Elasticsearch Retrieval System

Samuel Scudere-Weiss

Brian Gzemski

Jiaqi Li

2/28/17

Description: This elasticsearch application is a search over a database of trees. It allows the user to search for keywords and certain properties of any tree.

Dependencies: This software requires Elasticsearch, flask, and python 2.7

Building system: After unzipping the files start Elasticsearch and then run trees_es.py to build the index for the search program.

Running the program: make sure Elasticsearch is running, then run search.py and then go to the link provided in the console.

Modules:

search_page.html

This is the landing page for the search program. It is modified from the example given to us and has a very simple interface to allow the user to search. Keywords in the textbox are going to match the content of tree name and description. There are radio buttons for other fields and each field is a specific property of the trees. The search allows for users to put in any known information about a tree and “unknown” option is provided in case the user doesn't know.

result_page.html

The result page lists the names for all the matching tree documents and the documents are ranked based on their relevancy. Each name is clickable and would redirect to its corresponding target page.

target_article.html

This page displays the chosen document. It is dynamically generated using the find_by_name method in trees_es.py. The document of that specific tree is retrieved from the JSON corpus and displayed in a friendly format each field of the tree.

trees_es.py

This file builds the index for the retrieval system to use. The index is made using Elasticsearch search.py

This is the file that runs for searches to take place. It has a module for Elasticsearch and then a module for the flask interface

This program was tested using individual json from the whole corpus in order to test its basic functions on a manageable scale. Testing was also done using the Kibana elasticsearch interface to manage the data.

My Contribution:

I created all the three html files and the flask part in the trees_es.py. I handled the passing of parameters from html to python file, and also built the methods elasticsearch_search(), search() and result() the handles the search and converts them into the correct format for display, as well as the find_article() method that search the JSON corpus (not the elasticsearch one) by the name (“title” field) for a tree.