HW #5: ADDING SPELL CORRECT, AUTO COMPLETE AND SNIPPETS TO SEARCH ENGINE

STEPS TAKEN TO EXECUTE THIS HOMEWORK

I. SPELL CORRECTION

To implement this spell correct feature, Peter Norvig's spell correction program in PHP was downloaded and used. Peter Norvig's program, 'Spell_Corrector.php' was include in my file (index.php). This file was downloaded from https://www.phpclasses.org/package/4859-PHP-Suggest-corrected-spelling-text-in-pure-PHP.html#download after registration.

In order to build the dictionary i.e. big.txt dictionary specific to my news website (Boston Globe) used by Peter Norvig's spell correct code to calculate edit distance, Apache Tika was used. The Apache Tike JAR i.e. tika-app-1.16.jar was downloaded.

A Java program was written to construct this big.txt specific to Boston Globe news. To this Java project the Apache Tika JAR previously downloaded was imported.

In Homework 4, the Boston Globe dataset comprising of the corpus of web pages for Boston Globe (BG folder) and the CSV map file (Boston_Global_Map.csv) was downloaded from Google Drive link (https://drive.google.com/open?id=0B7BKTnqBZau-aWl1TTZ0NjBiRTA). The Java program essentially used Apache Tika to construct a dictionary from all the HTML files stored in this dataset.

A screenshot of the Java program used to generate big.txt using Apache Tika is attached here:

```java
import java.io.File;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.IOException;

import org.apache.tika.exception.TikaException;
import org.apache.tika.metadata.Metadata;
import org.apache.tika.parser.ParseContext;
import org.apache.tika.parser.html.HtmlParser;
import org.apache.tika.sax.BodyContentHandler;
import org.xml.sax.SAXException;

public class BigTextGenerator {

   public static void main(final String[] args) throws IOException,SAXException, TikaException {

      FileWriter big_text = new FileWriter("/Users/nehapathapati/Desktop/BigText_BG.txt");
      File dir = new File("/Users/nehapathapati/Desktop/CSCI 572 - IR/Homework 4/BG/BG/");

      for(File file : dir.listFiles()) {

         FileInputStream inputstream = new FileInputStream(file);

         BodyContentHandler handler = new BodyContentHandler(-1);
         Metadata metadata = new Metadata();
         ParseContext pcontext = new ParseContext();

         HtmlParser htmlparser = new HtmlParser();
         htmlparser.parse(inputstream, handler, metadata, pcontext);
         String document_content = handler.toString().trim().replaceAll("\\s+", " ");

         big_text.append(document_content + " ");
      }
      big_text.close();
   }
}
```

The big.txt file generated is approximately 133KB in size and was used as the dictionary for Peter Norvig's spell correct program.

A preview of the big.txt file generated for Boston Globe is as follows:

file:///Users/nehapathapati/Sites/BigText_BG.txt

tactful way to put it — for almost three decades. So why is the king only now being chased from his throne? Continue reading » Television review PBS's 'The Collection' is dressed for excess Set in the fashion capital of 1947 Paris, the "Masterpiece" series is hobbled by too many subplots. Continue reading » Critic's Notebook In Springfield, Dr. Seuss is at center of a cultural clash An image created by Theodor Geisel years ago is resonating in an unintended — and to some, an unpleasant — way today. Continue reading » The real people behind the 'Spotlight' characters Here's a list of the Globe staff members that appear in the movie, and the actors and actresses who play them. Continue reading » Music Ruby Rose Fox had to go deep to find her voice Even at an early age, singing was her passion and — after a family tragedy — a form of therapy. Continue reading » the story behind the book | kate tuttle A super reader becomes a novel writer Nancy Pearl has made a career out of advocating for other people's books. Continue reading » Movie Review 'Loving Vincent' gets style points for Van Gogh Artists diligently handpainted each frame based on film material created using a combination of live actors and digital animation. Continue reading » HOME CONTACT PRIVACY POLICY SUBSCRIBE © 2017 Boston Globe Media Partners, LLC.. Menu North Metro Sports Business & Tech Opinion Politics Lifestyle Arts Cars Real Estate Most popular on BostonGlobe.com Based on what you've read recently, you might be interested in theses stories Report: Matt Damon helped kill earlier New York Times story about Harvey Weinstein Dan Shaughnessy: Red Sox stars did not deliver when it mattered most Homeowner's deck nightmare Suspect in custody after fatal shooting of Texas Tech officer Today's Paper Magazine Obituaries Weather Comics Crossword The Big Picture Digital Access 99 cents a week for the first 4 weeks Subscribe Subscribe Home Delivery Save 50% off the regular rate Subscribe Subscribe Already a subscriber? Members Sign In Digital Access 99 cents a week for the first 4 weeks Subscribe Subscribe Home Delivery Save 50% off the regular rate Subscribe Subscribe Already a subscriber? Members Sign In BURLINGTON Open wide E-Mail Share via e-mail To Add a message Your e-mail Facebook Twitter Google+ LinkedIn Comments Print The Boston Globe Tweet Share George Weinstein photo Dr. Daniela Toro of Burlington Orthodontics with a game she helped design at the Celebrate Burlington Festival, held on the Burlington Common Aug. 5. The game involved kids of all ages using blow guns designed to shoot a pellet through the "Open Wide" opening in a photo of Dr. Toro, an alligator, a frog, or members of her office staff to win a prize. August 11, 2017 Loading comments... Top 10 Trending Articles Most Viewed Most Commented Most Shared Real journalists. Real journalism. Subscribe to The Boston Globe today. My Account Manage my Account Mobile Customer Service Sign Up For Newsletters Contact Help FAQs Globe newsroom Advertise Social Facebook Twitter Google+ More ePaper News in Education Archives Privacy policy Terms of service Terms of purchase Work at Boston Globe Media © 2017 Boston Globe Media Partners, LLC Menu High schools Metro Sports Business & Tech Opinion Politics Lifestyle Arts Cars Real Estate Most popular on BostonGlobe.com Based on what you've read recently, you might be interested in theses stories Report: Matt Damon helped kill earlier New York Times story about Harvey Weinstein Dan Shaughnessy: Red Sox stars did not deliver when it mattered most Homeowner's deck nightmare Suspect in custody after fatal shooting of Texas Tech officer Today's Paper Magazine Obituaries Weather Comics Crossword The Big Picture Digital Access 99 cents a week for the first 4 weeks Subscribe Subscribe Home Delivery Save 50% off the regular rate Subscribe Subscribe Already a subscriber? Members Sign In Friday's high school football scores E-Mail Share via e-mail To Add a message Your e-mail Facebook Twitter Google+ LinkedIn Comments Print The Boston Globe Tweet Share Matthew J. Lee/Globe staff By Globe Staff October 07, 2017 Loading comments... Top 10 Trending Articles Most Viewed Most Commented Most Shared Real journalists. Real journalism. Subscribe to The Boston Globe today. My Account Manage my Account Mobile Customer Service Sign Up For Newsletters Contact Help FAQs Globe newsroom Advertise Social Facebook Twitter Google+ More ePaper News in Education Archives Privacy policy Terms of service Terms of purchase Work at Boston Globe Media © 2017 Boston Globe Media Partners, LLC Menu Ideas Metro Sports Business & Tech Opinion Politics Lifestyle Arts Cars Real Estate Most popular on BostonGlobe.com Based on what you've read recently, you might be interested in theses stories Report: Matt Damon helped kill earlier New York Times story about

II. AUTOCOMPLETE

Solr's default autosuggest feature, namely the 'SuggestComponent' was utilized to implement this feature. To enable this feature, the following changes were made to solrconfig.xml.

In the search component 'suggest', the following parameters are set:

- The parameter 'lookupImpl' is set to 'FuzzyLookupFactory' in order to use Levenshtein distance to calculate edit distances for suggesting words.
- The parameter 'suggestAnalyzerFieldType' is set to 'string', which indicates the field type for query suggestions.

A screenshot of changes made to Suggest Component in solrcofig.xml is as follows:

```
879
880     <!-- Auto Suggest -->
881   <searchComponent name="suggest" class="solr.SuggestComponent">
882       <lst name="suggester">
883           <str name="name">suggest</str>
884           <str name="lookupImpl">FuzzyLookupFactory</str>
885           <str name="field">_text_</str>
886           <str name="suggestAnalyzerFieldType">string</str>
887       </lst>
888   </searchComponent>
889
```

In the search component 'suggest', the following parameters are set:

- The request incorporates the "suggest" search component defined previously.
- In the request handler component, the field 'suggest.count' is set to 5 to allow 5 suggestions every time the user begins to type a query in the search box.

A screenshot of changes made to the suggest request handler in solrcofig.xml is as follows:

```
889
890   <!-- A request handler for demonstrating the auto suggest component. -->
891   <requestHandler class="solr.SearchHandler" name="/suggest">
892       <lst name="defaults">
893           <str name="suggest">true</str>
894           <str name="suggest.count">5</str>
895           <str name="suggest.dictionary">suggest</str>
896       </lst>
897       <arr name="components">
898           <str>suggest</str>
899       </arr>
900   </requestHandler>
901
```

The changes made to solr.config.xml are saved. Once these changes are saved, the autocomplete functionality will work and was tested using the Solr UI.

III. SNIPPET GENERATION

To generate snippets for search results returned, the following steps were executed:

1. For each search result obtained, the webpage was opened to extract snippets from. So, in order to parse these HTML pages, an external HTML DOM parser library 'Simple_HTML_DOM' was used.
2. For every search result, the document ID is extracted. Also, the Bostn_Globe_Map.csv file is loaded into an associative array in PHP with the index as the ID of the HTML file and the value as the corresponding URL. So, for every search result, the document ID is indexed into this array to get the webpage URL.
3. Using the 'file_get_html' method of 'simple_html_dom.php', the DOM from the URL is extracted and stored.
4. Only the paragraph elements from this DOM are extracted as snippets are only looked for here.
5. The 'script' tags are removed from these paragraph elements if present.
6. The HTML content is split based on separators (".", "," and ":") and stored in an array
7. This array is then scanned to eliminate elements that are empty or which contain the phrase "Share via e-mail".
8. The snippets are then formed from this resulting array using the following 6 strategies. The strategies are listed here in the order they are tested. A subsequent method is used only if all the ones above it fail.
   a) If the entire query (even if it has multiple words) is present in a sentence as phrase, then that sentence is returned as the snippet.
   b) Else, if all the query words are present in a sentence although not continuously, the sentence is chosen as the snippet.
   c) The query is split into bigrams. If a sentence contains any of the bigrams too, it is chosen as a snippet.
   d) Then, a sentence that contains at least one of the query words is selected as the snippet.
   e) If all the above fail, then the meta-description tag of the header is checked if it contains any query words.
   f) As the last resort, the title of the document is returned as a snippet if it contains any query terms.
   g) If all the above fail, an empty string is returned as the snippet.
9. The length of the snippet is kept to 160 characters.
   If the length of the sentence chosen as the snippet is less than 160 characters, then portions of the previous and next sentences are used to make it a length of 160 characters. If the length of the sentence chosen as the snippet is greater than 160 characters, then it is truncated after ensuring that the query words are present in it.

TOOLS USED:

1. Spelling Correction
   Peter Norvig's spell correct code in PHP is used and it is downloaded from https://www.phpclasses.org/package/4859-PHP-Suggest-corrected-spelling-text-in-pure-PHP.html#download.
2. Autocomplete
   Solr's built in autosuggest component was used.
3. Snippet Generation

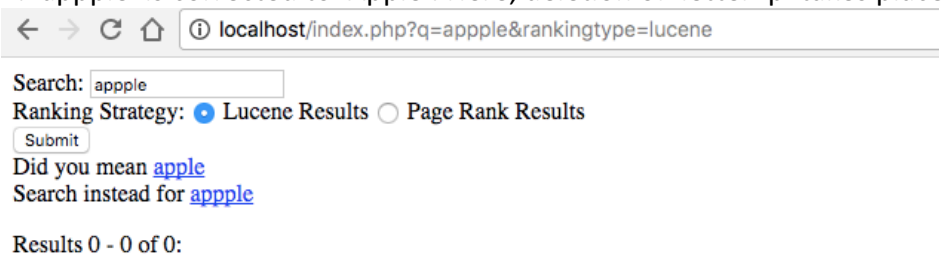
The external library Simple HTML DOM was used for HTML DOM parsing. Simple HTML DOM parser code from http://simplehtmldom.sourceforge.net/.

ANALYSIS OF RESULTS:

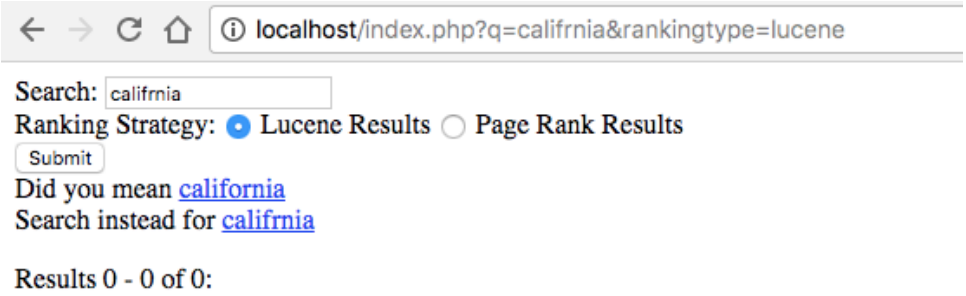
SPELL CORRECT

Five examples of spell correct are shown below:

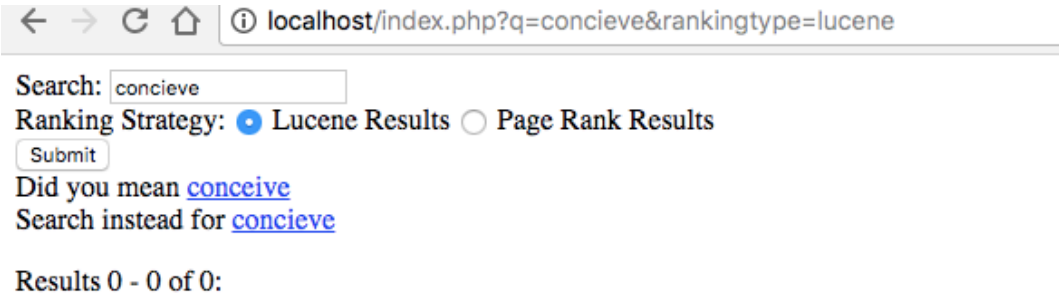
1. 'appple' is corrected to 'Apple'. Here, deletion of letter 'p' takes place to get the right word.



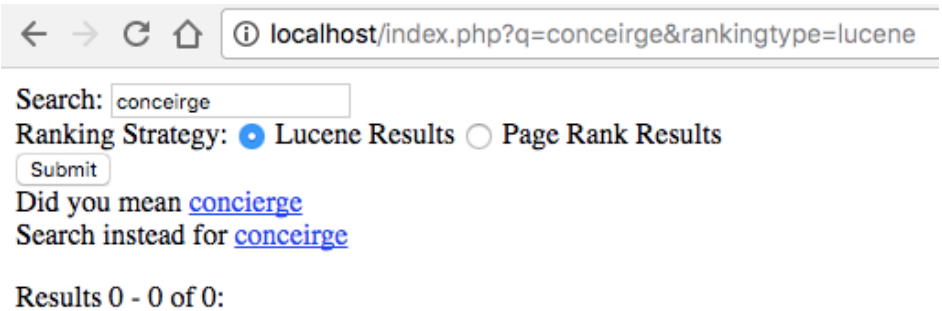
2. 'califrnia' is corrected to 'california'. Here, addition of letter 'p' takes place to get the right word.



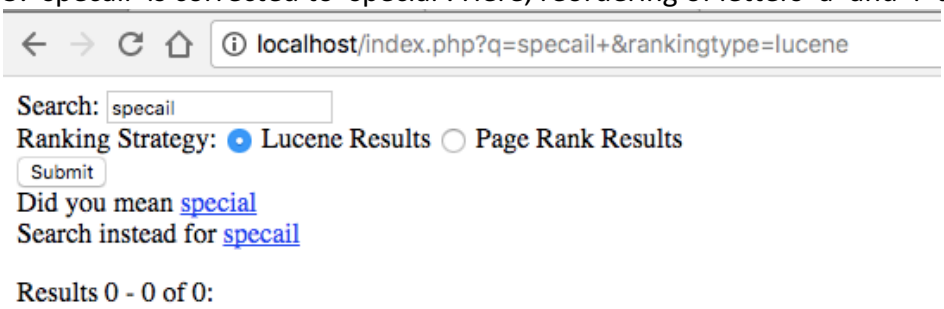
3. 'concieve' is corrected to 'conceive'. Here, reordering of letters 'i' and 'e' takes place to get the right word.



4. 'conceirge' is corrected to 'concierge'. Here, reordering of letters 'e' and 'i' takes place to get the right word.



5. ‘specail’ is corrected to ‘special’. Here, reordering of letters ‘a’ and ‘i’ takes place to get the right word.



AUTOCOMPLETE

Five examples of auto complete are shown below: