Homework #5: Social Network Analytics

In this assignment you will work as a team to design and implement a social-media analysis framework for Twitter and other 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. You will also implement sample plugins for your framework and implement plugins for another team's framework.

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 framework 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 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 data source immediately benefits from all existing data analyses. By supporting two types of plugins, your framework encourages the 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 exposing 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.
- Demonstrate effecting testing techniques for interactive software components.
- Develop plugins for a framework written by others.
- Reuse existing open-source, third-party libraries.
- Coordinate software design and development within a team and between teams.

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 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.

Milestones and Deadlines

This assignment contains a team sign-up deadline and three milestones:

- Sign up your team (due Thursday, October 30, 11:59 p.m.)[1]
- Milestone A: Design and present your framework (due Wednesday, November 5, 8:59 a.m.).
- Milestone B: Framework implementation and sample plugins (due Thursday, November 13, 11:59 p.m.). Submit your work by Friday, November 14, 10:00 a.m. to be considered as a framework selected for plugins for Milestone C.
- Milestone C: Plugin implementations for another framework (due Thursday, November 20, 11:59 p.m.).

For this assignment you must work in teams of 2-3 students, using a shared Git repository to coordinate and turn in your work. You must form your team and sign up for a design presentation time (Milestone A) by Thursday, October 30, 11:59 p.m. using the web form and instructions that we will post to Piazza. Presentation slots for each recitation time are limited and available on a first-come-first-served basis, so please sign up as early as possible if your team is only available for limited recitation times.

If you are looking for team members to work with, you may use the Piazza "Search for Teammates" feature.

[1] See https://piazza.com/class/hxjs50tyk8q63q?cid=676 for more information

Late days

Each team may use up to a total of 2 team late days for this assignment in Milestones B and C. Your team may **not** use any late days for Milestone A. If you submit Milestone B after Friday, November 14, 10:00 a.m. your team cannot be considered as a framework to be selected for plugins for Milestone C—so try to use few late days for Milestone B if possible.

Your team's late days for this assignment are independent of your team members' late days for other assignments in this course. In other words, the number of late days you have used on previous homeworks does not determine the number of late days you can use for this assignment. Likewise, the number of late days your team uses for this assignment will not affect the number of late days you can use for Homework 6.

Milestone A: Design and present your framework (due Wednesday, November 5, 8:59 a.m.; you may not use late days)

For the first milestone you will design your framework and present your design in recitation. Your team will submit three files: (1) a short, high-level description of your framework, (2) an object model for your framework, and (3), a presentation describing your framework and some of the design decisions you made. Because all teams must present their designs during recitation, **you may not use any late days for Milestone A**.

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+, GitHub and other social networks. Gain a basic familiarity of their general terms. Your framework must be compatible with Twitter and GitHub; see Milestones B and C.
- 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, and GitHub has a limit of 3,600 authenticated requests per hour and 60 unauthenticated requests per hour.
- 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.

We describe the files you must submit for Milestone A below. You should turn in each of these files to the `hw5a` directory in your team's shared Git repository before the Milestone A deadline.

A high-level description of your framework

In `hw5a/description.pdf` write a short (less than 1 page), high-level description of your framework. Describe the features your framework provides for its plugins: the data your framework requests from data plugins and provides to analysis plugins, how and when will it query for this data, and how and when it delivers results to the plugins. You should discuss how the data provided can be queried from the APIs. Also describe the extension points that clients can plug into and any constraints the framework imposes on its plugins.

An object model of your framework

In `hw5a/design.pdf` create an object model (written as UML class diagram) for your proposed framework. Your object model does not need to be exhaustive, but should include the representation of the social network data, the plugin interfaces for data and analysis plugins, and the core of the framework infrastructure. It helps to also sketch the interaction between framework, data plugins, analysis plugins, and the GUI as an interaction diagram. We do not require, but strongly encourage, such interaction diagram(s). Your design does not need to mention Java Swing classes/methods, or any details related to the third-party libraries your framework will use.

A presentation of your design

In `hw5a/presentation.pdf`, prepare a short, 5-7 minute presentation describing your

framework and the key design decisions you've made. Describe the common tasks your framework will implement and the extension points it will provide to plugins. You will not have time to describe your object model in detail, but highlight your main decisions and the plugin interfaces, as well as the interactions between the plugins and the framework. We encourage you to use not more than 5 slides for the presentation.

Your primary audience is your peers and your goal is to convince them to write a plugin for your framework. Focus on the important design decisions you have made and how these decisions will make it easy to implement plugins.

Your entire team will give the presentation in the recitation you signed up for. All team members should have an active part in the presentation. A time limit of 7 minutes for the entire presentation will be strictly enforced.

Milestone B: Framework and sample plugin implementation (due Thursday, November 13, 11:59 p.m.)

Implement and document your framework and write sample plugins for your framework. A two-person team must implement two data plugins and one analysis plugin. A three-person team must implement two data plugins and three analysis plugins. One of your sample data plugins must be for Twitter.

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, and we also included `Twitter4J` and `egit-github` in your team's repository. You may use more libraries—including libraries for data analysis and visualization—if you wish.
- You must test your framework implementation with JUnit tests. Your tests should

achieve reasonably good coverage and should demonstrate good unit testing principles. You must use test stubs, rather than your actual plugin implementations, to test your framework; a test stub is just an implementation whose purpose is to simulate the behavior of some software component (such as a plugin) for the purpose of testing another software component (such as your framework). We also recommend, but do not require, that you test your plugin implementations.

- Your code should build on **Travis CI** and be runnable using an Ant `run` target.

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.

Your framework should be designed and *documented* so that other teams could use your framework in Milestone C. Apply the principles for writing understandable software that you have learned in this course: descriptive class and method names, information hiding and encapsulation, design patterns, etc. Provide clear documentation so that others could easily write plugins for your framework, including Javadoc documentation for all protected and public methods and a short `hw5b/README.txt` file explaining the steps required to implement plugins. Your sample plugins should be a useful example for others.

In the `hw5b` project in your team's shared Git repository, turn in your framework implementation in the `edu.cmu.cs.cs214.hw5.framework` package and turn in your sample plugin(s) in the `edu.cmu.cs.cs214.hw5.plugin` package.

Finally, remember that your team must submit your framework implementation by Friday, November 14, 10:00 a.m. to be considered as a framework to be plugged into for Milestone C. If your team's framework is selected, then you will receive 10 points extra credit for Milestone B and you will not need to write any plugins for Milestone C.

Milestone C: plugin implementations for another framework (due Thursday, November 20, 11:59 p.m.)

Soon after the Milestone B deadline we will select several framework implementations for other teams to plug into for Milestone C. We will select from frameworks finished by Friday, November 14, 10:00 a.m. . If we select your framework your team must support your framework for others. If we do not select your framework you must implement plugins for one of the selected frameworks.

Supporting a selected framework

If your framework is selected you must provide technical support for the teams building plugins for it. This includes answering their questions promptly on Piazza, fixing bugs in your framework (if necessary), and addressing misunderstandings teams have about your framework. Overall, your goal is to keep the other teams happy and ease their task of implementing plugins for your framework.

As you support your framework you should keep all communication between you and other teams public, using Piazza for all technical support. This is because, (1) you will be graded by the quality of technical support you provide, and (2) all your development teams can benefit from seeing the problems addressed by others, as well as how they were eventually resolved. You should not provide support beyond what is legal, socially acceptable, or that would violate the course cheating policy. If in doubt, ask the course staff.

At the conclusion of this assignment, you must also submit a short report (1 paragraph to at most 1 page) of your experience providing support for your framework. Please submit the file in `hw5c/feedback.pdf` before the Milestone C late due date.

Writing plugins for another team's framework

If your framework was not selected, your team must develop plugins for another team's framework. Two-person teams must develop two data plugin and two analysis plugins. Three-person teams must develop three data plugins and three analysis plugins for the other team's framework.

One of your data plugins must be for GitHub. At least one of your analysis plugins must use an extra third-party library; you may use any library you want, i.e., graph or text-search libraries for your data analysis, or a library with graphical widgets to visualize results, such as JFreeChart. Your plugins for Milestone C cannot use the same social network (Twitter) or analysis as the sample plugins for the framework you are plugging into, but you may otherwise base your Milestone C plugins on your team's own plugins from Milestone B.

You will access a selected framework and its documentation by cloning the public Git repository of another team and referencing the cloned repository as a project dependency in Eclipse; we will post some instructions to Piazza for how to do this Piazza in the coming weeks. We will also designate a Piazza thread for each selected framework for you to communicate with the members of the selected framework teams. Please also note that you may not change any framework code directly; your plugins must work for the official version of the framework you are plugging into. If you think you have found a bug in the framework, report the problem on the framework team's support thread for them to address.

Please submit your plugins to the `edu.cmu.cs.cs214.hw5.plugin` package in the `hw5c` project located in your team's shared Git repository. At the conclusion of this assignment, you must also provide feedback (1 paragraph to at most 1 page) on the quality of technical support from the framework-providing team. Please submit the file in `hw5c/feedback.pdf` before the Milestone C due date.

Evaluation

We will grade your work approximately as follows:

- Milestone A (75 points):

  Framework description: 10 points

  Object model: 35 points

  Presentation: 30 points

- Milestone B (125 points + up to 10 points extra credit)

  Core framework implementation: 65 points

  Sample plugins (numbers specified above): 35 points

  High-quality documentation for plugin developers: 20 points

  Successful build on Travis CI and an Ant `run` target: 5 points

  Extra credit if your framework is selected for Milestone C: 10 points

- Milestone C (75 points)

  Data and analysis plugins (numbers specified above): 65 points

  Technical support for a selected framework: 65 points

  Experience report: 10 points

For full credit we expect:

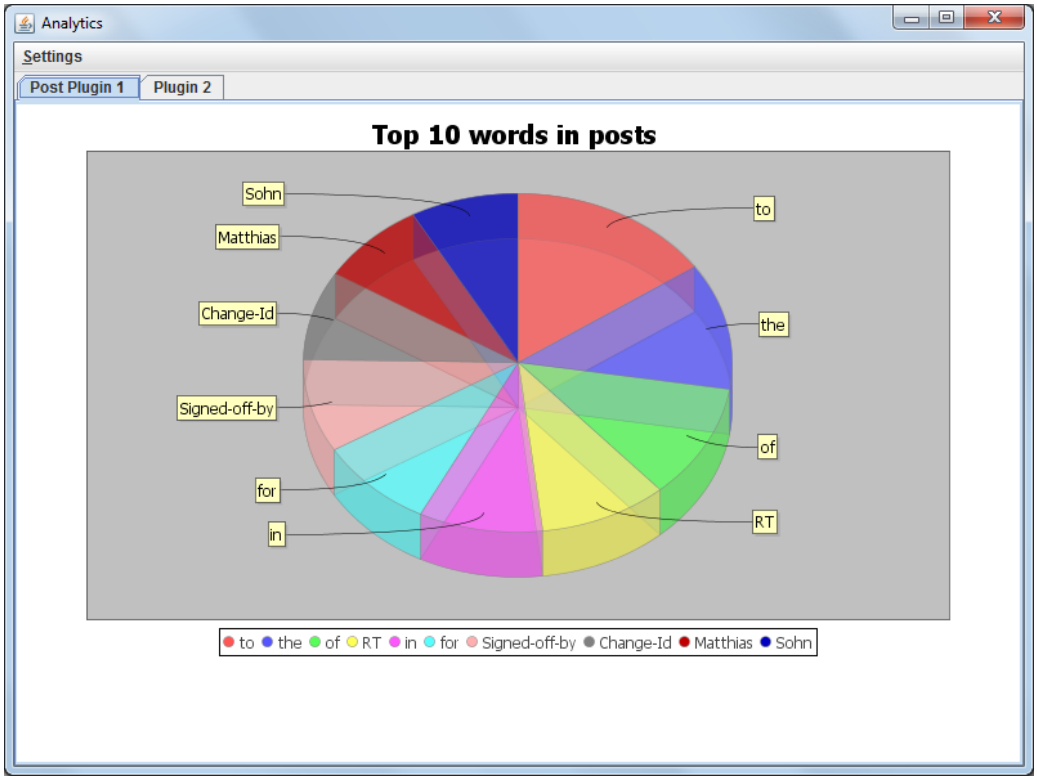
- A framework design that performs a useful and reusable part of the analysis task.
- A framework that remains in control of the execution and orchestrates the behavior of all plugins.
- A framework which can query data from at least two social networks, one of which must be Twitter (in Milestone B) and one of which must be GitHub (in Milestone C).
- An object model and engaging presentation that reflect and justify your design decisions and describe the hot spots and cold spots of the framework.
- An implementation of the framework that satisfies the requirements in the introduction and Milestone B section of this document.

- Principled testing of your framework implementation, including the use of test stubs for your plugin interfaces.
- Appropriate use of external libraries: one library in the framework to access social-media data for each social network and one library in at least one analysis plugin.
- Clear, readable code that follows standard naming conventions and good style.
- Absence of critical bugs; we will run FindBugs, and you should run FindBugs, too.
- A clean build on Travis CI and an Ant `run` target 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 as a Tweet for Twitter but a commit message in GitHub. 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
- GitHub
  - GitHub Library
  - Developer Registration
    - You may use personal access tokens instead of registering an application since you only are required to access public information.
  - GitHub API Reference

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