A Data Mining Framework for Robust Web Scraping and Plug-and-Play Functionality
Earl Lee
Advised by Ruzica Piskac
Yale University
CPSC 490
In the past decade, we have seen exponential growth of digitized information. As more people gain access to the Internet, and Internet usage increases, transcription of the world’s knowledge onto the web increases in parallel. The Internet serves as the best platform for information because it makes data is easily replicable and accessible. Though not all information on the web is completely digitized, information that is allows for more sophisticated analysis using programming and modern computation techniques. One study, conducted by the International Data Corporation, cites doubling of information on the Internet at every two years [1]. Now more than ever, there are large data sets available to the public on the Internet.

Open data is one such type of data available on the Internet. Open data is any kind of data set that is freely and easily accessible and also machine readable. It is published by governments, corporations, non-profit organizations, and more. Since it is machine readable and generally packaged nicely, open data is prime for immediate utilization. According to a report published by McKinsey & Company, open data utilized in just seven major sectors—education, transportation, consumer products, consumer finance, electrical, oil and gas, and healthcare—can generate up to $3 trillion in value annually [2]. However, open data is only a small fraction of all the information available on the web.

There are also large data sets of in plain view of the average Internet user. Public tweets, Facebook posts, and other content created on social media sites offer insight into the mindset of the public. Lastly, there is private data, such as those collected by credit card companies on consumer purchases or by search engines on browsing history.

The aggregation of different types of data sets generally offers more advantages and insight than any single data set itself. For example, stock prices of large retail corporations combined with credit card transactions may yield foresight into what stocks will rise in price based on patterns found in credit card transactions for various public companies. In fact, the SEC is currently investigating fraud by former Capital One employees who did just that. Bonan Huang and Nan Huang allegedly queried Capital One’s database of credit card transactions to project earnings by the fast casual giant Chipotle [3]. The pair extrapolated Chipotle sales volume from transaction data, which allowed the them to predict earnings calls and purchase options appropriately before earnings calls became public—a form of insider trading. The duo net 1819% return on their investing during a three-year span.

Washington D.C.-based startup FiscalNote leverages legislative data to predict the chances a bill will pass into law. FiscalNote aggregates information such as bill text, bill metadata, legislator voting history, legislator metadata, committee legislator compositions, and more. Patterns found in historical and current data across a variety of data sets are then used in machine learning algorithms to produce predictive analytics. The company has demonstrated up to 94% accuracy rate on its predictions, convincing corporate giants such as JPMorgan Chase & Co. and Uber to purchase subscriptions to their legislative data platform.\(^1\)

Though the Capital One researchers had access to a centralized database with structured data on credit card transactions, FiscalNote must first convert raw, unstructured on state and federal websites into structured data and create a database before running analytics on the data. This process requires a host of components including web scraping, entity matching, relational mapping, data standardization, and

\(^1\) Disclaimer: I am a current employee of FiscalNote, Inc. and have been since 2013.
database population. Furthermore, all this must be done in an online fashion, processing data continuously as it becomes available.

PROPOSAL

The success of the Capital One employees and FiscalNote in leveraging large data sets to gain competitive intelligence in finance and government relations present a convincing argument for the value of large data sets found on the Internet—colloquially known as big data. I plan to focus on the data mining aspect of the big data, since data analytics is more specialized depending on each scenario. Specifically, I hope to build a simple yet flexible framework for online aggregation and processing of data from a variety of different sources, following a plug-and-play model for adding data sources to the data mining infrastructure. I will call this framework DataFall. The framework should be flexible enough to allow quick implementation with different types of data ranging from financial data to healthcare data.\(^2\)

Currently, there exist many ways to access large data sets available on the Internet. These methods of accessing data can be considered inputs in a data mining framework. Listed below are a few:

- Public application programming interfaces (APIs) e.g. those announced and documented on websites and designed for use by the public
- Private APIs e.g. those utilized by websites themselves but not publicly announced or documented
- Direct web scraping
- Direct access to SQL or NoSQL databases
- File transfer protocol (FTP)

The purpose of DataFall is to consume continuously updating sets of raw data, process them by a user-defined schema, map relations between entities, and populate a database with the processed data. DataFall will focus on operation of web scrapers in a real-time fashion as well as scripts that query APIs for data. Essentially, users should be able to write web scrapers and API scripts and easily integrate them into the system as long as they output a predetermined format.

Furthermore, DataFall should allow for easy monitoring of the entire data processing pipeline and targeted detailed diagnosis if anything goes wrong along the way, so that fixes can be executed swiftly. Users should be able to see what data inputs are currently reading in new data, progress of entity relational mapping for freshly imported data, and progress of population of databases.

DataFall will accept a variety of different databases, using user-defined serializers to handle the injection of data. It also aims to be language agnostic with regards to scripts used for data input. Furthermore, DataFall should be able to ingest data as fast as possible from the time it is available on the web.

\(^2\) Though I was not directly involved in development of all parts of the data scraping framework at FiscalNote, from a top-level perspective, I am aware of its problems and ad-hoc development which has led to a less-than-optimal result. My past work has primarily been at the early stages in the pipeline, building scrapers and fashioning them as inputs into the pipeline. DataFall is partly an effort to build a completely new framework from ground up with issues related to online processing and fragile scrapers in mind. Lack of familiarity with the details of the current framework will allow me to independently develop a new framework from scratch without the pressure of commercial deadlines that may compromise quality over execution time. Unlike the framework used at FiscalNote, DataFall will not be tailored to any specific type of data and will be able to be customized to the user’s needs.
Because the APIs for data on a website is only used by a small fraction of total users of a website, most websites do not have APIs for information they present. Thus, web scraping remains an important tool in the data mining world. With this in mind, DataFall aims to introduce new tools for more robust web scrapers and promote certain methodologies that may yield more reliable scrapers. The problem with scrapers is that they fundamentally rely on xpaths or CSS paths to find nodes in the document object model (DOM) of a web page. Unfortunately, if any node between the root of the DOM tree and the xpath-targeted node changes, then the xpath may fail and must be re-written. Instead of relying heavily on xpaths and the DOM tree, we might look into utilizing textual analysis to find information on a page, since that is the way humans perceive an actual page.

**Figure 1. An overview of the data processing pipeline**

**DELIVERABLES**

- DataFall 0.10 - A simple data mining framework that accepts scrapers and API scripts for a schema with a few models and relations between models
- DataFall 0.20 - A more developed data mining framework that consumes data in real-time, as the information becomes available
- DataFall 0.25 - A view of activity by each individual data input and components of the data pipeline
- DataFall 0.30 - New methodologies and techniques for web scraping that may help mitigate damage caused to web scrapers by slight changes in page layout and design
- Paper summarizing DataFall and detailing methodologies for robust web scraping
REFERENCES

