Reconstruction and Analysis of Twitter Conversation Graphs

Peter Cogan†, Matthew Andrews‡, Milan Bradonjic‡
W.Sean Kennedy‡, Alessandra Sala†, Gabriel Tucci‡
†Alcatel-Lucent, Bell Laboratories, Ireland.
‡Alcatel-Lucent, Bell Laboratories, NJ, USA.
{peter.cogan,milan.bradonjic,sean.kennedy}@alcatel-lucent.com
andrews@research.bell-labs.com
{alessandra.sala,gabriel.tucci}@alcatel-lucent.com

ABSTRACT

User interactions over social networks has been an emergent theme over the last several years. In contrast to previous work we focus on characterizing user communications patterns around an initial post, or conversation root. Specifically, we focus on how other users respond to these roots and how the complete conversation initiated by this root evolves over time. For this purpose we focus our investigation on Twitter, the biggest micro-blogging social network. To the best of our knowledge this is the first such method that is able to reconstruct complete conversations around initial tweets. We propose a robust approach for reconstructing complete conversations and compare the resulting graph structures against those obtained from previous crawling strategies based on keyword searches. Our crawl provides a large scale dataset, ideal for computer scientists to run large scale experimental evaluations, however our dataset is made of a collection of small scale, highly controlled and complete conversation graphs ideal for a sociological investigation. We believe our work will provide the proper dataset to establish concrete collaborations with interdisciplinary expertise.

Categories and Subject Descriptors

E.1 [**Data Structures**]: Graphs and Networks; H.3.5 [**Information Storage and Retrieval**]: Online Information Services - Data sharing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval Information filtering

ACM HotSocial'12 August 2012 Beijing, China

Keywords

Social Network, Graph Mining, Online Interactions

1. INTRODUCTION

Communication over social networks has become an emergent research theme over the last several years followed by a large body of literature and several major conferences. We aim to understand how online users communicate around specfic topics, which may provide a novel perspective in mining online user behavior and modeling the information spread in social networks.

This work provides the first investigation on methods for constructing *complete conversations* around initial posts. Performing such a study requires data collection from social networks which presents a number of challenges. Gathering data from large social networks such as Facebook [1] and Twitter [2] has become harder than it used to be several years ago due to external pressure to improve privacy and internal pressure to generate revenue. For example: (i) Facebook has significantly improved user privacy awareness and control, which has led to most Facebook users keeping their data private; (ii) Twitter employs third parties to help monetize the release of data which can be a hurdle to research. However, both Twitter and Facebook still provide APIs to support third party applications, and these are also used by researchers to gather data from those networks. (iii) APIs are rate limited which implies slow crawling time.

Our current focus is on the social networking site Twitter due to its popularity, general open nature, and availability of limited APIs for extracting data. Users write short posts, or tweets, containing various content such as current activities, personal messaging, and shared links for interesting media. Once a user has posted a tweet, all her followers will receive the tweet. Tweets posted by a public profile user can be viewed online by anyone. Whenever a user receives a tweet, she may retweet the tweet (send it on to their own set of followers), or alternatively may reply to the tweet

(respond to it by using her own original tweet). Our focus is to construct and characterize conversation graphs that represent how users interact within the network.

In this work we present and compare two data collection methods. The first method, widely used in the community, utilizes keyword filtering which allows us to retrieve tweets matching particular keywords, i.e. it finds all tweets which mention the keyword "Obama". Results using a filter query to generate a conversation graph are demonstrated in Section 4. The second method is motivated by the fact that keyword filtering cannot recreate complete individual conversations since tweets within a single conversation may not all match a single keyword. Therefore, we design a novel and more elaborate approach, described in detail in Section 3, that recursively uses the Twitter Search APIs. This method first selects a random reply and then traces the chain of replies back to the root (i.e. the initial tweet) of the conversation. Once the root has been found all of the tweets within the conversation can be tracked.

Since we have the capability of reconstructing complete conversation around initial Tweets, we then explore how to characterize those graphs. There are a number of ways in which a conversation graph can be defined. We will consider the following types of structures:

1. **Mention graphs:** The first type is a graph where vertices are users and (directed) edges represent any interaction between the users. Such interactions include replies to each other's tweets, retweets, mentions (a mention is when a user refers to another user directly by username), etc.

2. **Reply trees:** The second type is a graph where vertices are tweets and a (directed) edge represents one tweet that is a reply to another. In this case a tweet can only reply to one other tweet, thus the connected components of such graphs are trees.

3. **User graphs:** The third type can be viewed as a projection of the second type. In this case vertices are users and a (directed) edge exists if one user replies to another.

The main difference between the first type and the second and third types is that the first type includes retweets and mentions in the definition of an edge whereas the second and third types only include replies in the definition. One reason to examine a graph with replies only is that intuitively a reply represents the most direct communication sign between two users. Hence, by studying graphs formed by replies we can gain insights which allow us to characterize users' communication patterns in social networks.

In this paper we focus on two main question. First, *how do we efficiently reconstruct conversation graphs without violating the limits that are imposed by the Twitter APIs?* Second, *how can we characterize the structure of the resulting conversation graphs?* Results from the two methods are described in Section 4. We find two extremely common types of conversation graphs, namely paths and stars. Paths typically correspond to two users having a back-and-forth conversation. Stars correspond to multiple users replying to a single popular tweet. However, there are a number of conversations that fall between these two extremes and we investigate the structures that are formed.

2. PREVIOUS WORK

Conversation graphs form a dynamic representation of how information is propagated throughout the social network over time. To the best of our knowledge, there has not been previous work on the structure of reply-based conversation graphs or on methods to collect these graphs without violating Twitter API limits.

However, the general topic of information propagation in social networks has been addressed previously. For example, in [8] the authors proposed a simple mathematical model that generates basic conversation structures and takes into account the identities of each member of the conversation in Usenet, Yahoo! Groups, and Twitter. Moreover, some recent studies looked at how information flow depends on the structural properties of user interactions [4, 9]. Concretely, social cascades (the information spread in social networks) in Flickr are the focus of [4]. The pictures in Flickr represent the information unit around which the interactions happen. Becoming a fan of a particular picture is interpreted as information flowing from one user to another. The 98% of the social cascades generated from pictures with less than five fans happen within the one hop neighborhood of the picture uploaders. For popular pictures, instead, more than 50% of their fans are outside the uploader's one hop neighborhood. This analysis suggested that the main feature leading to a broader information spread is the popularity of a picture. On the other hand, the study of [9] provides a preliminary analysis of the topological structure of retweets. The forest of retweet trees has a large number of one or two-hop chains. The majority of retweet trees have a height smaller than six, and no trees beyond eleven hops were found.

Moreover, modeling the interaction among users plays a fundamental role in understanding how the information is disseminated in the network [13, 6] and how to maximize its spread [7, 5]. In [9], Kwak et al. proposed the first large scale analysis of the topological and temporal structures of retweets to capture popularity of users, trending topics and temporal patterns to describe how the information propagates in the network. The temporal dimension of the information spread plays a significant role both to characterize users' retweets

and to identify influentials [10, 14]. Inspired by sociological and viral marketing studies, Cha et al in [3] propose an in-depth analysis to characterize user influence based on the investigation of several factors that may play an important role in identifying influentials in Twitter. Finally, social cascades have also been used to demonstrate that, differently from static properties, in dynamic interactions the users' geographic locality is a key factor in characterizing how the information spreads in the network [12].

Complementary to the aforementioned, our work is mainly focused on constructing reply-based conversation graphs subject to the API limits and understanding the distribution of their structural properties ranging from one common extreme form (paths) to another (stars). To the best of our knowledge, these topics have not been the focus of prior work.

3. RECONSTRUCTING CONVERSATIONS

In this section we describe two methods for constructing twitter conversation graphs. The first, named *filtered conversation* method, crawls the conversation around a set of predefined keywords. We use the Twitter filter API to collect all tweets containing any one of these keywords over a fixed period of time. We then create a mention graph from this set of tweets. We remark that given a set of tweets any of the three types of conversation graphs can be constructed in linear time. The goal of this method is to understand the structure of the conversation graphs surrounding specific topics, in particular, involving specific keywords. However, as mentioned earlier, this method will not capture tweets that are part of a conversation but do not mention the conversation topic. To address this our second technique, named the *complete conversation* method, aims to reconstruct entire conversations without this restriction that each tweet of a conversation must contain specific keywords. The complexity of reconstructing entire conversations derives from the fact that although a tweet payload contains a large amount of meta-data, it does not include a list of tweets that reply to it. The inclusion of such a list would be impractical since it can be very long, and is of course dynamic. Our complete conversation method therefore requires a number of recursive calls to the Twitter APIs to find these replies. In the following we first describe some of the technical details of the Twitter APIs required to implement our strategy and finally give the details of the algorithm to reconstruct complete conversations.

Twitter APIs Required for the Complete Conversation Algorithm. The REST API allows access to specific tweets, follower lists, retweet lists, etc. Each tweet is assigned a unique unsigned 64 bit integer identifier - this is the ID. Usage of many of the APIs are limited to 350 calls per hour for authenticated applications, thus any conversation collection algorithm which relies on the REST API will be severely limited. In order to mitigate this issue, this algorithm takes advantage of the search API which returns tweets matching a specified keyword. Although this API is also rate-limited, it offers the opportunity to recreate more conversations than would otherwise be possible.

Algorithm 1 Iterative Root Finder

```
1: S_n is the tweet passed as an input
2: k=0
3: repeat
4:    if type(S_{n-k})==Reply then
5:       Call Search API to search for tweets addressed to user
6:       Identify S_{n-k-1} by matching its ID to the field
         in_reply_to_status_id from S_{n-k}
7:    end if
8:    k=k+1
9: until (type(S_{n-k-1})==Tweet)
10: RETURN true
```

3.1 Complete Conversation Algorithm Design

Our algorithm to reconstruct complete conversations has been designed for a parallel environment. First, the streaming API is used to filter tweets with specific keywords which are the subject of interest. The set of tweets retrieved from the filter operation is denoted $\mathbf{D}$. Each tweet D_i in $\mathbf{D}$ is part of a conversation C_i - however it may not be the *root* of conversation C_i. A tweet can be of three types, *root*, *reply* or *retweet*, of which roots and replies are used to create reply trees. Let the type of tweet t be denoted type(t). The goal of the complete conversation algorithm is, given D_i, to determine the root of the conversation C_i, and then to determine the set of all tweets part of C_i. If D_i is not a root tweet, the Iterative Root Finder algorithm is used to determine the root of C_i.

Let $\mathbf{S}$ be a subset of the conversation C_i where S_0 is the root of the conversation C_i and S_n is a single tweet of the conversation retrieved using the filter (i.e. D_i in the above context). $\mathbf{S}$ is defined such that S_{n-k} is a reply to S_{n-k-1} for $k \in \{0, \ldots, |S| - 2\}$, with this pattern repeating to S_0. The goal of the Iterative Root Finder algorithm is to identify S_0 given S_n. Note that when the algorithm starts, $|\mathbf{S}|$ is unknown.

Once the conversation root S_0 has been established, the remainder of C_i can be sought using the Iterative Search algorithm. Let $\mathbf{T}_j$ be the vector of all tweets in C_i at iteration j and $\mathbf{M}_j$ be the vector of all contributors to C_i at iteration k. The k^{th} tweet at iteration j is denoted $T_{j,k}$.

Since new tweets and new conversation members are continuously added, the Iterative Search algorithm is run repeatedly until exit conditions, e.g. the conversation has ended, are met.

One of the key components of this algorithm is the ability to find replies to tweets. The Twitter REST API

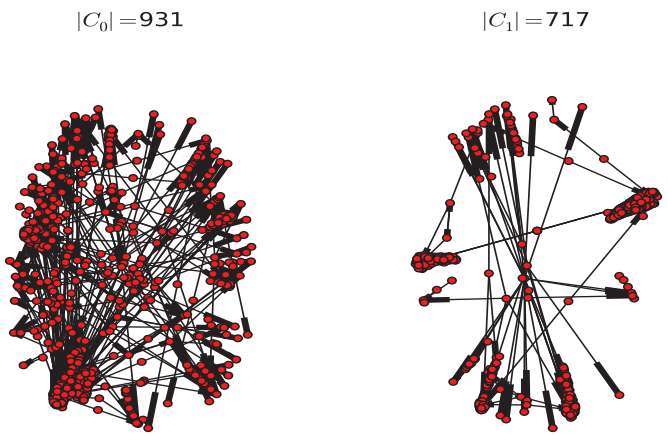


Figure 3: The largest 2 connected components for the mention graph created by filtering on the keyword "flu"

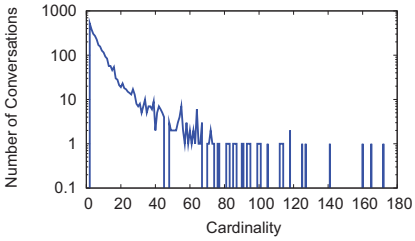


Figure 4: Cardinality

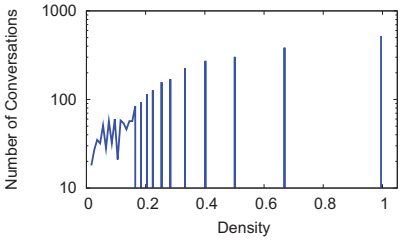


Figure 5: Density

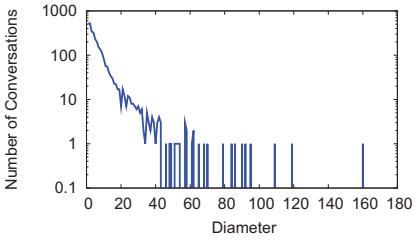


Figure 6: Diameter

Algorithm 2 Iterative Search

1: i=1
2: **repeat**
3: **for** $j = 1$ to $j \leq |\mathbf{T}_i|$ **do**
4: Search for tweets addressed to author of $\mathbf{T}_{ij}$
5: Extract replies to $\mathbf{T}_{ij}$ by matching field in_reply_to_status_id - add tweets to P. Add new members to Q
6: **end for**
7: Assign $\mathbf{T}_{i+1} = (\mathbf{T}_i\ P)$
8: Assign $\mathbf{M}_{i+1} = (\mathbf{M}_i\ Q)$
9: i=i+1
10: **until** $(—\mathbf{T}—_{i-1} == —\mathbf{T}—_i\ \&\&\ —\mathbf{M}—_{i-1} == —\mathbf{M}—_i)$

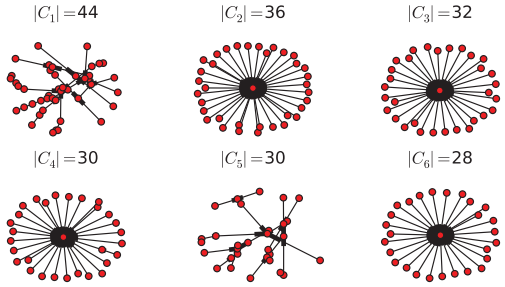


Figure 1: The largest 6 connected components for a reply tree created by filtering on the keyword "GOP"

does not provide a mechanism to directly find replies, thus we use the search API to find replies to users. This is possible because a reply to a user will always begin with @username, thus a search call will return all the replies to that user. In order to determine whether the returned tweet is an element of the conversation, we can check the in_reply_to_status_id field. This field was only added to the search output in December 2011 which made this investigation possible. Note that the search API returns up to a maximum of 1500 tweets from the previous week. Thus we cannot track old conversations, and we cannot track conversations whose members receive huge numbers of replies. In order to limit the number of search API calls we also employ the since_id field which ensures we do not search for tweets before the root. The algorithm was tested by injecting multi-user test conversations with known structure into Twitter.

Conversation Lifetime. Our analysis has shown that the vast majority of conversations are not continued if the oldest tweet in the conversation is more than

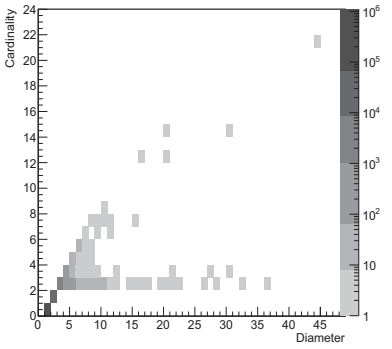


Figure 2: Heat map comparing diameter to cardinality for the connected components for a reply tree created by filtering on the keyword "GOP"

6 hours old. Thus in our algorithm if the oldest tweet in a conversation is more than 6 hours old we stop tracking it.[a]

Outlier Cases. There are some circumstances under which we are unable to correctly recreate a reply tree as it originally occurred. For example, the user has the option of deleting a tweet, or a user with a public profile could write a tweet in reply to a user with a private profile. Less than 5% of conversations are affected this way.

Handling Rate Limits. Twitter applies limits to the usage of the APIs which ensure Twitter servers do not get overloaded with DOS attacks, runaway scripts, etc. Our algorithm handles these limits by inserting delays when the search API limit is reached. The frequency with which the search API can be called is not publicly available. When the search rate limit is hit, the search API can not be used for a period of time. This period of time is provided to the user via the HTTP header and is typically close to 8 minutes.

Hardware Resources. The algorithm was run on a single multi-core machine running OSX 10.7.2. Processor speed is largely irrelevant due to the large latency invoked by calling the search API, plus the rate limiting delays which must be inserted. The repeated calls to the search API result in the storage of a large number of tweets. During our collection period approximately 2 million tweets were collected which must be stored in memory as they could be added to conversations during any iteration. For this work, the data structures stored all the tweet meta information in memory, however most of this is redundant and could be backed up to disk while a leaner data structure is kept in memory.

[a] It is hard to determine that a conversation will not be replied at some indeterminate time in the future. However, the algorithm has to eventually stop following conversations since there are practical limits to how many search calls can be made while still aggregating active conversations.

4. EXPERIMENTAL ANAYSIS

Crawled Data. In this section we present the datasets obtained with the two proposed methods. The filtered conversation method was seeded with numerous keywords. In this report we focus on the keywords "GOP" and "flu". These two topics alone resulted in the collection of over two million tweets, over a crawling period of a month from January 11^{th} 2012 to February 11^{th} 2012. In contrast, using the complete conversation algorithm, we collected a total of 3114 conversations containing 33K tweets from February 1^{st} 2012 to February 5^{th} 2012.

4.1 Topology of Conversation Graphs

Keyword Based Graphs. We begin with an examination of graphs formed by the filtered conversation method. Figure 1 shows the 6 largest connected components of the *reply tree* constructed from the collection of tweets containing the keyword "GOP". We find a large number of connected components, i.e. disjoint reply trees, indicating numerous short conversations. Figure 2 plots the diameter as a function of the number of vertices for each connected component. This reveals a large number of components C whose diameter is nearly $|V(C)| - 1$; clearly, a graph H has diameter $|V(H)| - 1$ precisely when it is a path. The other extreme is a star whose diameter is 2 (for instance many of the graphs plotted in Figure 1).

In contrast, the structure of the *mention graphs* constructed from the same set of tweets is very different. For example, the largest connected component for the mention graph constructed from tweets containing the keyword "GOP", only on January 21^{st} 2012, has 18132 nodes and 32815 edges; clearly, this component is not a tree. Furthermore, the next largest connected component has cardinality 14. Less popular topics create smaller scale examples which are easier to visualize. For instance, Figure 3 shows the 4 largest connected components of a mention graph constructed from tweets containing the keyword "flu", again on January 21^{st} 2012. We note that unlike the giant component of the "GOP" mention graph, the "flu" mention graph has no single component whose cardinality dominates the remaining components. Interestingly, there are still a large number of components which are star-like, though, the largest two components have a much richer geometric structure.

Complete Conversations Graphs. We now report on the structures obtained by the complete conversation algorithm. Using this method we collected a total of 3114 conversations containing 33K tweets from February 1^{st} 2012 to February 5^{th} 2012 and constructed the corresponding reply trees and user graphs.

In Figures 4 and 6, we show the distributions of the

cardinality and the diameter for these graphs. In particular, the number of graphs with cardinality between 1 and 40 decreases exponentially. The same phenomenon can be observed for the diameter plot in Figure 6. Furthermore, we show the scatter plot of the cardinality versus diameter, in Figure 8. This immediately reveals a large number of conversations whose diameter is exactly one less than its cardinality, i.e paths. Paths are generated by back-and-forth conversations between two users. We also analyze the density of these graphs, as shown in Figure 5. The plot is dominated by paths wherein $|E| = |V| - 1$ where E is the set of edges and V is the set of vertices. Thus the peaks are at $2/|V|$ with the height of the peak determined by the number of such paths. Note that while the number of graphs with cardinality more than 40 drops dramatically, the number of graphs with density values ranging from 0.2 to 1 remains similar.

Paths make up more than 60% of the reply trees which means that this structure largely dominated the set of conversation graphs. We provide a visual representation, in Figure 7(a), of a large cardinality path plotted as its reply chain vs its user graph. The topology structures, extracted from the entire collection of our graphs, are often very far from paths. There are also conversation graphs whose diameter is 2, some having very large cardinality. These graphs correspond to instances whereby a single user generates a tweet to which a large number of people reply – however the users do not respond to each other's replies. When plotted, such a graph looks like a star as in Figure 7(b). Since each tweet is from a unique user, the corresponding user graph would look identical.

Paths and stars can be considered the two extremes of the possible conversation graph structures. Graphs in between these extremes reveal richer geometric structures and possibly more interesting sociological user interactions insights. Figure 7(c) shows an example of a more complex reply tree and in particular its extra complexity compared to its user graph Figure 7(d). We believe that by exploring the dynamic features of reply graphs, i.e. how they form and grow, we may catch fundamental insights to understand the underlying processes that govern user interactions.

5. CONCLUSIONS AND FUTURE WORK

The method for converation reconstruction described in this work enables a comparison of social interaction models to online social networks. For example, this work has unconvered the existence of two dominant conversation topologies. By presenting these results at this interdisciplinary workshop we hope to determine whether such topologies are representative of the current theories of social interaction. Do these topologies fit well with or challenge these models? Furthermore, by

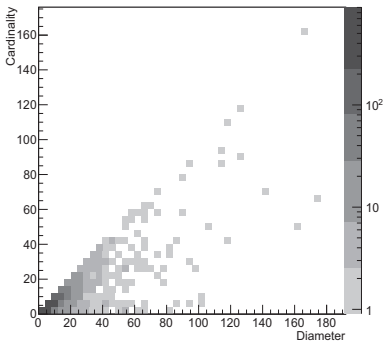


Figure 8: Heat map comparing diameter to cardinality

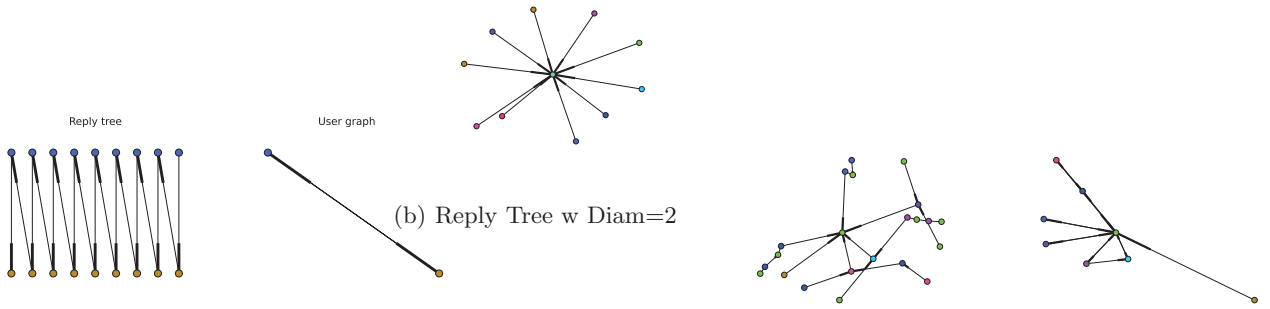
leveraging sociological domain expertise, we wish to determine whether extra information could be mined from twitter to aid interpretation of these discrete conversations in the context of these social interaction models.

One natural feature of any conversation graph is the fact that it evolves over time. In future work we would like to extend our studies of the structural and spatial aspects of the conversation graphs and address their temporal structure. One basic question here is what is the distribution of conversation length. As already mentioned for the vast majority of the reply trees that we examined there was no activity after 6 hours.

Another interesting topic to study is how fast a conversation spreads in both a temporal and spatial sense. As before, a conversation starts when one or several individuals tweet about a specific topic. For a temporal analysis let us assume that we count the number of retweets or replies in the conversation as a function of time. More specifically, let $F(t)$ be the number of retweets or replies up to time t and let $f(t)$ be the number of tweets between time $t-1$ and t where the granularity could be 10 minutes, 30 minutes or an hour. What is the typical behavior of the previous functions? At what time does the maximum of f occur? In other words, after how long does the conversation become most active? How does this depend on the topic of conversation? How often does it occur that the function f has several local maxima? If so, what is the reason for these local maxima?

Other questions concern the relationship between conversation graph dynamics and geographic location. How are these dynamics affected by the geographic dispersion of the users? Are conversations likely to grow more rapidly if the users are themselves physically close together or does physical proximity make no difference?

Lastly we would like to use our studies of conversation graph structure as a predictive tool. In [11], Liben-Nowell and Kleinberg studied the general problem of link prediction in social networks. We are interested in the most effective way to do link prediction in conversa-



(a) Back-and-forth conversation represented by a path.

(b) Reply Tree w Diam=2

(c) Irregular Reply tree

(d) Irregular User Graph

Figure 7: User graphs and reply trees - see text for details

tion graphs. In particular, can we use our classification of conversation graph types and past history of users to predict whether or not a user will join a conversation? Moreover, can we predict how they will join the conversation? For example, are they more likely to join by replying to the root node or by replying to one of the more recent tweets that is a current leaf of the conversation?

6. ACKNOWLEDGEMENTS

The work of Andrews, Bradonjić, Kennedy and Tucci was partially supported by NIST Grant No. 60NANB10D128. The work of Kennedy was partially supported by NSERC.

7. REFERENCES

[1] `http://facebook.com/`.
[2] `http://twitter.com/`.
[3] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proceedings of international AAAI Conference on Weblogs and Social*, ICWSM '10.
[4] M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proc. of WWW*, 2009.
[5] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proc. of KDD*, 2009.
[6] V. Gómez, H. J. Kappen, and A. Kaltenbrunner. Modeling the structure and evolution of discussion cascades. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, HT '11, pages 181–190, New York, NY, USA, 2011. ACM.
[7] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc. of KDD*, 2003.
[8] R. Kumar, M. Mahdian, and M. McGlohon. Dynamics of conversations. In *Proc. of KDD*, pages 553–562, New York, NY, USA, 2010. ACM.
[9] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proc. of WWW*, 2010.
[10] C. Lee, H. Kwak, H. Park, and S. Moon. Finding influentials based on the temporal order of information adoption in twitter. In *Proc. of WWW*, 2010.
[11] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management*, pages 556–559, 2003.
[12] S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft. Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades. In *Proc. of WWW*, 2011.
[13] D. Wang, Z. Wen, H. Tong, C.-Y. Lin, C. Song, and A.-L. Barabási. Information spreading in context. In *Proc. of WWW*, pages 735–744, New York, NY, USA, 2011. ACM.
[14] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, 2011.