Big Data Spring 2018

Stella Sun ss8955, Zhiwei Yu zy1129, Ksenia Saenko ks4841

Project Proposal: **Search Engine for Structured Data**

Previous work and references

Cafarella, M. J. et al. WebTables:Exploring the Power of Tables on the Web.

Dasu, T. et al. Mining Database Structure; Or, How to Build a Data Quality Browser.

Agrawal, S. et al. DBXplorer: A System for Keyword-Based Search over Relational Database.

Luo, Y. (n.d.). Spark:A Keyword Search Engine on Relational Databases.

Rejaraman, A., & Ullman, J. (n.d.). *Mining of Massive Datasets*.

Problem description and goal

Problem description: efficiency and quality of any data project depends on the ability to accurately and timely obtain the necessary information. The current search capabilities of Open Data NYC portal are limited to search over table titles. A user cannot search contents of tables, such as specifying a column name or keyword to retrieve from the rows. A user also cannot search by table properties, such as a number of rows in a table. The current interface is inconvenient for the user and results in slower workflow and inability to find all relevant tables.

Goal: We propose to build a search engine for structured data with improved query capabilities for the NYC Open Data portal https://opendata.cityofnewyork.us/

- Our search engine will allow users to query the contents of a large collection of data sets using structured queries.
- Users will be able to specify their search criteria. For example, they could ask for all data sets that contain a column called "taxi", or a value "1988" from a table with at least 1000 rows, or a column "address" that contains the string "village".
- To accomplish that, we will provide capabilities to search by 3 levels: table title, columns, and keyword search within the columns.

Datasets and methods

Dataset: subset from all categories of data in NYC Open Data

Method:

- Get the data: We will use Socrata API to download the json files in Spark
- Store the data: We will use the shared directory on HDFS to store the raw dataset
- Design schemas: describing table properties and relationships
- Indexing tables: create 2-dimensional index tables for column and content search using MapReduce

- Design user queries: After all table structures are completed, we use PySpark to create views from all tables, and write functions in python to search
- User Interface: We use python to prompt the user to enter keywords, and we call search functions to search in Spark, and print out the sorted results
- We'll use Github for version control while working on the code

Evaluation criteria

- Time it takes to search raw data vs. indexed data
- Increased query capabilities compared to NYC Open Data website
- Ability to perform the following query types:
    - Ask for all data sets that contain a column called "taxi"
    - or a value "1988" from a table with at least 1000 rows
    - or a column "address" that contains the string "village"