OpenRefine (Phase 1 enhancements)

November 5, 2017

OpenRefine Foundation

Overview

This proposal summarizes a portion of the effort to enhance OpenRefine with additional features and fixes requested from our community and in particular news organizations. The majority of this proposal is aligned around smaller and easier to implement code changes and enhancements in OpenRefine.

Future proposals (not this one) will allow even larger data sizes to be handled, a faster and enhanced streamlined UI, and batch mode operations.

Goals for Phase 1 Enhancement Funding

With this proposal, in general, we wish to enhance our users' capability towards the following:

1. enhancing collaboration and reconciling with new metadata editing
2. re-enabling the ability to reconcile data with reconciling services such as Wikidata.
3. re-enabling collaboration by exporting project to Google Drive.
4. new import and export capabilities with databases supporting JDBC
5. GUI enhancements to support larger data size and usability
6. enhanced documentation, tutorials, translations, community events, and support mechanisms.

Development Milestones

I. Metadata Support - COMPLETED

The community wants a way to add metadata about a project, such as citing sources, descriptions, dates, categories, etc. The can be done by providing an edit metadata option and storing the data and metadata to be shared with an OpenRefine project.

II. Wikidata Reconciling - COMPLETED

Previous versions of OpenRefine supported a Freebase reconcile option (now deprecated). The community wanted a replacement service and interface to handle matching a column of entities to a Reconcile Service API in order to pull in and add additional data for that column of entities.

This work involves many parts, some of which are highlighted here:

1. A Wikidata Reconcile Service endpoint - This is now completed but requires some additional changes to support additional reconcile options for users.
2. A new and improved GUI interface that allows easier matching by having a split view that can be moved or browser tabbed and has a larger window rather than the current small fixed dialog window, in order to give more information and make it much easier for our human users to make judgments while reconciling.
3. A schema alignment view to allow uploading users data to Wikidata against their data model. - Some of this has been completed but additional warning dialogs, informational messages, and general cleanup is needed. There might also be additional work for options based on community feedback once the Wikidata Reconciling feature is available in the upcoming OpenRefine release.

III. Google Spreadsheets import/export - COMPLETED

Previous versions of OpenRefine had the ability to import and export to Google Spreadsheets, but unfortunately, over the years our code has not been kept up to date with the latest Google API. This needs to be updated so that users can share OpenRefine projects and data for easier collaboration among peers and teams.

IV. Database Table import/export - COMPLETED

Historically, OpenRefine has been limited compared to other data tools in that it does not have a way to connect to a database table. This is especially useful at export time, when there is a need to save a cleaned CSV for example into a database table. Importing from a database is useful also. It can help to join clean data in a database table against messy data in OpenRefine, in order to clean and prepare it for use. Database Drivers exist for many databases such as Oracle, MySQL, Postgres, and even many schema-less databases such as MongoDB. Most database drivers use JDBC which makes it easier for us to develop against, and others typically use a custom Java driver that sometimes is non-trivial to integrate with. Since OpenRefine is built with Java this should be relatively straightforward to utilize existing JDBC drivers for our import/export operations and for support of MongoDB there is a Java driver available.

V. GUI enhancements - NOT STARTED

The current datagrid view in OpenRefine does not support many things expected from a typical spreadsheet that helps work with rows and columns of data. The following is a listing of some of the major usability features that impact our community and have been repeatedly requested:

1. Modern UI framework to support faster rendering & enhanced extensibility
2. Allowing resizeable columns
3. Faster rendering especially with Record mode view from XML and JSON data
4. Toggle option for always showing unprintable characters

Planned Support Activities

Planned Event Attendances:

Following the Python foundation model, the OpenRefine foundation provides grants to

- Help community members organize events;
- Help members to attend or speak at conferences to promote OpenRefine;
- Organize a conference for OpenRefine developers and contributors. The goal of this event is to focus on developing the contributor community and not promote usage of OpenRefine. Themes (among others) covers
  - How to contribute to the core;

- How to write an extension;
- How to contribute to the documentation.

General Support Efforts:

OpenRefine code base is based on legacy technologies and architecture patterns. They present technical barriers for new contributors to engage with the community and limit the attractiveness of OpenRefine for developers. In addition to the GUI enhancement, we want to:

- Refactor the application to follow modern framework guideline and separate the back and front end;
- Improve both the user and developers documentation.

Those changes will take several months to implements and will be done via long-term sponsorship programs like:

- Fellowship programs with a couple of experienced developers
- Mentorship of more junior developer via partnerships with university and Google Summer of Code programs.