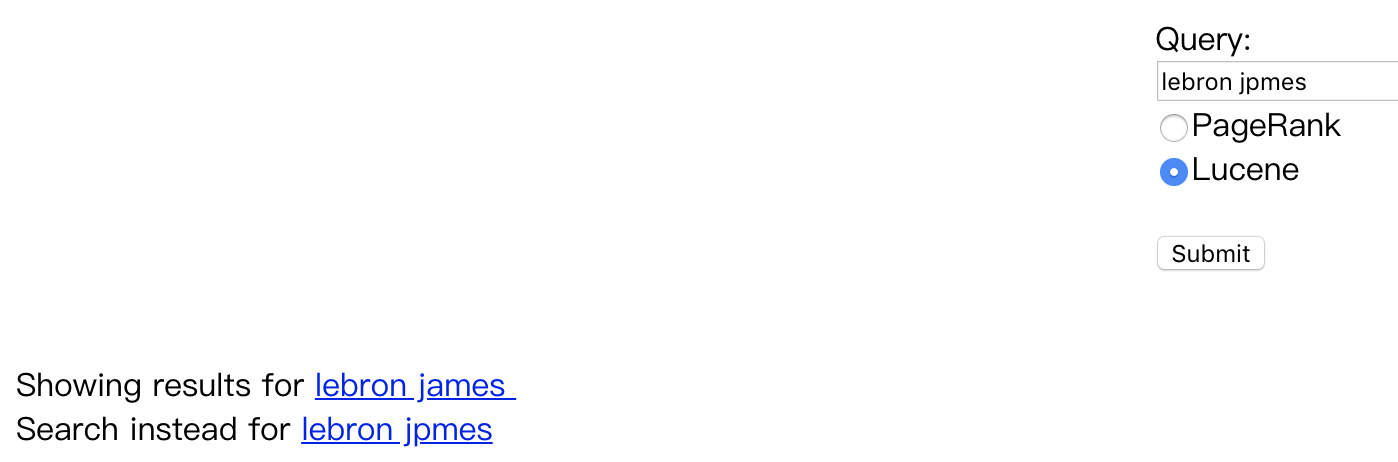
**CSCI572\_Hw5\_Report\_Haoyu Jiang**

In this assignment, I would use an external spelling correction program in conjunction with Solr, enhance the autocomplete functionality of Solr and show snippet that includes one or more of the query keywords along with each link, based on the web interface created in hw4.

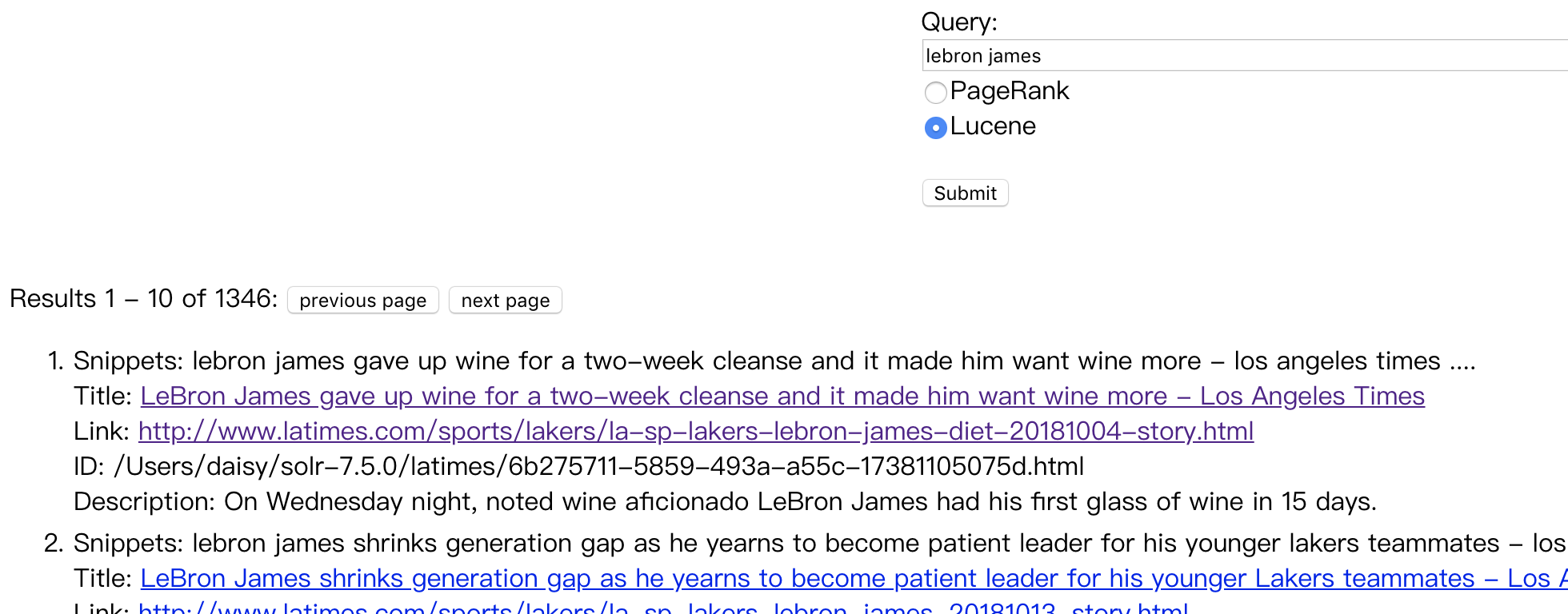
**Part 1: Spelling Correction**

Step 1. I generated big.txt by writing bigtext.java script for Norvig’s spelling program PhP version called SpellCorrector.php. Apache Tika parser was used in bigtext.java to deal with html files and write contents of la times websites to big.txt.

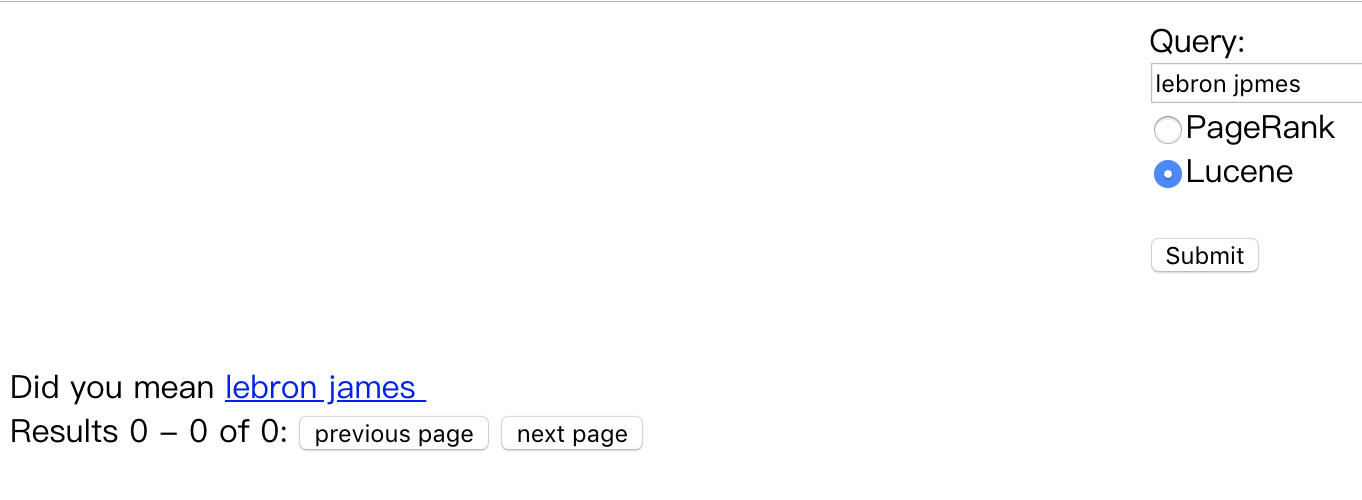
Step 2. I included SpellCorrector.php on my client.php and write code to compare each word in query with SpellCorrector::correct(word). If some words of the query found to be revised, it would show “ Showing results for corrected query” and “Search instead for original query” as followed:



Both corrected query and original query are clickable. The results after clicking “submit” would base on corrected query. The response format is exactly same with Google.

The results after clicking corrected query would base on corrected query as followed:

while results after clicking original query would base on original query even when it contained wrong word. It would also show “Did you mean corrected query” as followed:



**Part 2: Autocomplete**

Step 1. I configured suggester in solrconfig.xml followed the link <https://lucene.apache.org/solr/guide/6_6/suggester.html> and checked the suggestion in Solr UI which would provide matched terms in my index starting with the provided characters. Besides, I revised config for making 'AND' as default instead of 'OR' for multi-word queries in solr and reload the data.

Step 2. Then I decided to use Ajax Autocomplete for jQuery which allowed you to easily create autocomplete/autosuggest boxes for text input fields followed the link <https://www.devbridge.com/sourcery/components/jquery-autocomplete/>. At the same time, I used curl function in php to get the suggestions from solr in json fomat and process these suggestions into suitable format for Ajax Autocomplete for jQuery. During the process, if the query contains multiple words, I split query into words and the suggestions would be generated based on last word of query along with the former words unchanged.

**Part 3: Snippets**

Step 1. For results of each query, I have included simple\_html\_dom.php into client.php. The file\_get\_contents() was firstly used to read file as string and then call the function in simple\_html\_dom.php to create DOM for the string and dump the contents without tag from html.

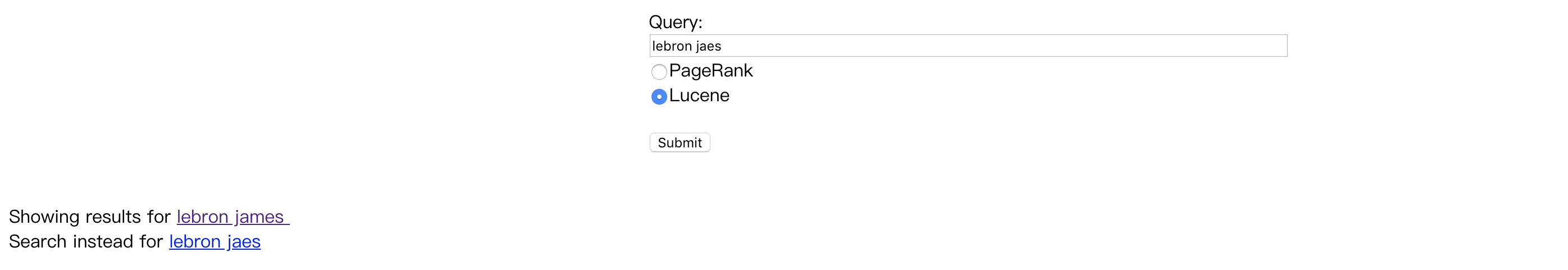
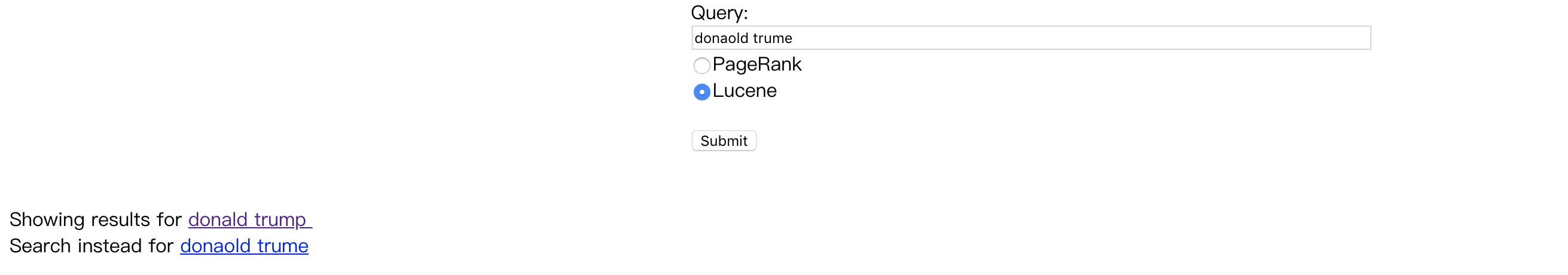
Step 2. For each search result, I looked for a string match of the query terms with the web page and return the first sentence that provides a match. If no match is found, I used description to match query terms as feedback.

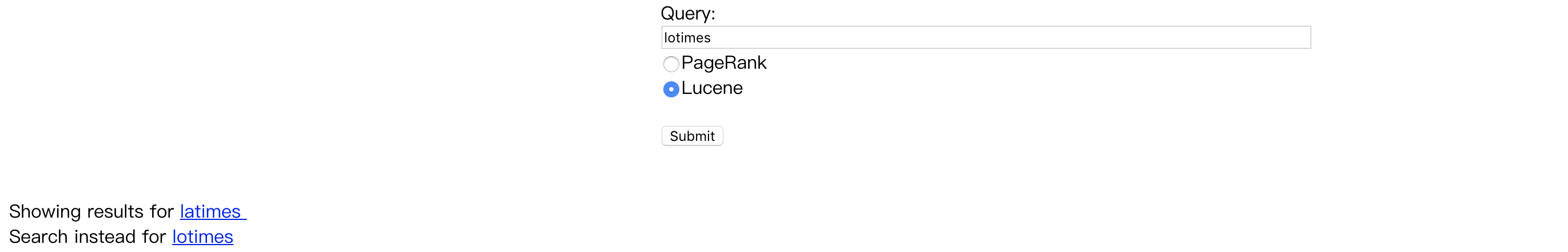
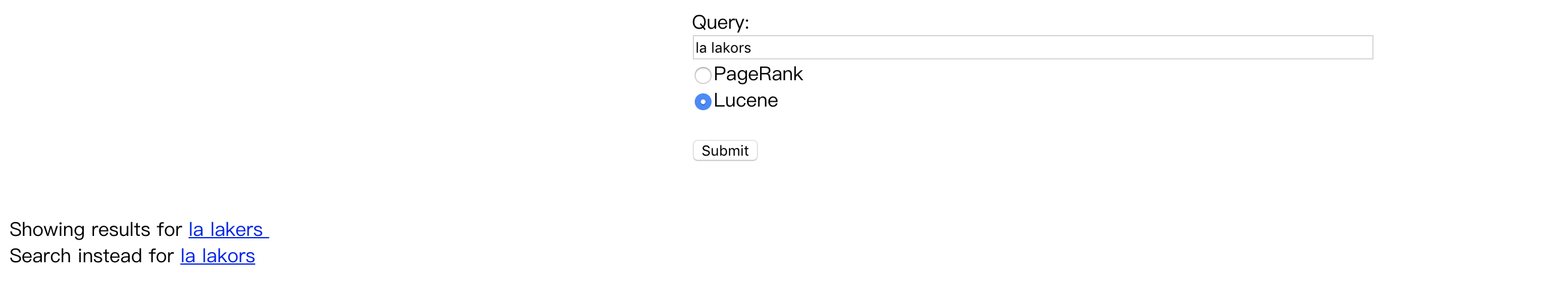
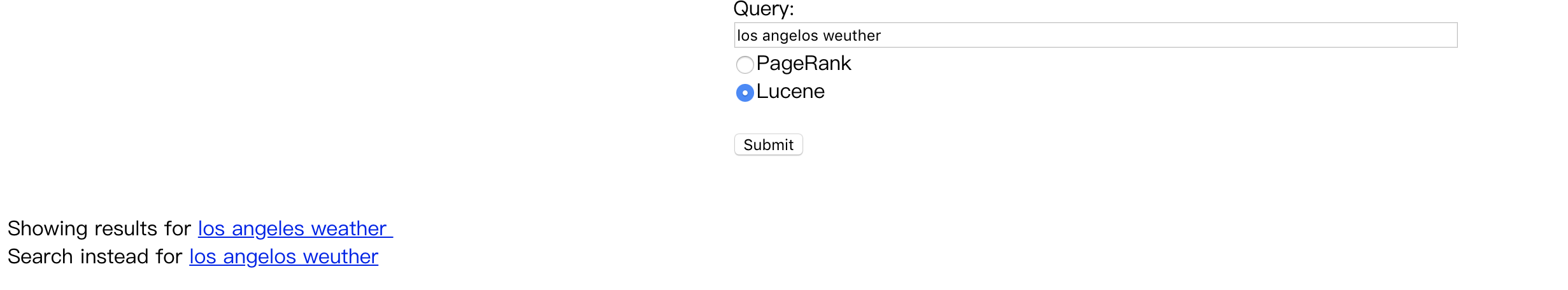
For multiple term queries, I have followed the processing orders as finding a sentence with all the terms together. If not, return the sentence that has all the terms in it, even if they are not together or in same order. If none of the fore-mentioned are found, return the first sentence with at least one query term in it. If no match is found, I used description to match query terms as feedback.

Step 3. For the matched snippets, I processed the snippets further and tried to restrict to around 160 characters.

**Part 4: Analysis of the results:**

FIVE examples of misspelled terms that are correctly handled by spelling correction program.

**** ****

**** **** 

FIVE examples of auto-completion. 