RApID: A System for Real-time Analysis of Information Diffusion in Twitter

Io Taxidou, Peter M. Fischer
University of Freiburg, Germany
{taxidou,peter.fischer} @informatik.uni-freiburg.de

ABSTRACT

The advent of social media has facilitated the study of information diffusion, expressing information spreading and influence among users on social graphs. In this demo paper, we present a system for real-time analysis of information diffusion on Twitter; it constructs the so-called *information cascades* that capture how information is being propagated from user to user. We face the challenge of managing and presenting large and fast-evolving graph data. For this purpose, we have developed methods for computing and visualizing information flow dynamically, offering rich structural and temporal information. The interface offers the possibility to interact with the dynamic, evolving cascades and gives valuable insights in terms of how information propagates on real-time and how users are influenced from each other.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: General; H.5.2 [**Information Interfaces and Representation**]: User Interfaces

Keywords

Information diffusion; Information cascades; Visualizations

1. INTRODUCTION

Social media provides rich means of interactions among people in which they create, share, and exchange information. The advent of social platforms and the constant engagement of users on them provides a treasure of information for analysis. A crucial area that has recently gained attention in the data management community is the study of information diffusion, i.e. tracing, understanding and predicting how a piece of information is spreading in social networks [3]. Studying information flow yields valuable insights in the interaction patterns of users and provides opportunities to identify user roles like e.g., opinion leaders or spammers. These interactions are represented as *information cascades* which are temporal sub-graphs of the underlying social graph. They provide a model of information diffusion, where the nodes correspond to users and edges reveal "who was influenced by whom". Real-time analysis of information diffusion is of particular interest for many, currently not well-supported use cases: Online journalists need to understand timely how to evaluate the sources of their information, while celebrities and politicians need to react quickly to public opinion, rumor spreading and the "echo" on their own publications.

A valuable source for studying information diffusion is Twitter which maintains a social graph of *friends* and *followers*. The act of *retweeting*, which is forwarding another user's tweet by giving credit to it, facilitates explicit information diffusion from user to user. In many countries, Twitter provides a relevant and representative coverage of the population, making it a *social sensor*. Finally, the fact that messages and the social graph is open to some extent makes such analysis viable in practice.

In this demo, we present a system that reconstructs and visualizes information cascades in Twitter with the following contributions: (1) We provide an infrastructure for our algorithm [4] that reconstructs information cascades on real-time. (2) We develop methods to visualize large cascades with collapsing techniques in order to understand their structure and properties. (3) We visualize temporal and geographical distribution of the cascades, as they grow in time.

There is only limited work on this kind of problem: reconstructing large information cascades on real-time is not a trivial task under data limitations and lack of observable diffusion paths [4]. Additionally, visualizing cascades of thousands of nodes in a dynamic way, that is assigning edges of user interactions while they are happening, is an equally challenging problem. Closest to our work is an online application called "Whisper" [2] which expresses the diffusion process in a coarse-grained manner. It contains temporal and spatial characteristics with lightweight sentiment analysis, but does not provide the accuary and level of detail that we aim for.

In Section 2 we provide an overview of the cascade reconstruction algorithm, followed by the techniques for visualizing large cascades in Section 3. We describe the components of the system infrastructure in Section 4 and provide a short demo tour in Section 5. Lastly, Section 6 concludes and gives future directions.

2. ALGORITHM AND MODELS

In order to facilitate the understanding of challenges for computing information cascades, we briefly discuss the algorithm presented in [4]. Since diffusion paths are almost never explicitly available, the algorithm takes the occurrence of specific information propagated in messages as well as the social graph connections into account. Based on this information, it derives possible likely influencers and thus the diffusion paths. For the specific case of Twitter retweets, only a reference to the initiator of the retweet cascade is provided, but no information on intermediate influencers.

The core part of this algorithm is the influence assignment mechanism, expressing the models proposed in [1]: In the naive case,

http://dx.doi.org/10.1145/2661829.2661849
.

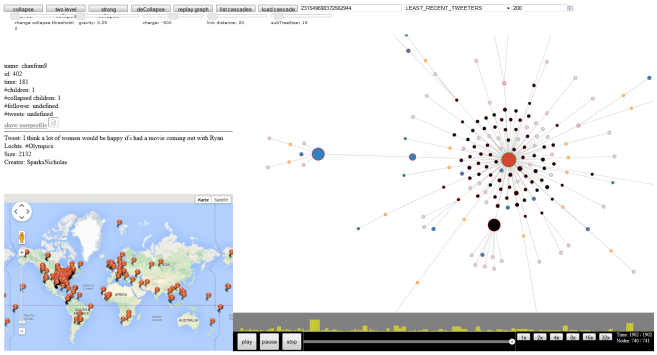


Figure 1: Visualized Information Cascade

when a user retweets a message, we assume that was influenced by all of his friends who (re)tweeted the same message in the past. For that, we need to access all follower data of these predecessor users, which leads to a quadratic complexity with regard to the cascade size. However, identifying all possible edges (influencers) is not needed. The algorithm currently supports four different influence models in order to trim influence edges. We can assume that every user who retweets is influenced by only one of his friends: (1) the one with the most followers or (2) most retweets or (3) the most recent influencer or (4) the least recent influencer. According to the influence model the semantics of influence change which has an impact on the structure of the cascade.

3. VIZUALIZATIONS

The cascade visualization interface is depicted in Figure 1. Our goal is to show the high level structure of the cascades as they grow dynamically on time. The visualization includes nodes and edges only, since any other additional information on the cascades would be obscured for large sized ones. The interface provides four complementary aspects: (1) dynamic representation of the structure of large, evolving information cascades, (2) real-time distributions, like temporal activity and geographical spread (3) relevant information for users participating in the cascade, when the corresponding node is pointed with the mouse, like user name, number of followers and location and (4) information for the original tweet, like text and number of retweets.

While there are many solutions to visualize static graphs, very few solutions exist for dynamic, large-scale graphs. Large cascades can grow up to hundred of thousands of interactions, which obscure the overall structure and dynamic behavior when visualized. The dynamic behavior of cascade graphs demands that updates are integrated instantaneously and properties are recalculated at the same time. To address these issues, we developed our own web-based visualization techniques for dynamic, large-scale cascade graphs which perform adaptive collapsing of node groups. The key idea of collapsing is based on the following observation: There is a skewed distribution in the number of social media users who influence their friends to react further to their messages; the majority of users fail to influence their friends, while very few users trigger many reactions. This creates many "leaf" nodes that stop the information flow, e.g., retweeters that trigger no additional reactions. Combining/collapsing these leaf nodes with their "parent" into a "supernode" provides a first step to significantly reduce the number of visible nodes while not losing relevant graph structure.

Given the complex structure of information cascades, we developed two more aggressive collapsing strategies that gradually shed the structure, but retain most relevant information in the nodes: *Two level collapsing* aggregates the leave nodes of the cascade in their direct parent node iteratively, while the depth of the collapsed sub-tree should not exceed two levels. This method takes as a parameter the maximum number of nodes that can be collapsed in every parent node. The multilevel collapsing works iteratively as the two-level collapsing, without any sub-tree depth restriction. It is parametrized only by the number of possible collapsed nodes.

The size and the color of the nodes carries additional information as illustrated in Figure 2: The size of each node is proportional to the number of the collapsed nodes it carries. The number in the core of nodes shows the depth of the sub-tree that is collapsed. The red node in the middle is the root, while the green outlined nodes are leaves and yellow ones are bridges that connect two sub-graphs. Grey nodes have one child collapsed while blue nodes have more than one child collapsed. Since real-time reconstruction works on possibly incomplete data, nodes with no observable parent are assigned to the root and highlighted in black. Figure 2 shows a cascade of 34.221 nodes without any collapsing, with two level collapsing (1.095 visible nodes) and multilevel collapsing (212 visible nodes). For the last case, the parameter of sub-tree size is set to 5. More aggressive collapsing can be achieved if we increase this parameter. We observe that the high level structure, e.g. the two big components connected by a bridge, is not being lost.

These strategies were implemented on top of the D3 Javascript graph visualization library[1], supporting all major browsers. We can successfully represent cascades with up to 60K nodes. The selected cascade for dynamic reconstruction is streamed using Web Sockets in the form of edges, containing the incoming node and the parent node. The influence model is selected and used accordingly during reconstruction. The nodes that are going to be collapsed according to the collapsing strategy are integrated directly into their parent nodes. Newly visible nodes take dynamically their place and the cascade graph is being reorganized. While the nodes are arriving, their geographic location is pinpointed into a map. At any time during the dynamic reconstruction, the user can decollapse and collapse again nodes by clicking at them or change the collapsing strategy dynamically. Selecting a collapsed node will show the location of all collasped leafs in addition to the parent, so that possible locality of influence can be seen. When the reconstruction is finished, the cascade can be replayed back and forth, stopped and paused while the temporal distribution is depicted.

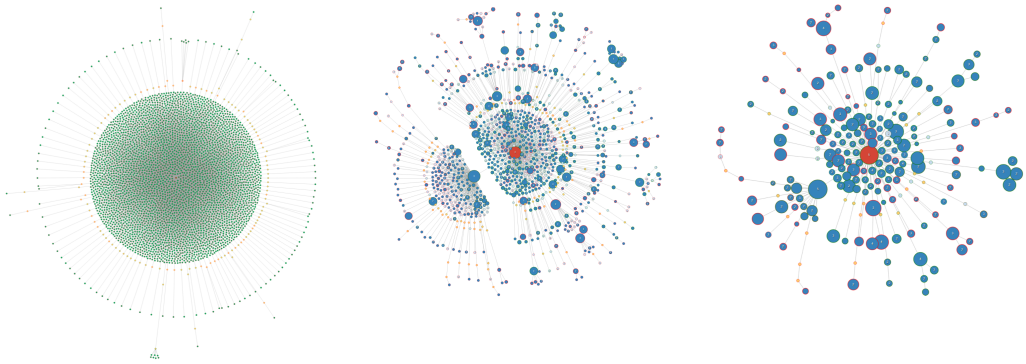
4. SYSTEM

The system we present in this demo is based on the architecture shown in Figure 3. The nature of visualization and reconstruction requires matching dynamic data for messages and the social graph. We therefore need to perform (1) data collection and filtering and (2) historic data storage and retrieval. Based on this data, we can perform (3) computations for streaming cascade reconstruction using the algorithm presented in Section 2. This algorithm is implemented on top of Storm[2], a scalable, distributed data stream processing platform suitable for real-time computations. The results are then streamed to the (4) front-end visualization, which was described in Section 3. We therefore briefly present components (1) and (2) in this Section.

***(1) Input Subscription Stream and Social Graph Manager*:** We used Twitter as a source for messages and social graph information. The input subscription stream manager serves as an interface to the different streams provided by Twitter, like the sample stream that

[1]http://d3js.org

[2]http://storm.incubator.apache.org/



(a) Full Cascade (b) Two level (c) Multi level

Figure 2: Collapsing Techniques

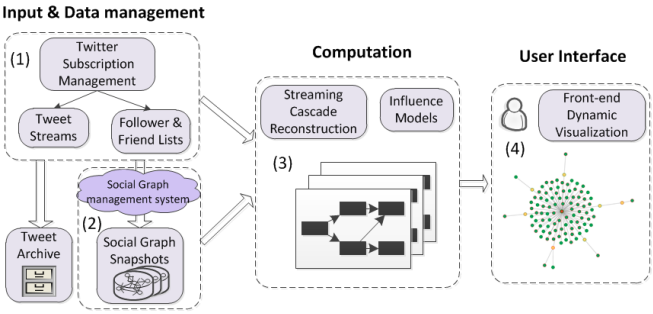


Figure 3: Architecture

is 1% of the public stream or filter subscriptions for terms/hashtags and users. Specific cascades can be addressed by a combination of these means. The subscription management delivers the (filtered) tweet steams to the downstream processing components and stores it into an archive for offline evaluation.

The input subscription social graph manager retrieves the social graph data in the form of followers or friend lists needed for reconstruction. We implemented a crawling facility and we maintain a cache of crawled social graph parts. An updating facility is also used in order to keep the social graph updated and in synchronization with the current cascades. Given its sheer size (100s of millions of users with their connections), we have to judiciously perform retrieval to have access to relevant and up-to-date information based on observed relevant activity.

***(2) Social Graph management and maintenance*:** In order to reconstruct with with low latency, the algorithm presented in Section 2 needs efficient access to the huge social graph. To provide these means, we are working on distributing of the graph which exploits the locality of social interactions. For the purpose of the demo, in which we have a limited amount of users, we pre-load the relevant part of the social graph for reconstruction into main memory and load the remainder on demand.

5. DEMO TOUR

For the demo, we plan to show our cascade reconstruction infrastructure on a single machine, with *offline*-pre-recorded and *online*-live social media streams.

In the offline scenario, we will provide the ability to select pre-recorded cascades, load the corresponding social graph fragment and perform the reconstruction. In the online scenario, we will choose among the currently most active running cascades and trace messages while they are arriving. Since we cannot pre-determine the part of the social graph needed for reconstruction, it will be loaded on demand and access will be slower in this case. We are currently working on distributing computation and social graph storage in such a way that real-time reconstruction is achieved for a full social graph by exploiting interaction locality.

In both cases, we will visualize the reconstructed cascades with different collapsing strategies as described in Section 3. These visualizations will provide a static and a dynamic mode. Static visualizations show the complete cascade in order to provide an overview on the complete structure. Dynamic visualizations show how the cascades evolve over time with the option of dynamic collapsing. In addition to the structure, the temporal and geographical distribution are visualized on a map.

For both demonstration scenarios, cascades are going to be reconstructed using different influence assignment models as described in Section 2. Thus we observe how the topology of the cascade changes by selecting a different influence model.

6. CONCLUSION AND FUTURE WORK

We present an innovative system for real-time analysis and visualization of information cascades on Twitter, providing immediate insights on how information is spreading in social media. The front end provides rich structural, temporal and user-oriented information while cascades are unfolding dynamically. We devise visualization techniques in order to present big cascades by preserving relevant information and the underlying structure. Information cascades are reconstructed with different influence models while the demo tour supports both offline and online modes. Currently, we are working on the distribution of the social graph for faster access on the online scenario, where data is not known in advance.

7. REFERENCES

[1] E. Bakshy et al. Everyone's an influencer: Quantifying influence on Twitter. In *WSDM*, pages 65–74, 2011.

[2] N. Cao et al. Whisper: Tracing the spatiotemporal process of information diffusion in real time. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2649–2658, 2012.

[3] A. Guille et al. Information diffusion in online social networks: A survey. *SIGMOD Record*, 42(2):17, 2013.

[4] I. Taxidou and P. M. Fischer. Online analysis of information diffusion in twitter. In *WWW (Companion Volume)*, pages 1313–1318, 2014.