

# TwitViz: Exploring Twitter Network For Your Interests

Denzil Ferreira, Maria Freitas, João Rodrigues, Vítor Ferreira  
University of Madeira, Department of Mathematics and Engineering  
{lizned.arierref, carmofreitas, joao.pedro.rodrigues,vitor.dominor}@gmail.com  
twitter.com/{lizned.arierref, omrac, jprod, archimast}

## Abstract

In this work we introduce Twitviz, a desktop application that allows users to explore their social networks on Twitter based on their interests. Twitviz provides two main tools in the form social network graphs: the Keyword view displays relationships between keywords of interest and users tweeting about them; and the user's Network view emphasizing on reciprocal followers. We believe these visualizations provide an easy and useful way of identifying both known and unknown users with related interests, thereby promoting expansion and specialization of their social network.

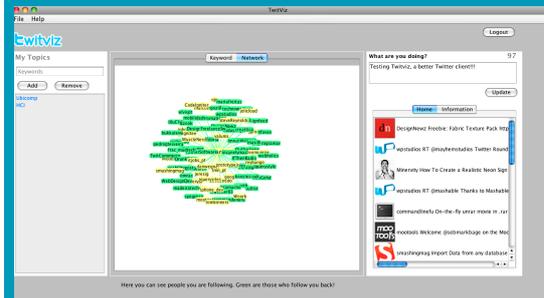
## Problem framing

One aspect that usually is neglected by social networking services is the lack of support for initial group formation process. Usually these services rely on already existing social networks databases and import them into a new user account. However, a motivating aspect of web social networks is the facilitation of the discovery of people, connections and communities. Twitter in particular provides poor support for discovery of communities of interest beyond following a public user that represents a company, event or group of interest. Ad-hoc discovery of topics of interest and their followers is limited to the keyword searching tools and the addition of "hashtags" for keyword searching as a way to categorize their tweets. Additionally, Twitter, like most social web applications, has a poor visualization of a user's social network in the form of a multipage list. Twitviz explores social network graphs to facilitate discovery of communities of interest and to identify users that reciprocally follow each other and therefore may be bound by similar interests.

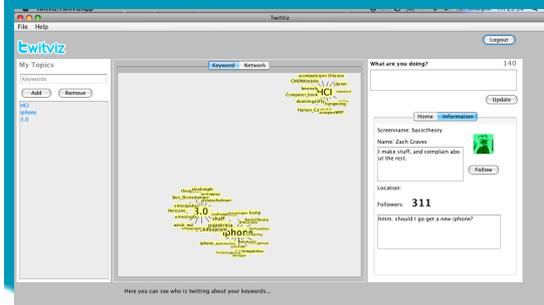
## Our solution

Twitviz is composed of a visualizations pane with two tabs: Keywords and Network. To provide input for the Keywords visualization, the user is provided with a simple form interface. Twitviz then queries the Twitter search service for updates containing those keywords. From the resulting updates, the application retrieves the originating Twitter users and builds a connection between the keyword in question and all the retrieved users. These connections are presented through visual graphs with user nodes that the Twitviz user may click to display another user's profile details and his/her last tweet. This visualization is updated every 20 seconds by retrieving the most recent posts from the Twitter public line and adding those users tweeting about any of the keywords of interest. The Network tab provides a graph of the user's first-level social network of followed users. Each user is represented as a colored label with yellow labels representing non-reciprocal followed users, while green labels represent users that reciprocally follow the current user. Additionally, Twitviz provides a simple Twitter client that allows the user to post updates and view a list of recent tweets from his network.

Our solution was implemented in Java (JDK1.6) and used the Twitter4J Java library for the TwitterAPI (<http://yusuke.homeip.net/twitter4j/en/index.html>). The Prefuse visualization toolkit (<http://prefuse.org/>) was used for the generation of the visualizations.



Twitviz Network view



Twitviz Keyword view

## Discussion

After the user successful login, using his Following list and reverse checking if the followed user has the user on his Following list, we can determine if the connection between both users is bidirectional. By coloring the nodes and edges of the visualization with green and yellow, users can distinguish followed users that follow them back. Information about the user is gathered upon clicking on the followed user, giving the possibility to the user to cancel his following status. If we had implemented a relationship historical database, we could highlight the users that had stopped/started following us, since last login. Another approach for differentiating the bidirectional connections from unidirectional could be by using the Prefuse Grouping API. This could be used to visually group nodes together. As we can't delete keywords (which are nodes on a GraphML file), due to a restriction on Prefuse GraphML API, we should have used a relational database for storing keywords and users. This would have allowed us to do faster queries on the nodes, reducing the amount of memory and file I/O operations Twitviz requires.