Homework Assignment 5

I have used same environment for assignment 5 as I have used in the assignment 4. Following is the environment setup.

Environment Setup:

1. Installed ubuntu operating system and all the required software.
2. Downloaded apache solr from http://www.apache.org/dyn/closer.lua/lucene/solr/7.5.0 and installed it.
3. Apache Solr is launched using the command bin/solr.cmd start. Verified the installation of Solr from browser using http://localhost:8983/solr
4. Created a core called - **irassignment4_version1** Using bin/solr create -c **irassignment4_version1.**
5. Given my usc-id, I am working on LA times. Downloaded webpage dump from drive and extracted it.
6. Made field changes in managed-schema and uploaded the data that needs to be indexed. A total of 16,826 files are indexed. Commanded used is - bin/post -c **irassignment4_version1** <path to html folder>
7. At the end of this step, we have environment with apache solr installed and LA times webpages are indexed.

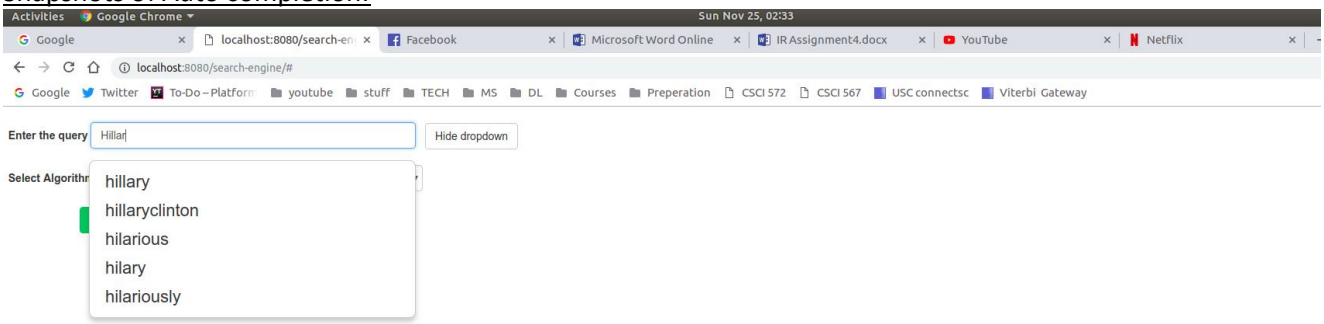
Auto-complete:

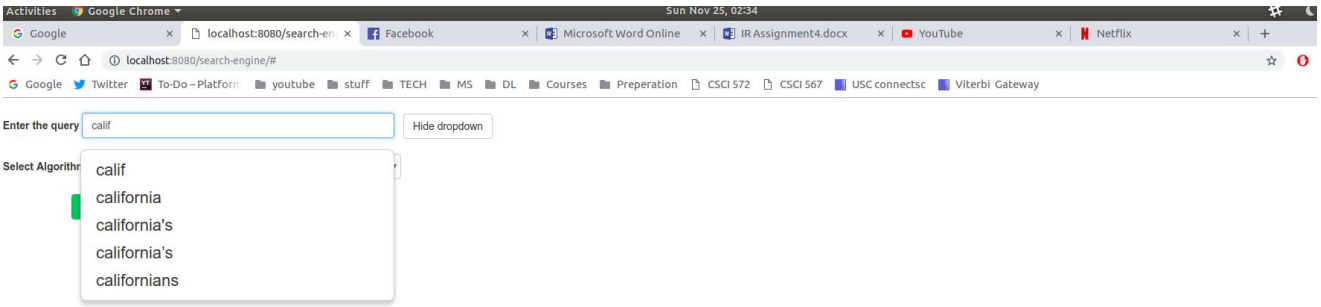
To create auto complete functionality used **java** rest api's are used. As soon as there is change detected in the input box(using angular ng-change), /suggest rest api is invoked. through rest calls.

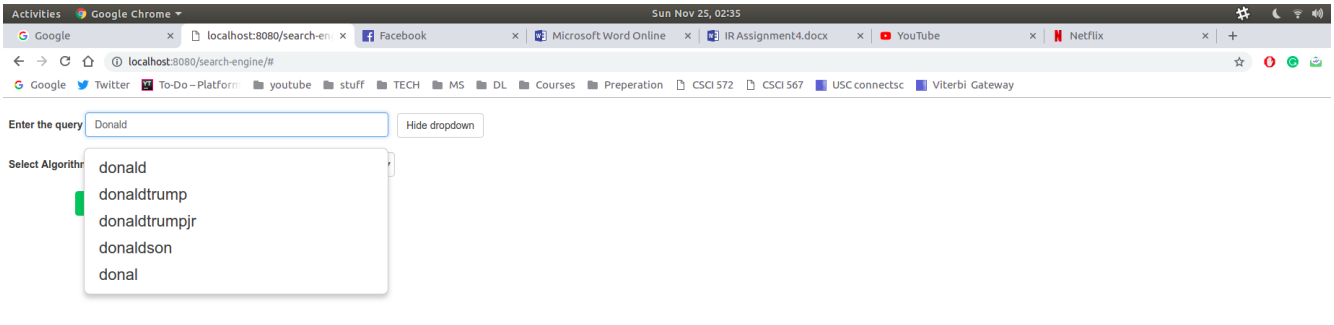
Following changes are made in solr-config.xml to enable suggestions in apache solr.

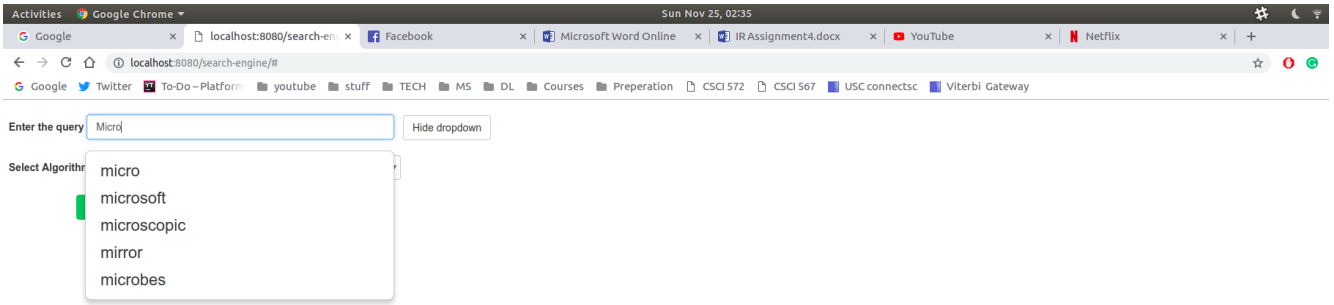
```
<searchComponent class="solr.SuggestComponent" name="suggest">
<lst name="suggester">
    <str name="name">suggest</str>
    <str name="lookupImpl">FuzzyLookupFactory</str>
    <str name="field">_text_</str>
    <str name="suggestAnalyzerFieldType">string</str>
  </lst>
 </searchComponent>

<requestHandler class="solr.SearchHandler" name="/suggest">
  <lst name="defaults">
    <str name="suggest">true</str>
    <str name="suggest.count">5</str>
    <str name="suggest.dictionary">suggest</str>
  </lst>
  <arr name="components">
    <str>suggest</str>
  </arr>
 </requestHandler>
```

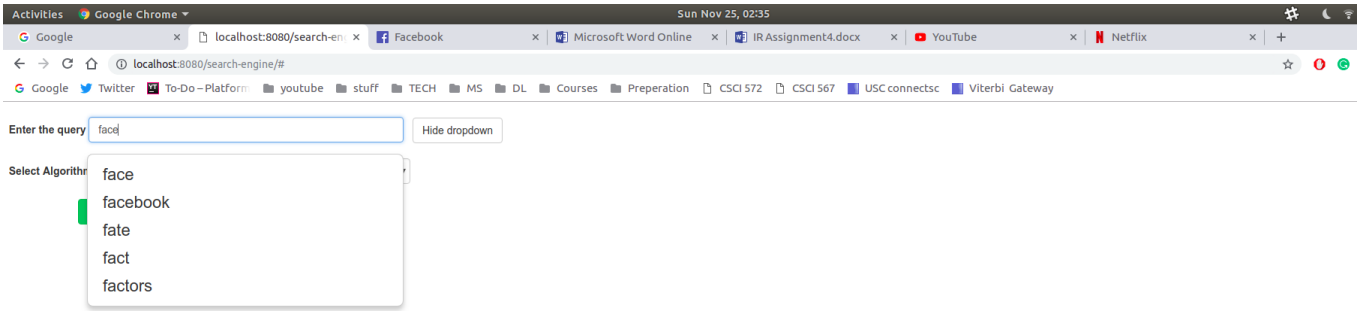
Snapshots of Auto completion:







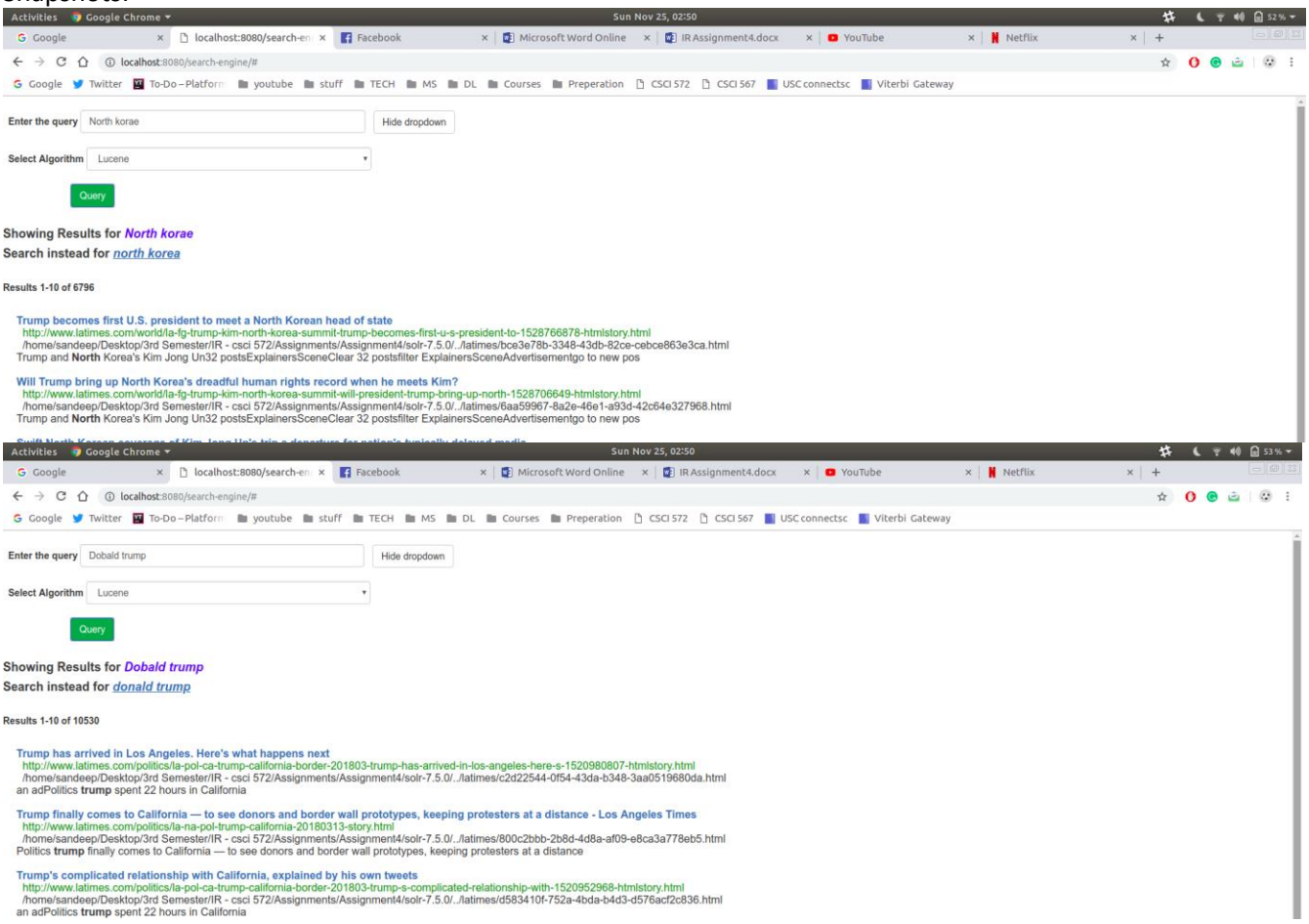


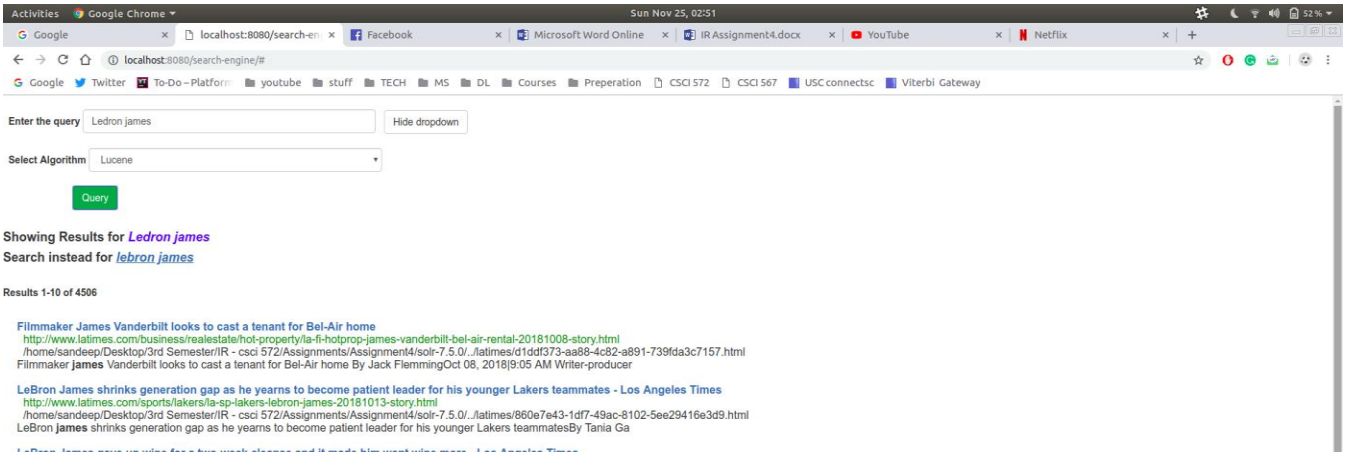


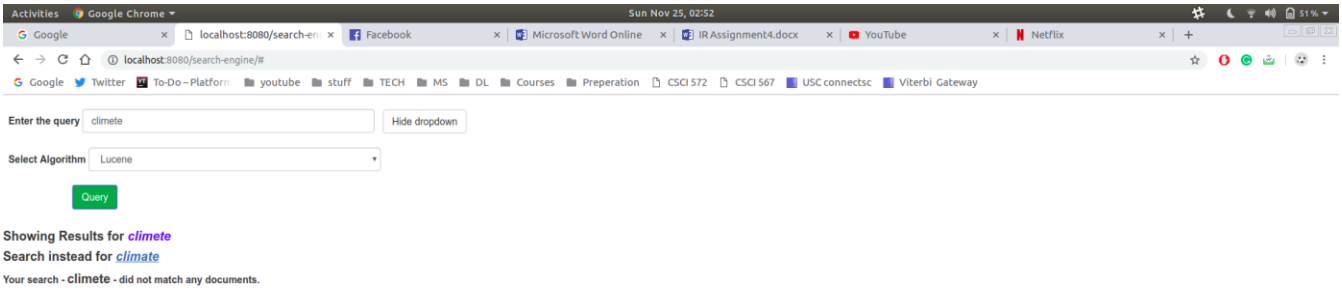
Misspelled Term correction:

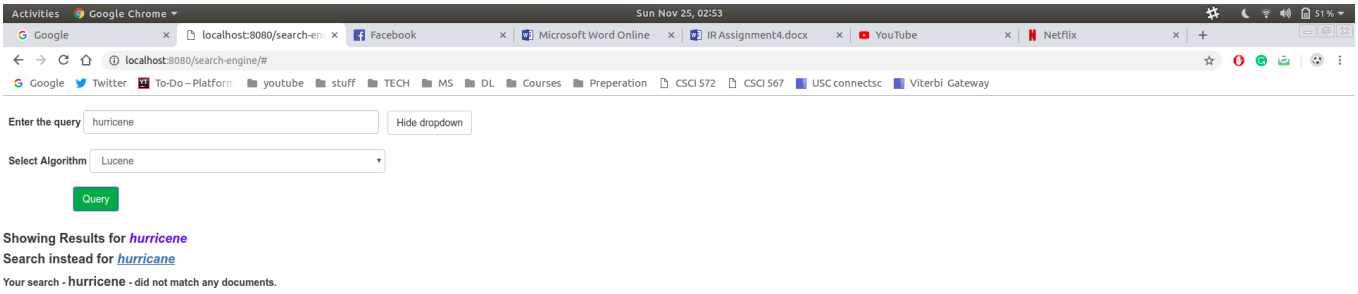
1. Used Norvig's Spell Correction Java Version program.
2. Generate big.txt using Apache Tika and iterating over all the html files of LA times.
3. Invoked the spell correction from the UI when user request for search, showed the suggestion and when suggestion is clicked, corresponding results are showed.

Snapshots:









Steps to run the code:

Front-end part of the application built using AngularJS, HTML and CSS and Bootstrap.

Backend part of the application is built using java and communication to the front-end is over rest api.

The code is written in java plus angular js and maintained through maven. So, for the ease I have created a .war file called search-engine.war

Download apache-tomee-plus-8.0.0-M1 and copy search-engine.war to the Webapps folder.

Make sure that apache tomee http is running on **8080** port (check in conf/server.xml)

Run **bin/catalina.sh run** or **bin/startup.sh**

Go to any browser and run http://localhost:8080/search-engine

It pops with the search form, you can give queries and check the results.

PRECAUTIONS:

Make sure the folder where you copied apache tomee won't have any spaces in its absolute path