags) about it may spread through the network quite sporadically and independently of the edges between the nodes that mention them. For example, consider a sample network in Figure 4. Suppose node o mentions some meme (i.e. *gets infected*), followed by a, and finally followed by i. Clearly, a could not have gotten this meme from o, as there is no directed path between them. On the other hand, assuming that information cannot flow via more than one edge at a time, i could not have gotten this meme from a or o either. The shortest path to a from the "infected" nodes is $\{i, h, g, d, a\}$ consisting of 4 edges.

In general, whenever a newly infected node x has no edge from any of the currently infected nodes, we can assume x could not have possibly gotten infected through the network. Now suppose that the social network distance between any two nodes is proportional to the social distance between two people in real life. Then, intuitively, some major event diffused by many external sources is likely to spread over larger

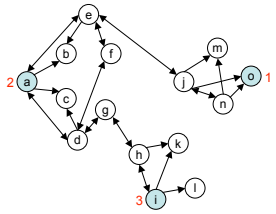

Figure 4: External influence example

distances within the social network than some minor event (event local to some community). Accordingly, to measure the influence of external sources on meme propagation we would like to have some notion of network distance traversed by the infection to reach all currently infected nodes.

For any given infection $\langle x, \langle v_1, v_2, \ldots, v_n \rangle\rangle$, as defined in Section 2, to measure the contribution of external sources to its propagation by each step i, we develop the following metric, intuitively equal to the minimum distance traversed by the infection in the network:

$$\theta_i = \sum_{t=2}^{i} \min_{1 \le k < t} \{d(v_t, v_k), d_{max} + \epsilon\}, \text{ where}$$

- ϵ is some number > 0
- d_{max} is the longest distance in the network between any two connected nodes

So, using our example from Figure 4, this metric at the third step is equal to $(8 + \epsilon) + 4$.

4.2 External Influence Algorithm

The naive implementation of the external influence metric described in the previous section would require running $\frac{n(n-1)}{2}$ BFS's for an infection with n nodes. On large graphs with more than 10^7 nodes and over 10^9 edges, this would be almost impossible to compute for any large infection (more than 1000 nodes). In this section, we present an efficient algorithm we developed that on average reduces computation time from hours to milliseconds on the network of our size.

A simple unidirectional implementation of our algorithm is shown in Algorithm 1. *Multi-source BFS* effectively performs BFS from multiple sources simultaneously with a total of $n-1$ runs required to computer the external influence metric for an infection of n nodes (as opposed to $O(n^2)$ BFS's with the naive implementation). We not only reduce by an order of magnitude the number of searches but also significantly speed up each search individually. The implementation we use in practice has three more optimizations: 1) it is bidirectional, thus explores nodes from the sources and the target at the same time, 2) it chooses at each step the shortest frontier (set of nodes with the highest priority in a particular direction), 3) it caches found distances to nodes between multiple calls to the function. All of these optimizations dramatically reduce computation time.

4.3 External Influence on Twitter

Using the algorithm described, we calculated the external influence metric θ at each step for all of the URLs and tags. As different URLs and tags had different number of mentions, we performed the following normalization on the data.

Normalization: For each mention i of a meme (URL or tag), instead of taking the total θ_i value, we consider the change (increment) in θ_i: $\Delta\theta_i = \theta_i - \theta_{i-1}$. Then for each

Algorithm 1 `multiSourceBFS`(S, t, G)

```
queue ← priorityQueue.new()
for s ∈ S do
  queue.InsertWithPriority(s,1)
end for
while !queue.isEmpty() do
  d ← queue.getPriority() // current best priority
  v_i ← queue.getNext()
  for all v_j, s.t. (v_i, v_j) ∈ E do
    if v_j = t then
      return d + 1
    else
      queue.InsertWithPriority(v_j, d + 1)
    end if
  end for
end while
```

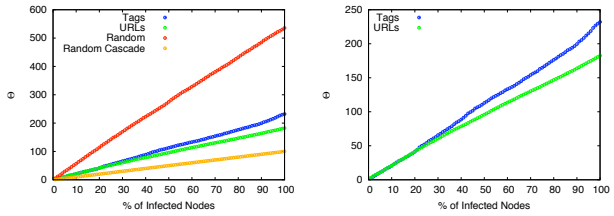

Figure 5: External influence metric θ over the lifetime of an infection a) Average URL and Tag behavior in relation to the `Random` and `RC` b) Average URL and Tag behavior in relation to each other.

meme, we divide the number of mentions into 100 bins, e.g. if a meme has 1200 mentions, each bin contains 12 mentions. For each bin, we take the average of $\Delta\theta_i$ values in that bin, i.e. if $\Delta\theta_1$ is the increment at the first mention, $\Delta\theta_2$ is the increment at the second mention, etc., the average increment of the first bin is $\Delta\hat{\theta}_1 = \frac{\sum_{i=1}^{12} \Delta\theta_i}{12}$. Now along the x-axis, we will always have 100 points/bins, and along the y-axis for each bin i, we will have $\hat{\theta}_i = \sum_{j=1}^{i} \Delta\hat{\theta}_j$. This way, not only we normalize the x-axis to have the same number of points independently of the number of mentions, but also normalize the y-axis not to be biased by the difference in mentions between different memes.

Results: Figure 5 shows θ (y-axis) as a function of % of infected nodes (x-axis) after the normalization described above. In (a), we show URLs and tags behavior, averaged across all URLs and tags, respectively, in relation to the `Random` and `RC` synthetic models; in (b) we compare the behavior of URLs and tags to each other. From (a), one can see that URLs and tags in its behavior are more similar to `RC` which suggests that the network contributes more to the propagation of memes on Twitter than the external sources. Furthermore, when we look at (b), we see that although URLs and tags propagate similarly in the beginning, θ for tags over time increases its slope. This indicates that the tags start traversing larger distances within the network over time and thus are more influenced by the external sources, whereas URLs maintain the same slope traversing smaller distances within the network, thus their propagation can be mostly attributed to the network. In the section that follows, we provide more insight into why tags traverse larger distances.

5. NETWORK INFLUENCE

In this section, we study how the network contributes to the propagation of tags and URLs on Twitter. In other words, we look at how much of meme popularity can be attributed to users copying the meme from the users they follow. Using the terminology introduced in Section 2, we look at 1) how infection graphs and, specifically, cascades (their connected components) look like for URLs and tags; 2) how they evolve as more nodes are infected; 3) what structural properties of edges and nodes can be indicative of cascade formation.

5.1 Structure of Infection Graphs

For each URL and tag, we have constructed its infection graph and looked at all of the cascades. On average, 55% of nodes are cascaded for URLs and 45% of nodes are cascaded for tags. As observed in [2] with recommendation networks and in [3] in blog graphs, we have found that infection graphs frequently have a large connected component. On average, it contains 86% of all cascaded nodes for URLs and 73% of all cascaded nodes for tags.

Next, we attempt to understand how the typical cascades look like for URLs and tags. If cascades are star-shaped, a few central nodes may be responsible for meme propagation on the network, whereas if cascades are chain-like or tree-shaped, nodes participating in the infection contribute to the cascades more evenly.

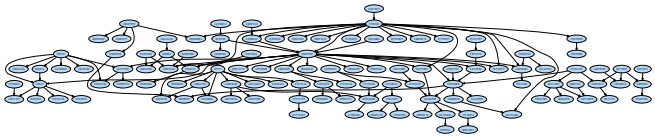

Figure 6: An example of a cascade

Figure 6 illustrates the largest cascade for one of the memes we analyzed. One can observe that overall it has a tree structure with a surprisingly high depth, given the low diameter of social networks. Furthermore, there are a few nodes with high out-degree that might have originated as star-shaped cascades before merging into the main one.

To understand the frequency of high out-degree nodes, we plot in Figure 7 (a) the number of nodes vs. the number of infections they produced averaged across all URLs and tags. The overall distribution follows the power law with a heavy tail. Clearly, the nodes in the tail contribute to the propagation of memes more heavily than the other nodes. Finally, observe how the `RC` closely mimics the dynamics of URLs and Tags. This is not surprising because the number of infected neighbors in `RC` is directly proportional to the degree of the node and inversely proportional to the time of infection, which will naturally generate a power law distribution.

To further understand the shape of cascades, we look at their average depth. Figure 7 (b) shows the depth of a cascade vs. the number of such cascades. Except for the bump at the end, the distribution again follows the power law. The bump is due to the largest cascade for each meme which, as we have observed in Figure 6, is generally quite deep.

Although there is almost always a prominent largest cascade, we have looked into how big are the rest of the cascades. Figure 8 shows the cascade size expressed as a % of the largest cascade vs. the number of nodes in such cascades. We specifically exclude the largest cascade from the figure.

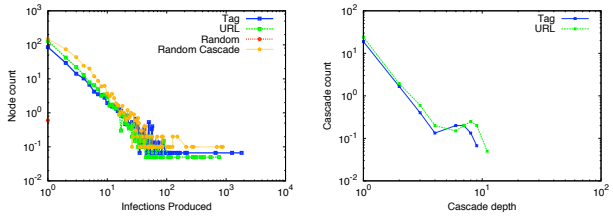

Figure 7: a) Number of infected neighbors vs. number of such nodes b) Cascade depth vs. number of such cascades.

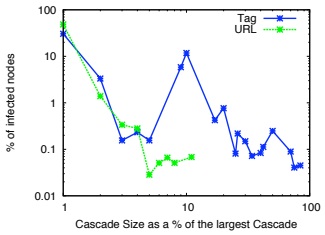

Figure 8: Cascade size vs. number of % of nodes in such cascades (largest cascade excluded)

What we have found is that some of the tags do not have a prominent largest cascade, but rather have many medium-sized cascades. Moreover, even when there is a prominent largest cascade, there are still quite a few medium-size cascades. As a result, there is a lot of weight under the left side of the Tags curve, whereas cascade size distribution for URLs, as expected, follows the power law with the second largest cascade being only 10% of the largest one.

5.2 How Cascades Evolve with Time

Having observed that tags exhibit more external influence and tend to have larger cascades, we would like to understand the cascade growth dynamics over time. First, we would like to verify that the higher external influence observed for tags is due to the emersion of disjoint larger cascades. Second, we would like to confirm that the growth of the largest cascade is generally due to the absorption of smaller cascades, as suggested in [2].

In Figure 9, we plot the number of cascaded nodes, size of the largest cascade, and the number merged cascades within the largest one as a function of the % of infected nodes for both URLs and tags. Observe the steep slope of both cascaded nodes and the largest cascade curves in the beginning of a meme lifetime. This indicates that the cascading behavior starts strongly and quickly forms the largest cascade. The same can be confirmed by the initial dip in the number of merged cascades within the largest component.

The second observation we make is that the growth of the largest cascade is much more rapid for tags than for URLs. This can be seen better in Figure 10 (a), which shows the growth of the largest cascade as function of the % of infected nodes. Furthermore, whereas the largest cascade has more or less steady growth for URLs, its growth actually slows down over time for tags.

Now that we know the general tendency in the growth of the largest cascade, we look into how the growth is achieved over time. Figure 10 (b) shows the number of merged cascades (as a % of the largest cascade size) within the largest one over time (as more nodes are infected). We note the

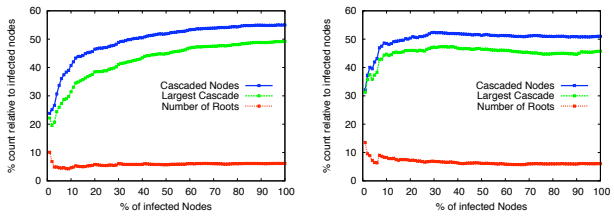

Figure 9: Cascade dynamics over time a) URLs b) Tags

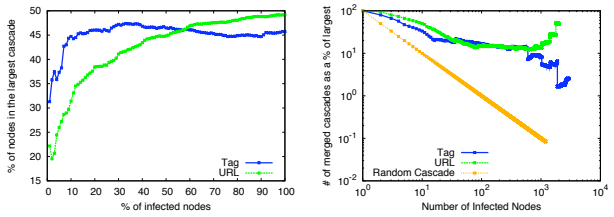

Figure 10: a) Growth of largest cascade over time b) Number of merged cascades within the largest cascade over time

initial decline in the % of merged cascades indicating that the initial growth of the largest cascade is mostly internal. In the middle, however, for both URLs and Tags we observe the ratio staying quite steady suggesting that the largest cascade starts absorbing many smaller ones. Finally, at the end of the infection lifetime, we observe two different trends for URLs and tags. Tags have multiple abrupt drops, whereas URLs have a few sharp spikes. This suggests that the largest cascade for tags either exhibits a lot of internal growth or more likely (based on the previous results) absorbs many larger cascades. In contrast, the largest cascade for URLs grows by absorbing many more small ones.

5.3 Which nodes tend to cascade

In this section, we look into which nodes are more likely to cascade. If one was to do a prediction on whether and how a meme is going to propagate through the network, it is important to understand what nodes tend to cascade and thus which parts of the network the infection is likely to reach.

First, we plot the adoption curve in Figure 11 (a), i.e. the probability of adoption/infection vs. the % of incoming infected neighbors. Here, we exclude the data points with few number of occurrences: for a given % of infected friends, if the number of such observed cases was less than 4000 we ignored it. With generally low probability of adoption, using too few cases may yield statistically insignificant conclusions. After such filtering, we can observe how the probability of adoption for tags generally increases with the % of infected neighbors. On the other hand, for URLs even though there is a general upward tendency, the results are inconclusive: the general trend is similar to the random behavior of RC . Our intuition for why the chances of infection increase with the number of infected neighbors for tags and not for URLs is that URLs are mostly used to share information, whereas tags are often used as a sign of solidarity. Peer pressure will only increase the chances of agreement for tags, whereas for URLs, redundancy in information will not increase the chances of adoption.

The next question we try to answer is whether strong links

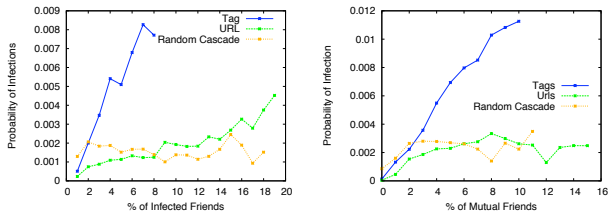

Figure 11: a) Adoption Curve b) Probability of adoption vs. % of mutual friends

or weak links tend to form cascades. Figure 11 (b) shows the probability of adoption/infection vs. the % of mutual neighbors. Specifically, if x is infected and y is its outgoing neighbor, we calculate the percentage of common outgoing edges for x and y. We used the same filtering criteria to ensure that our results are statistically significant. As seen from the figure, for tags, the more mutual neighbors y has with x, the more likely y is to get infected, i.e. strong links are more likely to propagate tags. On the other hand, for URLs the results are again inconclusive. The curve behaves in the same manner as RC , which suggests that strong links are just as likely to propagate as weak links. Intuitively, with tags, one is more likely to agree with somebody who is a close friend (possibly with many shared interests), whereas, with URLs, we are just as likely to share information obtained from close friends as the one from distant friends.

6. CONCLUSION

In this paper we looked at tag and URL propagation on Twitter social network. After developing a new metric to measure the external influence, we have found that external sources have a significant contribution to URL and tag propagation. As the tag becomes popular on Twitter, it travels to more distant parts of the network and forms relatively large cascades that over time start merging into the main giant cascade. On the other hand, URLs travel shorter distances in the network and tend to spread mostly by forming many small cascades that merge into the giant cascade. Finally, we found the that the probability of adoption of a tag is directly related to the number of neighbors who already adopted and the strength of ties with them. On the other hand, for URLs, such tendencies do not hold. Among the future directions we would like to validate the dynamics we have observed on a larger set of tags and URLs and try to develop a model that would predict future propagation of URLs or tags at an early stage.

7. REFERENCES

[1] R. Anderson and R. May. *Infectious Diseases of Humans: Dynamics and Controls*. Oxford University Press, 2002.

[2] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), 2007.

[3] J. Leskovec, M. Mcglohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *In SDM*, 2007.

[4] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.