

Report : Adding autosuggest, Spell Correction and Snippets to Search Engine.

1) Spell Correction

1.1) Trying out the solr spellcheck component

- Enabled the Spellcheck component and specified the source of the terms in the solrconfig.xml file, which is present in the conf folder of the core.

```
<!-- added index based spell checker -->
<searchComponent name="spellcheck" class="solr.SpellCheckComponent">
  <lst name="spellchecker">

    <str name="classname">solr.IndexBasedSpellChecker</str>
    <str name="spellcheckIndexDir">./spellchecker</str>
    <str name="field">_text_</str>
    <str name="buildOnCommit">true</str>

  </lst>
</searchComponent>
```

- The second step is to add this spellcheck component to the requestHandler “select”

```
<requestHandler name="/select" class="solr.SearchHandler">
  <lst name="defaults">
    <str name="echoParams">explicit</str>
    <int name="rows">10</int>
    <str name="df">_text_</str>
    <str name="q.op">AND</str>
    <str name="spellcheck">true</str>
    <str name="spellcheck.collate">true</str>
  </lst>

  <arr name="last-components">
    <str>spellcheck</str>
  </arr>
</requestHandler>
```

- Once this addition is made, we need to reload the core from the solr UI and we can check by entering a misspelled term for the query.

1.2) Using external spell check program

- Generated the big.txt through a java program which will be used by Norvig's spell correction program to get set of words to calculate edit distance.
- Downloaded Norvig's spelling program written in PHP
<https://www.phpclasses.org/package/4859-PHP-Suggest-corrected-spelling-text-in-pure-PHP.html#download>
- Changed the php code to check for spelling mistakes in the query terms.
 - \$correctValue = SpellCorrector::correct(\$value);

2) Auto Suggest functionality :

- The first step is to add a search component to solrconfig.xml and tell it to use the SuggestComponent.

```
<!-- added suggest component -->

<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>
```

- I added the request handler to solrconfig.xml.. The request handler definition must incorporate the "suggest" search component defined previously.

```
<!-- adding search handler -->

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

  <lst name="defaults">
    <str name="suggest">true</str>
    <str name="suggest.count">10</str>
    <str name="suggest.dictionary">suggest</str>
    <str name="nonFuzzyPrefix">4</str>
  </lst>

  <arr name="components">
    <str>suggest</str>
  </arr>

</requestHandler>
```

- Once this addition is made, we need to reload the core from the solr UI and we can check we are getting suggestions for query terms.
- Added the autosuggest feature on search page -
 - Created a php file 'suggest.php' which will relay the request and response for autosuggest from solr server.
 - Used AJAX to make a request to 'suggest.php' everytime the user is types a term in the query bar.

3) Snippets functionality:

- Extracted text out of html files (text in the <p> tags) using Jsoup Library and created corresponding text files which can be used by the search engine to generate snippet
- Changed the php file to generate snippet . - for every document being returned as result, get the first line in the corresponding text file that contains all of the query terms. If none exists get the first line which contains at least one of the query terms.

Some Example Queries and results

Autosuggest :

califo : california , calif , californians, california's californias
 dona : dona donald don't don done
 ille : illegal ill illegally illinois illness
 immi : immi immigration immediately immigrants immigrant
 break : break breadcrumb breakpointsizes breakingnews breaking

Spell correct :

califona : california
 illegl : illegal
 immigrnat : immigrant
 informatin : information
 mystireous : mysterious

