Student: Duc Ho

ID: 8944025003

**STEPS TO A SUCCESSFUL RUN**

\*\*\*DEPENDENCIES\*\*\*

-- **MacOSX/Linux** Environment with **Bash** support. Windows might need to install Cygwin[ NOT TESTED ]

-- **BeautifulSoup4**

pip install beautifulsoup4

More:<https://www.crummy.com/software/BeautifulSoup/bs4/doc/#installing-beautiful-soup>

-- **Flask**

pip install Flask

More: <http://flask.pocoo.org>

-- **Solr 6.2.1**

Download: <http://www.apache.org/dyn/closer.lua/lucene/solr/6.2.1>

\*\*\*Preliminary Setup\*\*\*

-- Copy solr-6.2.1 folder into the current folder[ where you found README]. This is the base of our execution.

-- Copy crawled data htmls into CrawledData folder

-- Copy mapping csvs[LATimes/HuffingtonPost] into config folder [Csvs are supplied with the submission]

\*\*\*Start\*\*\*

**./run.sh -start**

( Might need to chmod+x run.sh before running )

\*\*\*Code path/Justification\*\*\*

( Or explanation of the bash’s underpinning)

-- start solr > create\_core [core] > copy schema & config file > reload > run page rank and generate patch link json > copy external\_pagerank file > start indexing > patch link field with json > run web server

-- **run\_conf.properties** contains core name to create. By default, core name is “**srcore**”

**-- config folder contents:**

**-- mapHuffingtonPostDataFile.csv, mapLATimesDataFile: inputs to page\_rank.py**

**-- managed-schema :** a default schema file with added fields to store the link to website and pagerank score**:**

**<field name="**link**" type="**strings**" indexed="**true**" stored="**true**" />**

**<fieldType name="**external**" keyField="**id**" defVal="**0**" class="**solr.ExternalFileField**" valType="**pfloat**" />**

**<field name="**pageRankFile**" type="**external**" stored="**false**" indexed="**false**" />**

**-- solrconfig.xml :** a default solrconfig with added config for externalfilereader listener on field pagerankFile of type external

**--** Core is first created, schema and config files are then copied over to core folder. Solr is reloaded/restarted for the config to be in effect.

-- Solr is unable to retrieve the original url from the downloaded htmls ( the field “og\_url” is sometimes null ) which is required by the specs ( link clickable to the real site ). Luckily, these urls are provided with the **mapping csvs**. **page\_rank.py** generates these links in json (<https://cwiki.apache.org/confluence/display/solr/Uploading+Data+with+Index+Handlers#UploadingDatawithIndexHandlers-JSONFormattedIndexUpdates> ) along with computing the page rank – why wasting time generates edge/adjacent list when it could be done here ?. Pagerank parameter ***alpha=0.85*, *personalization=None*, *max\_iter=30*, *tol=1e-06*,*nstart=None*,*weight='weight'*,*dangling=None.*** *The outcomes****{pagerank.txt, patch\_link.json****} are generated in the folder* ***patch.*** *pagerank.txt is copied over to the core data folder.*

-- solr is then instructed to index the downloaded url, then is patched with **patch\_link.json** so that link field contains the original urls.

-- Flask webserver is run

-- localhost:5000/search 🡪 normal search result

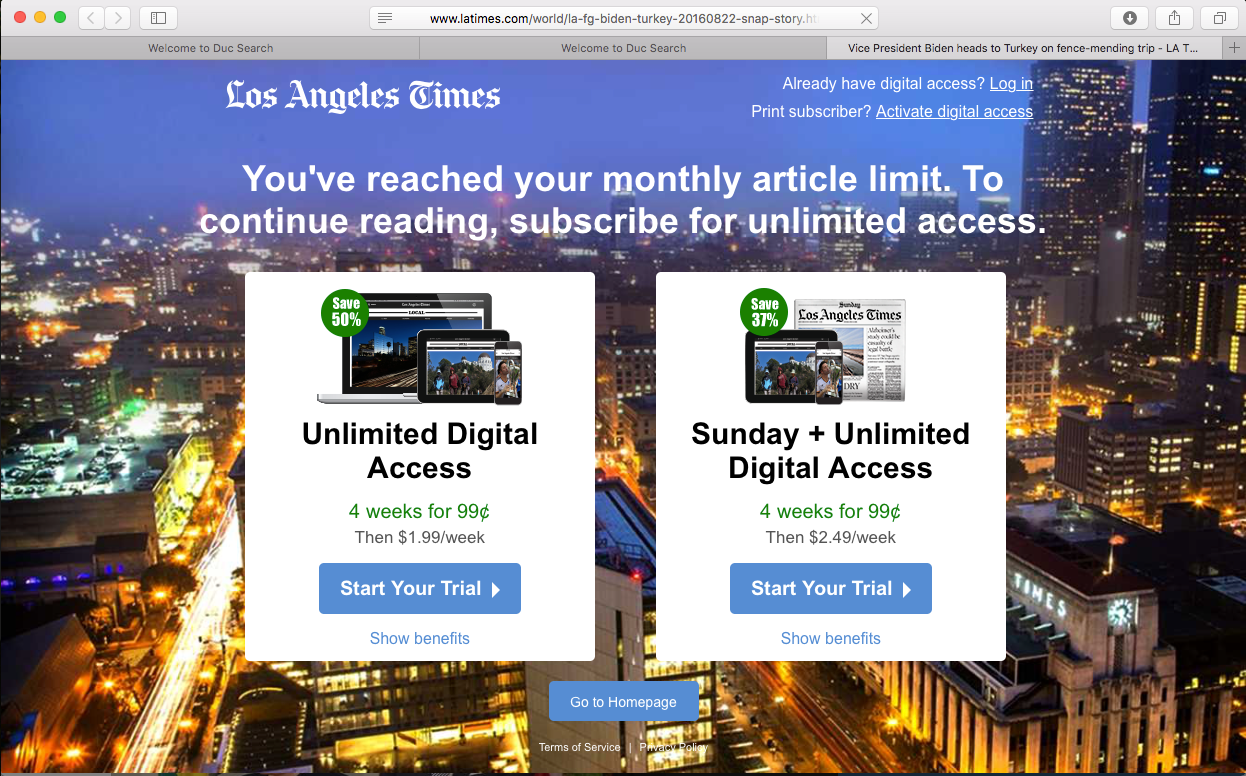
--localhost:5000/search\_pr 🡪 page rank search result

\*\*\*Report\*\*\*

-- Report folder contains files for submissions:

--**Screenshots**: pictures of each query type by **default** and **page rank** strategy. CAVEAT: Sites like LATimes or Huffington post limits the number of free access so taking screenshot of the site

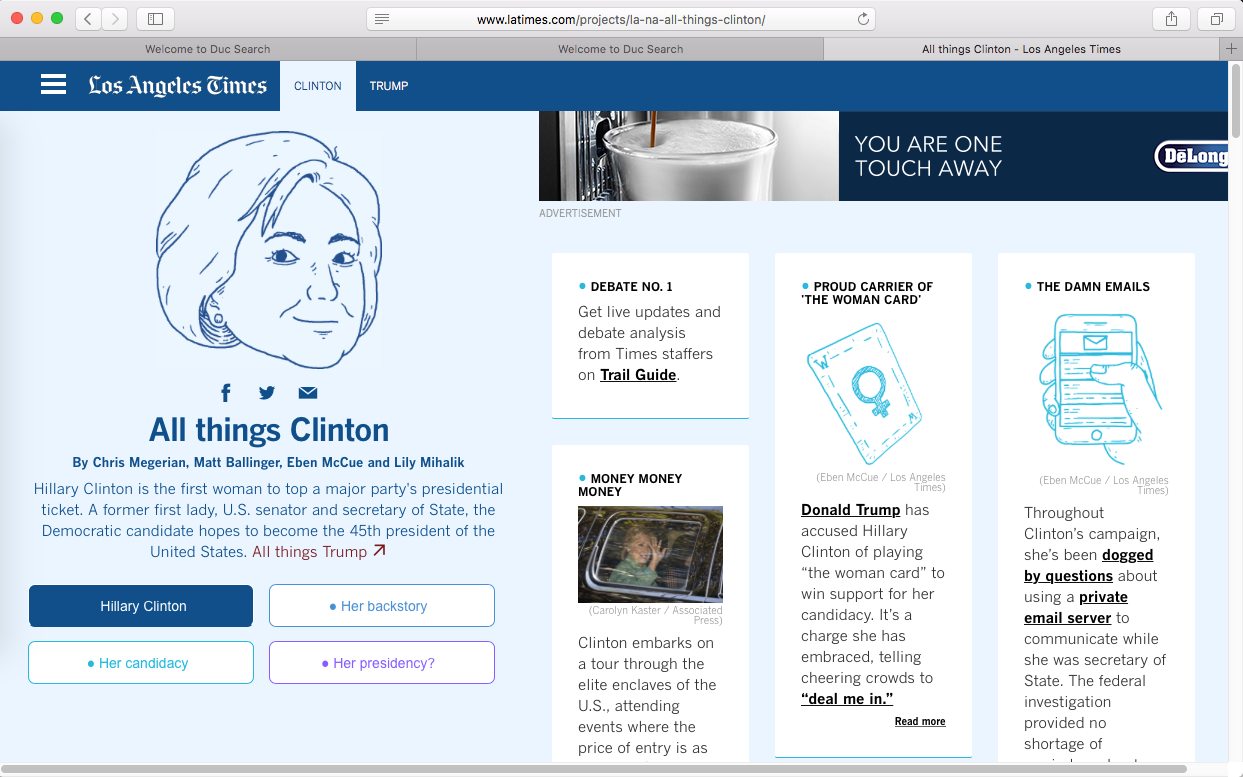
is sometimes really difficult; was greeted by some things of the like:



What I did that helps is copy the link from my localhost site and paste it into google.com to indirectly access and screenshot.

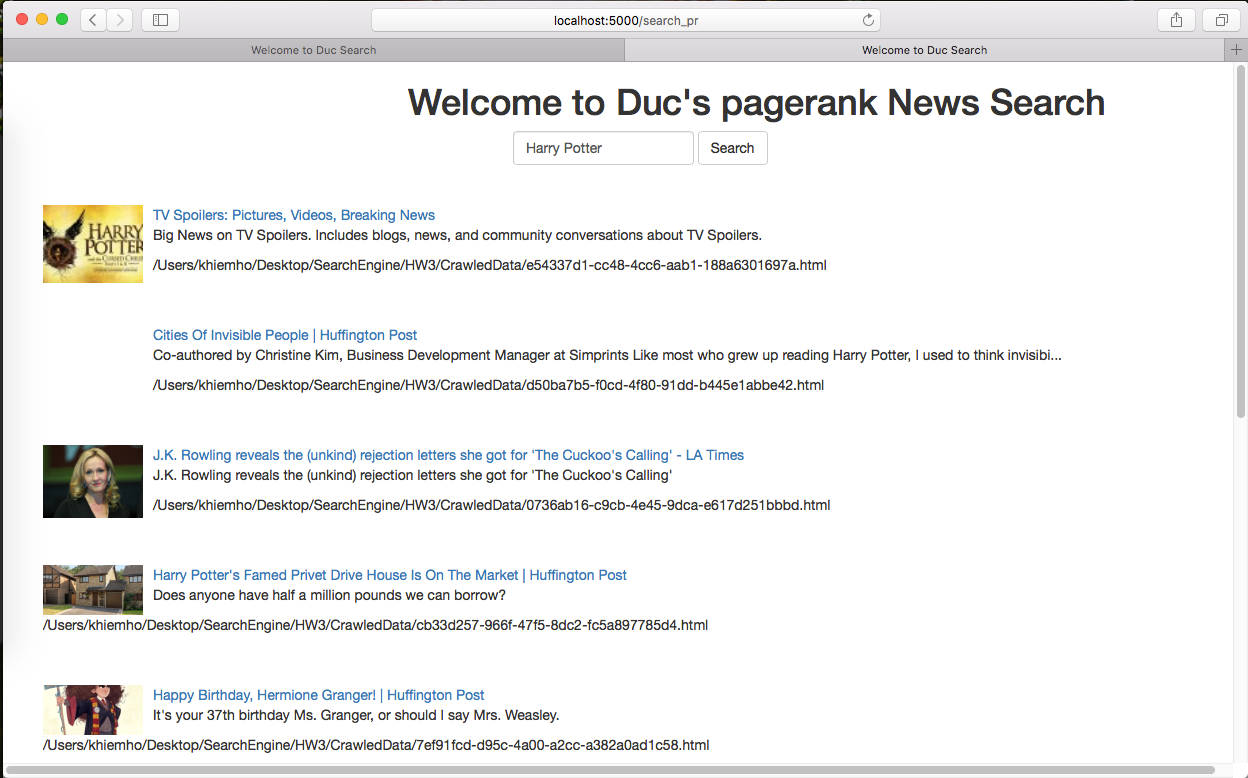
--**top\_ten\_results\_by\_category.xlsx**: excel files with query id and graph of overlaps. Some links are better than others simply because it has more outlinks existing in our mapping csv, not necessarily it contains the right content we need to see. Pagerank seems to be doing good with one-gram keyword, three-gram keywords seems to be hopelessly incorrect ( maybe needs to tweak those parameters passed to NetworkX method ? )

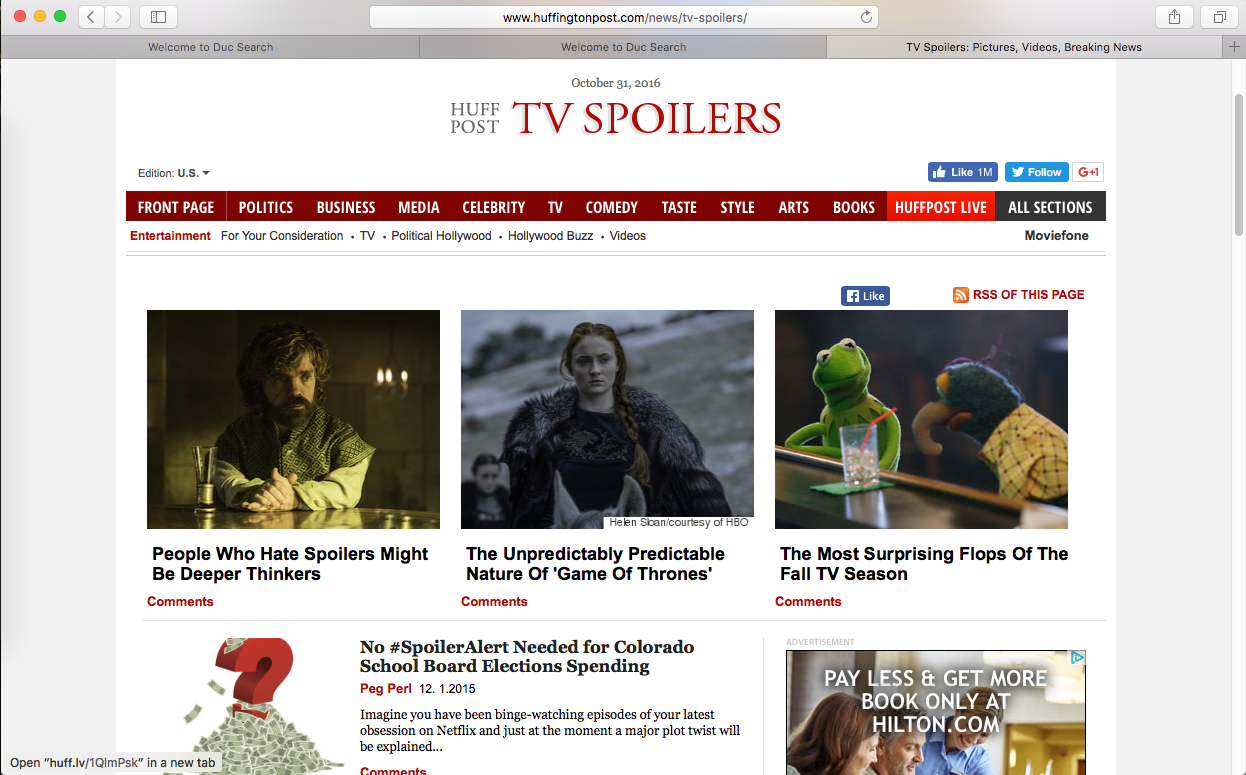
A search for “Donald Trump” return “Hillary Clinton” as president



Indexing needs running on a daily basis, otherwise outdated information might be happening. For example, top story returns harry porter image:

But got Game Of throne instead ( now it is top but crawled data is old ):





Anyway……! Have a good day