

Social Network Analysis Description

We create a framework for social network analysis based on a certain user.

The framework mainly analyze the following data of a host user: host's recent posts including post contents, post location and post time, host's friends and their recent posts.

In general, framework will query data from data plugin and deliver processed data to analysis plugin.

A data plugin is an adapter which call methods of API to raw data and do some simple format before parsing data to framework. A dataPlugin should provide two methods. In

extractPostsByName(), it is responsible for creating a list of Post which is a simple object defined by framework consisting of post content, location and time. In

extractConnectorsByName() method, it returns a list of string representing friends of a host user. When the user query for data, framework will call these two methods several times and update framework's userInfo field (**storing data in format**). UserInfo stores all the information needed in an analysis request. It consists of a host userBean and a list of friends userBean.

UserBean is a format data object storing data and providing data access.

Considering rate limit, we support data caching feature. Framework stores information of a UserBean in local files. Filename is unique in relation to userName. When requesting data, framework will first look for in local file. If the local copy doesn't exist or information is outdated, it will then request from data plugin.

Framework would then preprocess the data required form data plugins. It can sort posts by longitude, latitude, time and length. It can extract locations, times and contents of host's recent posts. It can compute a user's recent location by refering to locations of his recent posts. Framework can also compute host's activeness based on the time/date of his recent posts. It also provide these process result for host's friends. So the analysis plugin can compare the user with his friends in multiple dimensions. Framework don't parse all these process result to analysi plugin when trigger it's analysis process, instead, analysis plugin has a reference to framework which means plugin have indirect access to these processed data. When a user choose to start analysis (clicking start button), framework will call analysis plugin's analyze() method which return a JPanel to framework.

Constraints and extention: Our framework queries the information focused on a host user and his friends. The data in our scope are post's content, post time and post location. So analysis plugin is restricted on this limited scope of data, and it should only analyze a sinlge user in the network. Plugin can get data by calling framework's method. The framework provide some kinds of preprocessed data as mentioned above. We recomend analysis plugin to do analysis based on these extracted data (posts location, user's location, post's time, user's activeness, etc.) without implementing their own data processing. But client also have access to unprocessed data (original data in the format we defined), they can process the data in their own way and analyze on the processing result.

We made a interface of Framework as well as a FrameworkImpl as the implementation; and each analysis plugin has a field of the type "Framework". By this means, we made the plugins also reuseable and extensible. Because we can make other implementation of the Framework interface without changing the field inside the plugins.