REPORT

The part of project is divided into three parts :

1. Spell Correction

2. Auto-Complete

3. Text Snippet

Premise : The above mentioned functionalities are built on our last part of project where we implemented a search engine on Solr with a functionality to choose between Lucene and Pagerank.

Spell Correction :

1. For Spell Correction, I have used Peter Norvig's spell correction algorithm. Steps taken were as follows :

a. Download SpellCorrector.php from the web. Now this algorithm uses a bag of words' file called big.txt in the algorithm. Based on the occurrence of words' in this file, the rank of corrected words is decided and thus the closest corrected word is suggested.

b. Create a big.txt using Apache Tika Parser program developed in Java. The file created was 182MB. Using isalpha() function of python, remove all the non alphabetical characters. The size reduces by 1/4th.

c. Include Spellcorrector file, big.txt in the working directory. Include SpellCorrector in 'search.php' . Make sure correct() function in SpellCorrector uses big.txt. By function SpellCorrector::correct(query) predict the correct word.

d. Use the predicted word to give a correction of the spelling entered by user.

Auto-Complete :

1. I have used Solr's Suggest Componen to provide autocomplete suggestions to user entered queries. The steps were as follows :

a. Modify the solrconfig.xml file to include a suggest component with some basic properties such as the lookup implementation we are using which was 'FuzzyLookupFactory'. Add the

NonFuzzyPrefix field and I have set it's value to 5 to make sure the suggested words have the same prefix upto atleast 5 characters of the word if possible.

b. After adding the suggest component we will add a request handler component for the same. This will have basic properties of suggestion such as suggest.count which refers to the number of suggestions made by Suggest component.

c. To use this suggestions on our search page, we will use ajax and jquery in our 'search.php'. Now, I have added an autocomplete function on my input field which is processed by 'Service.php' and returns a drop-down suggestion of auto-suggestions.

d. Service.php using the 'query' term from the request out of search.php's autocomplete function calls the suggest component of the Solr and returns the array of suggested values.

e. Autocomplete function handles this and one can see the suggestions on search page.

Text Snippet :

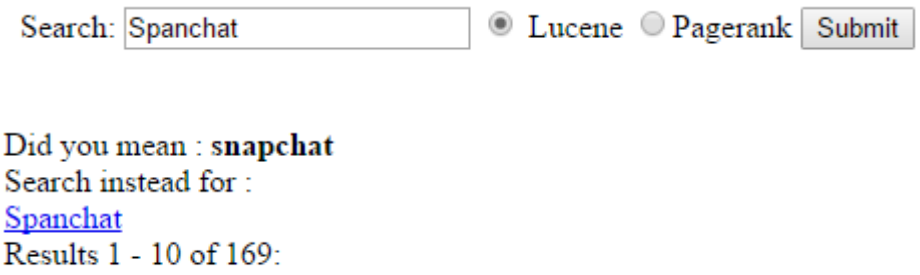
For the text snippet, I have used 'Simple_html_dom' parser. The parser fetches a dom object of the html file using file_get_contents() function of the parser. Through this dom object, we can fetch the plain text of our html file using 'plaintext' function of dom object.

Iterating through each sentence, we get those sentences which contain our query term and we include it into our snippet text and output the snippet.

Analysis of the Result :

SpellCorrection :

1. 'Spanchat' : Corrected word : Snapchat



2. 'braine' : Corrected word : brain

Did you mean : **brain**
Search instead for :
braine
Results 1 - 10 of 1418:

3. 'batmna' : Corrected word : batman

Search: batmna ◉ Lucene ○ Pagerank Submit

Did you mean : **batman**
Search instead for :
batmna
Results 1 - 10 of 238:

4. 'simbe' : Corrected word : some

Search: simbe ◉ Lucene ○ Pagerank Submit

Did you mean : **some**
Search instead for :
simbe
Results 1 - 10 of 14106:

5. 'illgle' : Corrected word : illegal

Search: Illegle ◉ Lucene ○ Pagerank Submit

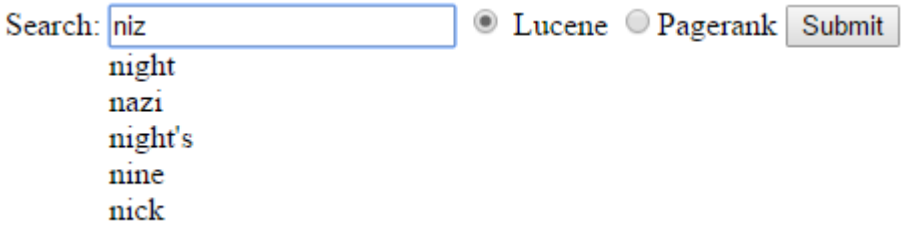
Did you mean : **illegal**
Search instead for :
Illegle
Results 1 - 10 of 1473:

AutoComplete :

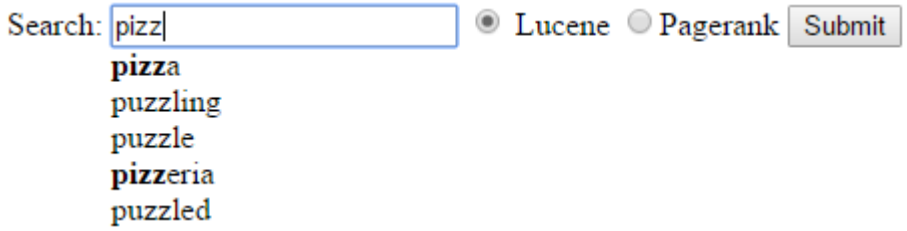
1. 'trizl'

Search: trizl ◉ Lucene ○ Pagerank Submit
trial
trials
triple
trilogy
trillion

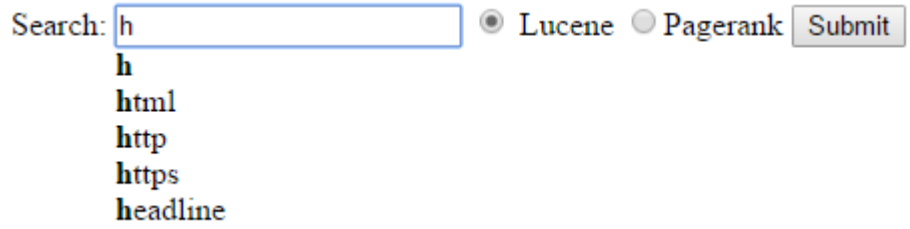
2. 'niz'


Search: niz
Lucene
Pagerank
Submit
night
nazi
night's
nine
nick

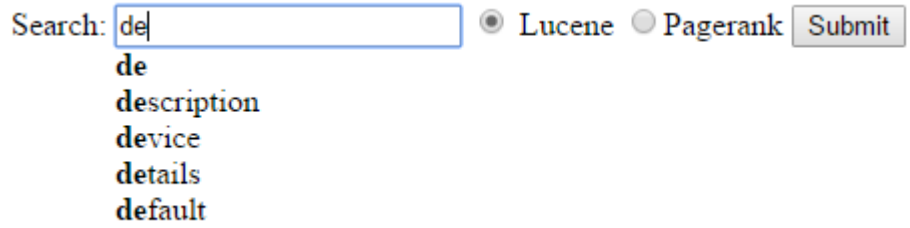
3. 'pizz'


Search: pizz
Lucene
Pagerank
Submit
pizza
puzzling
puzzle
pizzeria
puzzled

4. 'h'


Search: h
Lucene
Pagerank
Submit
h
html
http
https
headline

5. 'de'


Search: de
Lucene
Pagerank
Submit
de
description
device
details
default