**ENHANCING YOUR SERACH ENGINE**

1. **Steps you followed to complete this assignment. Include the details of what tools and techniques you used to implement spelling correction and autocomplete.**

**Spelling Correction:**

* For spell correction, I made use of the Peter Norvig spell correction program written in Java.
* I created a big.txt file by extracting the content from html web pages of New York Times by the following Apache Tika code(sample).

BodyContentHandler handler = **new** BodyContentHandler(-1);

Metadata metadata = **new** Metadata();

FileInputStream inputstream = **new** FileInputStream(**new** File(path));

ParseContext pcontext = **new** ParseContext();

//Html parser

HtmlParser htmlparser = **new** HtmlParser();

htmlparser.parse(inputstream, handler, metadata,pcontext);

System.***out***.println("Contents of the document:" + handler.toString());

String data=handler.toString();

String con=data.replace("\n", "");

fw.write(fi.trim() +" ");

* I gave the big txt generated by this java code as input to the Norvig spell corrector program written in PHP.
* The Norvig program gave the corrected word by calculating the edit distance 1 or 2.
* I used JavaScript as client side. I called the spell corrector PHP program using AJAX call to the spell corrector program whenever the user clicks submit button.
* Then I make an api call (using ajax) with the call to solr with the corrected word and displayed the results for the corrected word. I asked the user if he wants to see results for the misspelled word.
* If the user wanted to see results for misspelled words I made an API call (using ajax) to solr with the misspelled word as the query in the api url and displayed the results of the user clicks on the text “Do you want to display results for … instead?”
* I also displayed a message "Showing results for 'corrected query' " with the corrected spelling.

**Spell Suggestions:**

* For spell suggestions, I got the results by using the autocomplete functionality of Solr using the FuzzyLookupFactory feature of Solr/Lucene.
* For this to work, I made a few changes in the solrconfig.xml file present in the conf folder of core.

<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>

<str name="suggest.dictionary">suggest</str>

</arr>

</requestHandler

* I then went to solr and in the query, I changed the request handler to /suggest component and typed the query to check for suggestions and I got the suggestions.
* I used this suggestion api for the query by making an ajax call to PHP server which makes an api call for getting the suggestions for the input entered by the user.
* The returned results were set back to JavaScript and was parsed and displayed using the autocomplete jQuery plugin.
* So, this event was triggered after each character entered by the user.
* I also returned the results for the query on the go, that is changed the results as the user types in the query since I made the api call for displaying the results whenever the key is pressed in the search box. This is similar to how google displays the results.

I also made the following changes to display results for multi term queries. By default, solr uses OR operator. So, I changed it such that it AND's the query terms and produces result containing all the query terms from the search box.

i.e, <str name="q.op">AND</str>

within this tag:

<requestHandler name="/select" class="solr.SearchHandler">

and Inside <lst> default tag within requestHandler tag:

 <lst name="defaults"

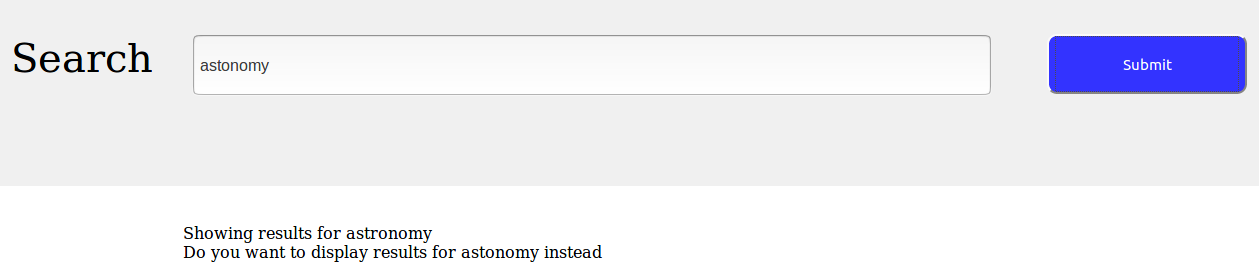
**Snippet Generation**:

* I checked the content and got the line matching the query. If I didn’t find then I returned the meta description from json results if any of the query term is matched.

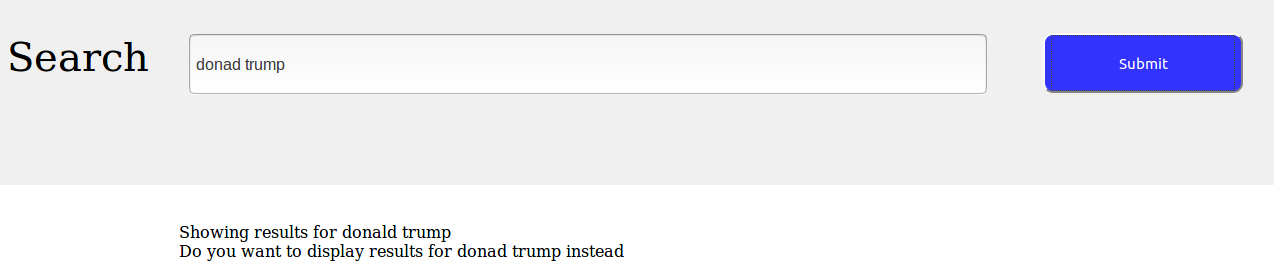
**2. Analysis of the results: In this you should provide FIVE examples of misspelled terms that are correctly handled by your spelling correction program. You should also provide FIVE examples of auto-completion.**

Spell Correction:

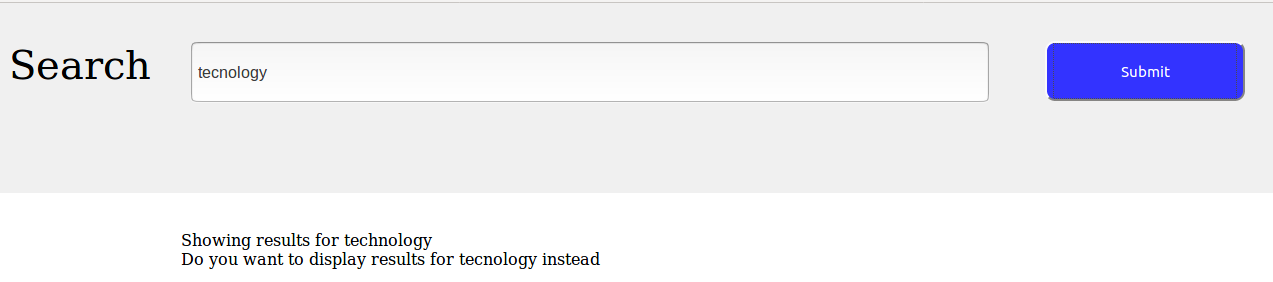
Query 1: astonomy for astronomy



Query 2: donad tump for donald trump



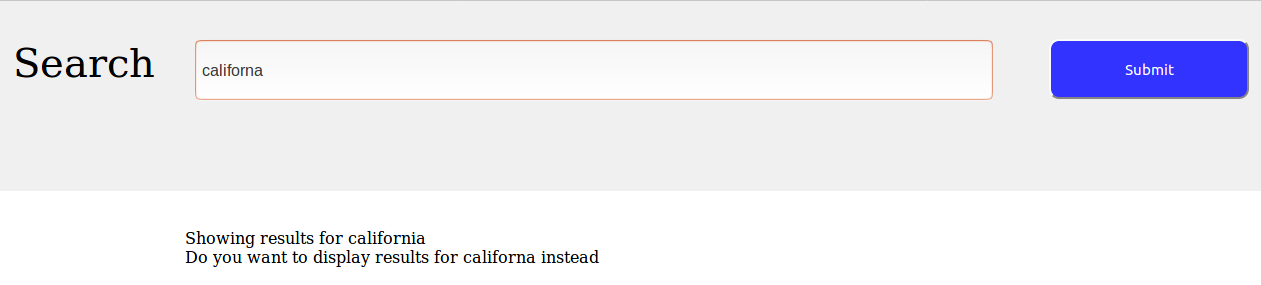
Query 3: technology for technology



Query 4: snapcha for snapchat

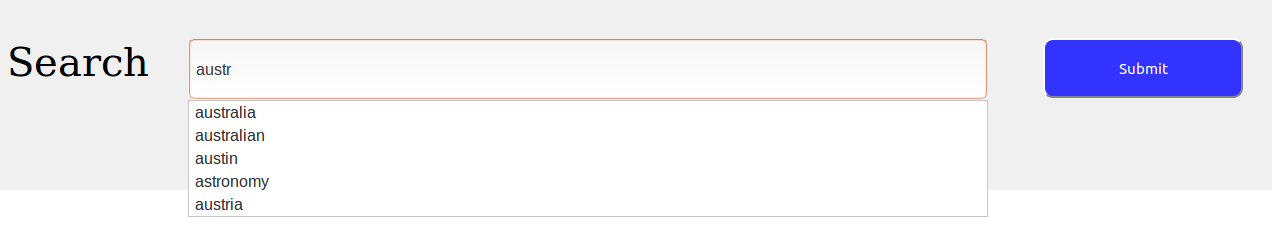


Query 5: californa

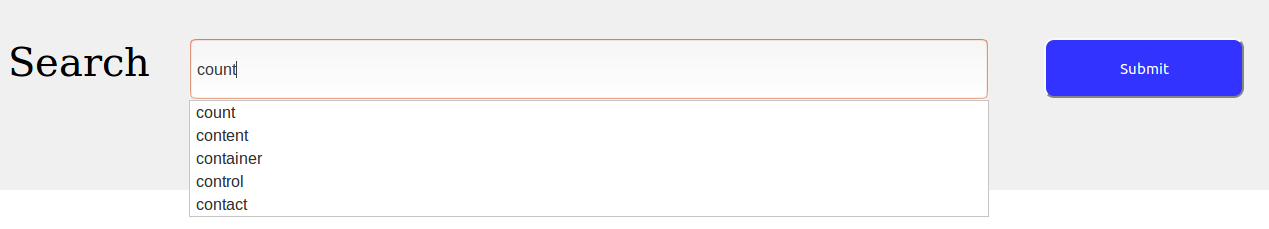


Spell Suggestions:

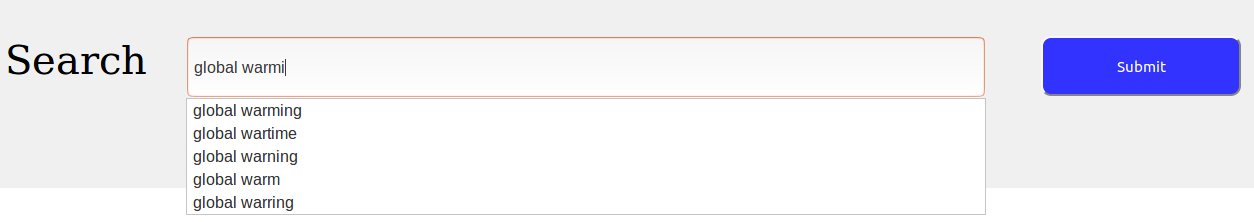
Query1: austra for australia



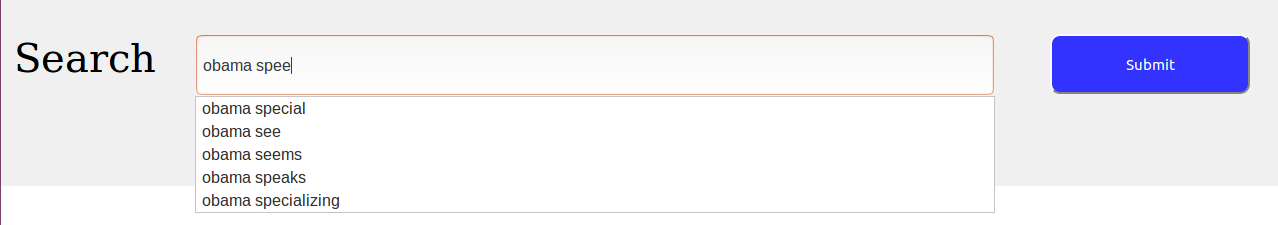
Query2: count for country



Query 3: global warmi for global warming



Query 4: Obama spee for Obama speech



Query 5: act for actress

