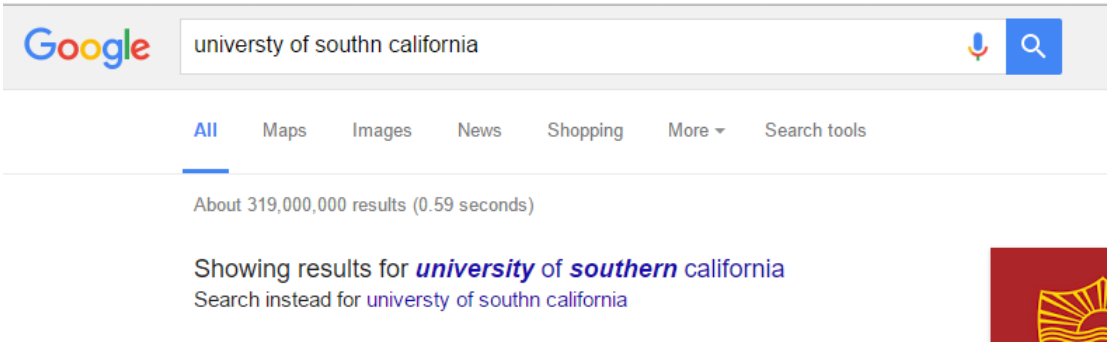
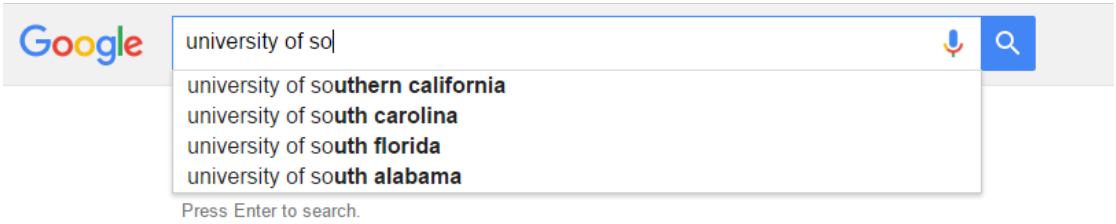
Enhancing Your Search Engine
With
Suggest in Solr

This tutorial explains how to enable autosuggest in Solr. You are required to use a separate spell-checking program to build your own spell correct feature. ***The actual exercise is in a separate document***.

Current search engines automatically identify spelling mistakes in user's queries and return relevant results for the best matching query. The screenshot below shows how "Google" displays results for "university of southern california", even when the query had spelling mistakes.



Another powerful feature that is present in current search engines is autocomplete or suggest. You might have noticed, while you are typing queries into Google, there are several suggestions that come up in the dropdown below the input box.



Suggest Component

The SuggestComponent in Solr provides users with automatic suggestions for query terms. You can use this to implement a powerful auto-suggest feature in your search application.

Although it is possible to use the Spell Checking functionality to power autosuggest behavior, Solr has a dedicated SuggestComponent designed for this functionality.

The first step is to add a search component to solrconfig.xml and tell it to use the SuggestComponent. Here is some sample code that could be used.

```
<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>
```

Here, the name of the "SuggestComponent" is "suggest", the field used to obtain the terms for suggestion is the previously defined "_text_" field. The lookupImpl parameter defines the algorithms used to look up terms in the suggest index. There are several possible implementations to choose from, and some require additional parameters to be configured. This is a suggester in which similarity between the query and suggest terms is measured using Levenshtein algorithm. The field type to use for the query-time and build-time term suggestion analysis is defined by the "suggestAnalyzerFieldType", here it is a string fieldtype.

After adding the search component, a request handler must be added to solrconfig.xml. This request handler works the same as any other request handler, and allows you to configure default parameters for serving suggestion requests. The request handler definition must incorporate the "suggest" search component defined previously.

```
<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>
</arr>
</requestHandler>
```

In the screenshot above, we have defined a request handler for "suggest". The default values for the number of suggestions is set to 5, by defining the value for element "suggest.count", the default dictionary to be used is defined by "suggest.dictionary".

Once the request handler is defined in solrconfig.xml, we need to reload the core as explained in the first section. Now let us try to see the suggest functionality in the Solr UI. Go to the Query view, and change the requesthandler from "/select" to "/suggest". In this example, I have typed just "califo" in the query, and executed the query.

The following is the response of the Solr query in json format. Here we can see that, for the term “califo”, 5 suggestions have been returned each in the order of the weight. The weight specifies its match with the original term.


```
{
  "responseHeader":{
    "status":0,
    "QTime":2},
  "suggest":{"suggest":{
      "califo":{
        "numFound":5,
        "suggestions":[{
            "term":"califo",
            "weight":1,
            "payload":""},
          {
            "term":"california",
            "weight":1304,
            "payload":""},
          {
            "term":"californiamarketcenter.com",
            "weight":15,
            "payload":""},
          {
            "term":"californias",
            "weight":11,
            "payload":""},
          {
            "term":"california's",
            "weight":6,
            "payload":""}]}}}}
```


You can leverage this suggest feature, by making the “suggest” request before actually making the “select” request to Solr, when the user is typing in the query. “Select” request can be sent only when the user presses “Enter”.

More details on this feature can be found in Solr Wiki.