***Article***

**Using Python and SQL increase the efficacy of searching through academic research**

**Hope Tambala**

Searching through academic research used to be a daunting task before computerization of the libraries inventory. And after the creation of online databases, the process became even more streamlined. To help further increase the efficiency for searching through information, I have created a program called “ChemFoParser.” ChemFoParser, short for Chemical Information Parsing, reduces the time to search through multiple journals to a few clicks in a program.

**INTRODUCTION**

The creation of a vast online database for reference services, such as MLibrary, has made researching more efficient and empowering the researcher with thousands upon thousands of journal articles, newspapers, reviews, and magazines that can be used at any computer, at home or in the library. “Readily available reference services have been a key component of the University of Michigan Library's (MLibrary) success in supporting and advancing campus research, teaching, and learning,”(reference).

The evolution of reference services suggests that strategies and innovative approaches will primarily focus on increasing the amount of information that will be at the fingertips of researchers. However, searching through this vast source of information can be tedious especially when you’re looking for articles under a specific discipline. We’ll use MLibrary as an example for the length of searching. First, the researcher opens an Internet browser and types in “MLibrary.” Receives search engines results, presses “MLibrary” link. Go to search field, type in certain keywords. Receive results from query. Many of the initial results you receive are from irrelevant sources. With the advanced options, the researcher will eventually receive a list of papers that would be relevant references. While MLibrary is a powerful tool, it requires more time to receive a list of respectable resources.

ChemFoParser’s goal is to streamline that process by pre-choosing the top-ranked journals in a given field, i.e. chemistry. By allowing the user to search through specific chemistry journals, they obtain relevant information from journals they trust. The purpose of this paper is to showcase how to use the program and present parts of the relevant Python and SQL code. In doing so, this paper will provide a step-by-step walkthrough of the search process used for this program.

**METHODS**

This paper divides the methods into 3 Sections: Sources, Python Code, and SQL code.

*Sources*

ChemFoParser uses the top 50 lists from SCImago as a reference for its own library of chemical information (reference). SCImago measures the journals impact, influence, and prestige by using a variety of parameters. “It expresses the average number of weighted citations received in the selected year by the documents published in the journal in the three previous years.” This is expressed as the “SJR” indicator.

The *SJR indicator* is a free journal metric which uses an algorithm similar to [PageRank](http://en.wikipedia.org/wiki/PageRank) (reference), the Google search results algorithm. There is another prominent ranking system developed by Thomson Reuters. The average citations per document in a 2-year period, abbreviated as Cites per Doc. (2y), is another index that measures the scientific impact of an average article published in the journal. SJR was used due to being widely available to the public and faster to use.

SCImago ranking list also includes the H index (Journal’s number of articles (h) that have at least received h citations over the past 3 years, number of articles published in that journal in the last three years as well as the number of references included in the journal’s published articles.

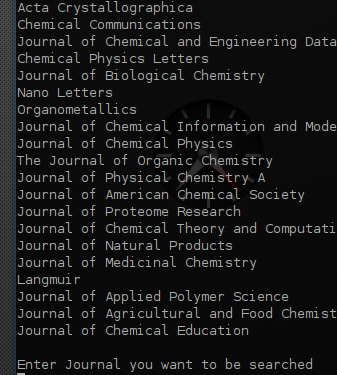
Each of these journals on the list have a host website. Each website has a RSS Feed for their most recent journal entries. RSS stands for "Really Simple Syndication". It is a way to easily distribute a list of headlines, update notices, and sometimes content to a wide number of people. It is used by computer programs that organize those headlines and notices for easy reading [REFERENCE].

This project uses RSS feeds due to rapid and recent publication of scientific journal articles. Many people are interested in many websites whose content changes on an unpredictable schedule. Repeatedly checking each website to see if there is any new content can be very tedious. Ranking this data could expedite news aggregation for personal or professional use. Due to the ever-evolving nature of these feeds, it seemed perfect for this project. The nature of these feeds allows the program to stay relevant and current

*Python Code*

Python is the framework and foundation of the ChemFoParser. Python 2.7.6 was used for this program.

Universal Feed Parser is a Python module for downloading and parsing syndicated feeds. It can handle RSS0.90, Netscape RSS 0.91,Userland RSS 0.91, RSS 0.92, RSS 0.93, RSS 0.94, RSS 1.0, RSS 2.0, Atom 0.3, Atom 1.0, and CDF feeds [REFERENCE]. It also parses several popular extension modules, including Dublin Core and Apple’s iTunes extensions. This library retrieves feeds from the journal websites in the sources list. This library’s functions allow me to retrieve each research paper’s author, title, and published date.

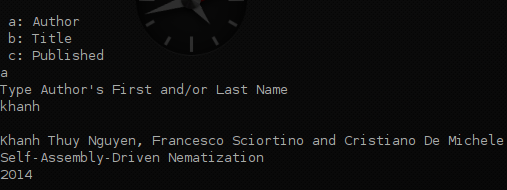
 Another library used was MySQLdb. MySQLdb is a thread-compatible interface to the popular MySQL database server that provides the Python database API.

*SQL Code*

SQL was used to create a temporary database to hold the information requested by the user. By holding this information in a database, ChemFoParser’s python script can perform quick and efficient queries from database ranked in a variety of ways. Due to the simplicity yet powerful statements in SQL, the program can execute elaborate and creative commands on the data. This includes the number of times a keyword is found in title of a research paper, specific author, and papers

**RESULTS**

Figure 1,2, and 3 represent screenshots of the program as it retrieves the RSS feed, asks for user input for several specifications for ranking, and the results of the query. Figure 1 is a screenshot of the list of journals printed at the opening of the program. Figure 2 gives a screenshot of the 3 ways the data queried can be organized. Figure 3 gives one of the results possible if the user searches the papers by the author’s name.

**DISCUSSION**

*Limitations*

Along the development of ChemFoParser, a few limitations were found with a few of the methods used.

Feedparser retrieves only what is found in the RSS Feed. There are many relevant categories to include helping with ChemFoParser’s ranking system, but not having the ability to pull an abstract significantly dampens the usefulness of the program. Feedparser pulls many details from a RSS feed. Author, title, published date, website link, and description of the feed for documentation. The latter two are irrelevant for ranking data. More development in this area will ensue.

Another problem that was encountered was the encoding of the information retrieved from the RSS feeds. SQL struggled with a few of the characters used in author’s names for research papers. This problem was rectified by essentially using python to forcefully encode and change those characters into readable text for SQL to read-in. However, these characters were changed into characters that may not represent the author’s name accurately. Despite this flaw, less than 5% of the entries came up with this problem.

Due to limited programming ability, many features couldn’t be added to the program. One feature that could help aggregate more articles at a time would be the ability to search through all journals in the program. This way, the SQL table would be able to have more information to rank and more results for the end-user to consider. This feature will be worked on.

**CONCLUSION**

ChemFoParser can go down many paths. With RSS news aggregation at it’s core, it promises to be a program that retrieves the most current and relevant scientific data. ChemFoParser has many avenues that it can go down.

In it’s current form, ChemFoParser is limited to searching through author names, journal article titles, and published dates. This is primarily due to Feedparser, but with different programming functions, the type of information that could be retrieved could be immense.

RSS feeds are powerful summaries to pull information from. By having the most current data all the time, this program stays relevant in the realm of informatics. With continual development, it can be a powerful personal news aggregator.

**REFERENCES**

"MySQLdb User's Guide." *MySQLdb User's Guide*. N.p., n.d. Web. 25 Apr. 2014.

Pilgrim, Mark. *Universal Feed Parser*. Program documentation. *Feedparser Basic Features*. Vers. 5.1.3. N.p., 2004. Web. 25 Apr. 2014.

"RSS Feeds." *What Is RSS: A Tutorial Introduction to Feeds and Aggregators*. N.p., n.d. Web. 25 Apr. 2014.

SCImago. "Journal Rankings in Chemistry." *SJR — SCImago Journal & Country Rank*. N.p., n.d. Web. 25 Apr. 2014.