

Working with a Document Store

DS4300: Large-Scale Storage and Retrieval

Prof. Rachlin

DESCRIPTION

The purpose of this assignment is to become broadly familiar with the Mongo query language and its capabilities and using mongo programmatically. The best way to learn something is to try to teach it. For this exercise I want you to create a mongo query language tutorial that demonstrates a broad range of querying capabilities. For example, for the “zip codes” tutorial we did in class, we explored some of Mongo’s abilities for geospatial searching in addition aggregation queries.

INSTRUCTIONS

Step 1. Choose a dataset. Pick a JSON-formatted dataset of your own choosing. You can either start with a JSON-formatted dataset or you can retrieve data from an on-line web service using Python’s *requests* library. Pick a dataset that is interesting in some way. Try to pick something that has some social, economic, or scientific significance. The dataset should be reasonably feature-rich and contain aggregated (hierarchical) data to truly showcase the power and flexibility of Mongo’s document storage model (and not merely a flat collection of keys and values derived from a CSV file.)

Step 2. Import your data into Mongo. Import your data into a database collection (or collections). You may do this programmatically or manually using either the *mongoimport* command-line utility or *Compass* – the graphical user-interface for Mongo.

Step 3. Create a tutorial. Ask 10 questions about your data. For each question:

- a) Declare the question (in English)
- b) Provide the mongo query that answers the question
- c) Output the results of the query (or at most, say, the first 5-10 results).

The 10 questions can be invoked on the client or programmatically. If you are doing this exercise entirely programmatically, you might consider using Jupyter Notebooks to document both the query and the output. Your queries should be varied to show the broad capabilities of the Mongo Query Language. Include, for example, some aggregations, filtering, sorting, etc.

Step 4. Programmatic Visualization. For at least one of your questions, write a program to query Mongo programmatically. I recommend PyMongo for this purpose, but you can use any language and libraries you like. (PyMongo is nice because JSON documents are returned as lists of dictionaries that are very easy to process.) Generate a visualization / plot of your result and provide an interpretation of your output, i.e., what does the visualization reveal about your dataset? Try to highlight something non-obvious.

WHAT TO SUBMIT:

1. A copy of your JSON-formatted dataset from Step 1 (.zipped)
2. Questions, queries, and output that are the basis of your tutorial from Step 3. Please submit this in .PDF format – *not* a word-document.
3. Code and your visualization for Step 4. Please format your figure as a .pdf, .png, or .jpg file.