Title: Assignment 5: Learning Elasticsearch

Author: Zhiheng Wang

Date: 4/14/2019

Description: Build a search engine on 2018 movies data with Elasticsearch and flask.

Dependencies:

Python 3.6.5

Flask (http://flask.pocoo.org/)

Elasticsearch (https://www.elastic.co/downloads/elasticsearch)
Elasticsearch (https://pypi.org/project/elasticsearch/)

Elasticsearch-dsl (https://elasticsearch-dsl.readthedocs.io/en/latest/)

Build Instructions: Install these packages in any sequences.

Run Instructions:

Run the elasticsearch server in the background

index.py: building an inverted index for the database

query.py: calling the search engine.

Modules:

class Movie(): Define document mapping (schema) by defining a class as a subclass of Document.

test_analyzer(): For testing analyzer

buildIndex(): buildIndex creates a new film index, deleting any existing index of the same name. It loads a json file containing the movie corpus and does bulk loading using a generator function.

results(): show result pages

documents(): display a particular document given a result number

Testing:

Top 3 search results for Search Text: crime drama "philip roth", with min runtime 130:

Drama, score: 10.024916

Abrahaminte Santhathikal, score: 9.678776

My Brother's Name Is Robert and He Is an Idiot, score: 8.974666

Tokenization:

Elasticsearch standard tokenizer for text search, and whitespace tokenizer for others.

Text Normalization:

Porter stemmer, lowercase, asciifolding for text. Lowercase for others.

Test Queries Examples:

Data: two corpus files (test_corpus.json, 2018_movies.json).

Time: indexing time less than a second