Homework #4: Social Network Analytics

In this assignment you will design and implement a social-media analysis framework for social networks. Your framework will define a basic skeleton for developers to implement support for other social networks and display their own social-media analyses. The framework will simplify the implementation of additional data sources and analyses by providing unifying mechanisms such as querying social networks for data, delivering query results to the plugins, and possibly storing messages and making the data available over time.

Your framework will support two types of plugins:

- *Data plugins* that execute queries on a social network when the framework requests social network data, returning the resulting data to the framework to be processed. Data plugins are triggered by the framework and do not directly communicate with other plugins.
- *Analysis plugins* that analyze and visualize data provided by the framework. An analysis plugin waits for data from the framework but does not directly communicate with data plugins or social networks. The framework will produce data and notify analysis plugins that new data is available, after which the plugins update their graphical analysis of the data that has arrived.

At the conclusion of this assignment your program should be able to display multiple analyses simultaneously to the screen. Your framework should allow a plugin developer to implement additional data sources and data analyses with only a small amount of work. Each data analysis plugin should be able to analyze data from all data sources. Thus, by implementing a new analysis plugin, all supported social networks benefit immediately from that new analysis. Similarly, providing support for a new social network as a new data source immediately benefits from all existing data analyses. By supporting two types of plugins, your framework encourages development of interesting analyses as well as the ability to the apply those analyses to arbitrary social networks.

Learning goals

The learning goals for this assignment are:

- Design and implement a black-box framework. Plan for reuse by identifying hot spots and cold spots, and expose extension points through a plugin interface.
- Perform domain analysis to identify commonalities and differences in social network data, APIs, and data analyses.
- Provide a common interface for plugins to query data from social networks.
- Provide a common interface for plugins to analyze social-network data.

- Develop plugins for a framework written by others.
- Reuse existing open-source, third-party libraries.

Be creative when you design and implement your framework's core features. You get to choose what types of data your framework will make available, as well as how and when the framework will deliver data to the plugins. Likewise, be creative in how your plugins will process this data and display analysis results to the screen. If you are unsure whether your framework meets the above requirements, please ask us on Piazza.

We have provided an example framework in the Appendix A that you can use as a reference as you brainstorm, design, and implement your framework's core features. We strongly recommend that you consult the appendix before reading any further.

Designing a framework can be challenging. We recommend you perform a domain analysis following these steps to get started:

- Determine what concepts are shared between Twitter, Facebook, Google+, and other social networks. Gain a basic familiarity of their general terms.
- Think of a few interesting ways in which social data could be analyzed and graphically displayed. Will it be possible for plugins to implement interesting, unique graphical analyses using the data your framework provides? What types of data analyses will be possible, and which in your opinion seem most interesting? What tasks are common to many analyses and could be performed by the framework? Search Google for existing social network analyses to understand the range of possible analyses; two good examples are Xefer and whotwi.
- If you are unsure whether certain data is available from the API, read the documentation or post a question on Piazza.

  Your analyses should only require read-only permissions, so—depending on the API—you likely won't need any POST methods or methods which retrieve info for a specific authenticated user.

  Additionally, you should be careful about API rate limits. Most APIs limit the number of API calls you may execute in a given period of time; the rate limit varies between different APIs. For Twitter the rate limit quota refreshes every 15 minutes and the limit varies for the different API call types. Google+ has a limit of 10,000 requests per day and 5 requests per second.
- Determine potential hot spots (extension points for plugins) and cold spots (commonalities implemented within the framework) of your framework. Decide what features your framework will provide and which decisions should be left to the plugins. This is a design decision; there is no single correct answer. Where possible make your framework general and support a broad set of analyses and data.

- It is helpful to consider the lifecycle of the framework as consisting of two stages: an *initialization stage* and a *query/delivery stage*.
  - The *initialization stage* is when the framework is first started and plugins are registered with the framework.
  - The *query/delivery stage* is when the framework silently works in the background, responding to user input, querying the data plugins for data, and delivering data to the analysis plugins.

  This might provide some insight on how to design your plugin interface, for example, to determine which lifecycle methods the framework will call when the plugins are first registered and initialized, and which callback methods you must define to send data to the plugins.

Implement and document your framework and write sample plugins for your framework. You must implement one data plugin and one analysis plugin.

Requirements for your framework

You have much freedom to be creative in your framework design and implementation. Your work, however, must satisfy the following requirements:

- Your framework must initialize the user interface and allot a portion of the screen for each analysis plugin to be displayed. The framework must be able to run multiple analysis plugins in the same application, but the analysis plugins do not need to be displayed at the same time.
- Your framework must support multiple data plugins. Your framework may support querying from multiple data plugins in the same application, but it is also sufficient to select one data plugin at a time (e.g., at startup).
- The data plugins must query social networking sites and return data in a common format to the framework. The framework must trigger the analysis plugins to update the analysis with new data. The analysis plugins should visualize new data when it is delivered by the framework.
- The analysis plugins should not request specific data from the framework (e.g., the data of a specific user). The framework provides the same data to all analysis plugins. But, the framework may ask an analysis plugin whether it needs special data and then get that special data for the analysis plugin.
- There should be no direct communication among multiple plugins; all communication is orchestrated by the framework. Analysis plugins should depend only on the framework and should not interact directly with other plugins or initiate direct queries to a

data source. Each analysis plugin should work with all data plugins, and vice versa.

- Your data plugins should use third-party libraries to obtain data from the social networks. We have suggested some libraries in Appendix B. You may use more libraries—including libraries for data analysis and visualization—if you wish.

Your sample plugins should be fully functional. Your sample analysis plugin(s) may be very simple though; you do not need to use external libraries or advanced visualizations.

We recommend (but do not require) that you test your framework implementation and data structures with JUnit tests. See the lecture slides for advice on how to test with stubs.

Evaluation

We will grade your work approximately as follows:

- Core framework implementation: 100 points
- Sample plugins: 100 points

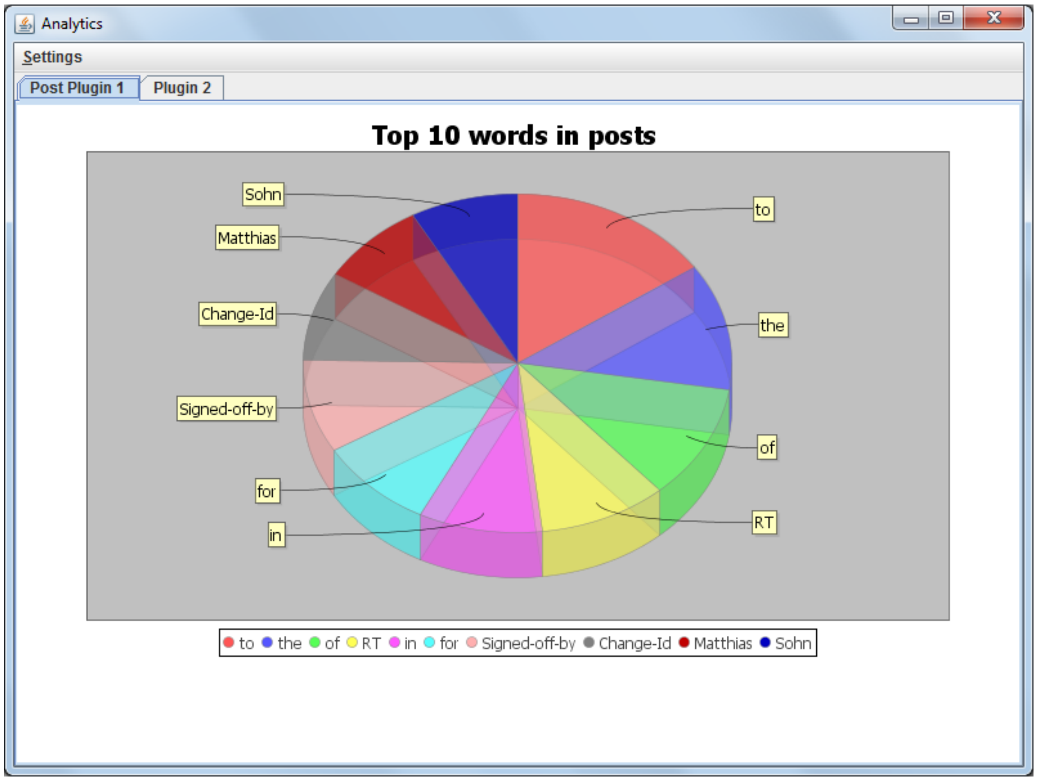
For full credit we expect:

- A framework design that performs a useful and reusable part of the analysis task.
- A framework that remains in control of the execution and orchestrates the behavior of all plugins.
- An implementation of the framework that satisfies the requirements.
- Appropriate use of external libraries: one library in the framework to access social-media data and one library in at least one analysis plugin.
- Clear, readable code that follows standard naming conventions and good style.
- Absence of critical bugs.
- A clean build that runs your framework.

As usual, this assignment allows substantial creativity and a wide range of excellent successful solutions. When in doubt about the assignment's requirements, use your best judgment, and document any assumptions you make.

Appendix A: Comparing User Posts

This section describes one possible framework that would meet the requirements of this assignment. You may implement other framework variations, however, and we encourage you to do so.



In this example, the analysis plugin analyzes the posts of a user, specified by a user ID. The specific user ID and data source can be selected in the “Settings” option of the framework. Note that the definition of a “post” could vary depending on the data source selected: a post in Twitter might be defined. The framework gets the posts from the data plugin and provides that information to the analysis plugin. The analysis plugin fills a `JPanel` (provided to it by the framework) with the analysis results. Our example only analyzes data from recent relevant queries; the example framework does not cache or store posts from previous queries for an extended period of time, although doing so is possible and would allow much richer analyses.

Note that while the above example only contains one visible analysis, the other analysis plugin is still receiving info even though it’s on the other tab and thus not currently visible.

Appendix B: Recommended Libraries

We recommend the following libraries for these social networks, though you are free to use others social networks and library wrappers:

- Twitter
  - Twitter4J Library
  - API Key Registration
  - Twitter API Documentation
  - Twitter Rate Limits
- Google Plus
  - Google+ Client Library
  - Quick Start (Note, this guide is outdated. You instead of selecting **Registered Apps** → **Register App** you should select **Credentials** → **Create New Key.**)
  - Google+ API Documentation
  - API Quotas
- Facebook
  - Facebook4J Library
  - App registration
  - Facebook Graph API
  - Facebook Rate Limiting

Avoid pushing passwords or private tokens to GitHub repositories. Load private data from separate files not committed to the repository.