Weekly Updates for Fung Institute Patent Database

Gabe Fierro
Coleman Fung Institute for Engineering Leadership
UC Berkeley
fierro@eecs.berkeley.edu

November 12, 2013

1 Introduction

Here, we discuss the infrastructure and configuration for a weekly process that downloads the latest weekly releases of USPTO patent files, runs the requisite data transformations, and integrates the processed data into the larger database.

This system has not yet been fully adjusted to deal with the weekly application releases as well as the grant releases, but development is underway and this document will be updated once those changes are made.

2 Architecture

There are five main stages to the patent processing infrastructure:

- **Parse**: Downloads the latest patent files, interprets the structured data, and inserts the compiled records into the database
- **Clean**: Runs the assignee, lawyer and geographic disambiguations on the data
- **Consolidation**: Prepares the raw inventor data for being sent to the disambiguation engine
- **Disambiguation**: Performs entity resolution on the raw inventor data
- **Integration**: Incorporates the disambiguated inventor records into the database

The code represents these stages as modular chunks that can be configured and run independently of each other. For the weekly update system, these stages need to be run end-to-end with no interspersed configuration.

At the time of writing, the complete process is not fully automated. The consolidation stage will output a file `disambiguator.csv` that is formatted for input into the disambiguation engine. After the disambiguation engine is run, the integration stage must be run manually. This manual step will be eliminated once the disambiguator has been properly configured.

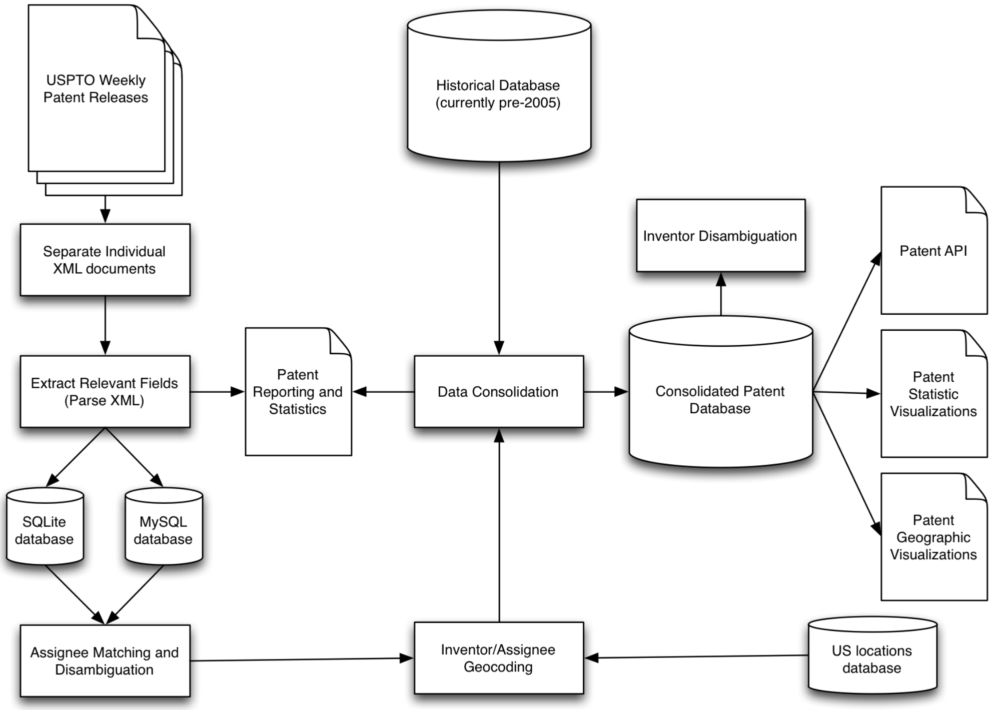
USPTO Weekly Patent Releases
Separate Individual XML documents
Extract Relevant Fields (Parse XML)
SQLite database
MySQL database
Assignee Matching and Disambiguation
Patent Reporting and Statistics
Historical Database (currently pre-2005)
Data Consolidation
Inventor/Assignee Geocoding
Inventor Disambiguation
Consolidated Patent Database
US locations database
Patent API
Patent Statistic Visualizations
Patent Geographic Visualizations

Figure 1: Toolchain for processing patents

3 Configuration

The ‘start.py‘ script provides a simple interface to the patent processor by means of a configuration file. Here is a sample configuration file:

```
[process]
parse=test
clean=True
consolidate=True
outputdir=.
lowmemory=False

[test]
datadir=test/fixtures/xml
dataregex=\d{4}_\d.xml

[2012]
downloaddir=tmpdownload
years=2012
```

The `[process]` section is mandatory, as it defines the roadmap of the data process. Each of the lines below the `[process]` line are configuration options. `parse` indicates which of the parsing configurations will be run (see below). `clean`, if `True`, runs the cleaning step on the output of the previously run parse. Similarly, `consolidate`, if `True`, runs the consolidation step on the output of the cleaning step. The `lowmemory` option, if `True`, will run the cleaning and consolidation steps in such a way that they require less memory, but are slower and sometimes less accurate. It is recommended to leave this as `True` unless the host computer has more than 64 GB of RAM.

The `parse` section takes as input the title of another configuration section, which defines the options for which patent files are to be downloaded and parsed. `parse` sections accept the following configuration options:

- **datadir**: specifies the path to the directory containing the XML files that we wish to parse. The path will be evaluated relative to the root directory of the preprocessor
- **dataregex**: specifies the regular expression that matches the filenames of the XML files we want to parse. This defaults to `ipg\d{6}.xml`, which matches USPTO grant files since 2005.
- **years**: specifies the range of years we want to download and parse. If the current year is specified, the script will download all possible files. If this option is provided, the `datadir` option will be ignored, and the files will be downloaded to the directory indicated by the `downloaddir` option (see below). If this option is *not* provided, then the parser will operate on the contents of `datadir`.

  Years are to be separated by commas, and ranges are indicated by using dashes. For example, to download the years 1995, 1997, 1998, 1999, 2000, 2001 and 2005, we would indicated this as `years=1995,1997-2001,2005`.
- **downloaddir**: specifies the target directory into which the needed patent files will be downloaded. This directory is evaluated relative to the root directory of the processor. If it does not exist, it will be created. If the directory already exists and already contains pre-downloaded files, the process will only download the needed files to avoid unnecessary work.

The final configuration section, `[xml-handlers]`, specifies which XML parser is to be used for which files. The USPTO has used eight different data schemas for patent grants and because it is difficult to automatically detect the schema of an XML document, this section makes it possible to indicate which XML parser should be used for which dates of patent releases.

This section should only have to be touched when a new parser is introduced. A date is indicated by either YYYY or YYYYMMDD, and ranges are indicated by dashes in between two dates. If only one date is provided, then the corresponding parser will be used for all subsequent dates. The current configuration is listed here:

```
[xml-handlers]
2005-20130108=lib.handlers.grant_handler_v42
20130115=lib.handlers.grant_handler_v44
default=lib.handlers.grant_handler_v42
```

4 Running

The installation process has been developed and tested on machines capable of running BASH, which is Mac OS X and most Linux environments.

To setup a local environment to run the preprocessor, it is best practice to setup `virtualenv` to handle local Python packages.

```
git clone https://github.com/funginstitute/patentprocessor/
cd patentprocessor
pip install virtualenv
virtualenv venv
. venv/bin/activate
pip install -r requirements.txt
```

Configure the database connection in `lib/alchemy/config.ini`, then configure the data process in `process.cfg`. The patent processor can then be started by running `python start.py process.cfg`.