CSCI 572 HW4

SPELL CHECKING AND AUTOCOMPLETE

PRADEEP KUMAR NANDA KUMAR

USC ID: 9834726643

DATED: 26-Apr-2016

TABLE OF CONTENTS

FILES ATTACHED

1. QueryProcessor.php and QueryRequest.htm
2. Manage-schema and solrconfig.xml
3. SpellCorrector.php with my amendments to make dictionary from a directory of pages
4. Apache-Solr-Php-Clinet: service.php with my amendments to enable "/suggest" query in Solr.
5. Serialized_dictionary.txt and Readable_Dictionary.txt
6. Screen Shots of analysis of results
7. Stopword.txt

REFERENCES:

1. http://www-scf.usc.edu/~csci572/Slides/* - and Lectures from csci 572 Spring 2016
2. Wikipedia – General Overview of Auto Suggestion

INTRODUCTION AND OBJECTIVE:

Web Search engines like Google, Bing, Yahoo, etc give suggestions on querying terms on the fly and also provide spell correction to enhance searching. Some engines like Google intertwines spell correction and suggestion. Red Underlining of incorrect terms along with suggestions, the phrase "Search Result for xxx instead of yyy" are special features of Google. Many no-words errors, typographical errors and cognitive errors can lead to unexpected search behavior. Tackling by spell checking and suggesting is one of the best suited remedies.

OBJECTIVE:

To enhance my search engine (that I created using GUI Based Apache Solr integration w/o external page ranking for HW3) with "auto-complete" using Apache Solr Auto-Suggest Features and "spell-check" using third party tools

Other Goals

- To learn and use Peter Norwig Spell Corrector
- To learn configure Solr for auto-suggest (i.e., auto-complete)
- To incorporate auto-complete and spell check in my UI

TOOLS ALREADY USED (for HW3)

- Ubuntu 14.04, XAMPP – PHP and Apache TomCat
- Apache Solr, Apache-Solr-Php-Client – Services.php as prescribed in the description
- Networkx Library for Python
- HTML; CSS+Bootstrap; JavaScript; Jquery; PHP, Ajax Call

TOOLS USED (for HW4):

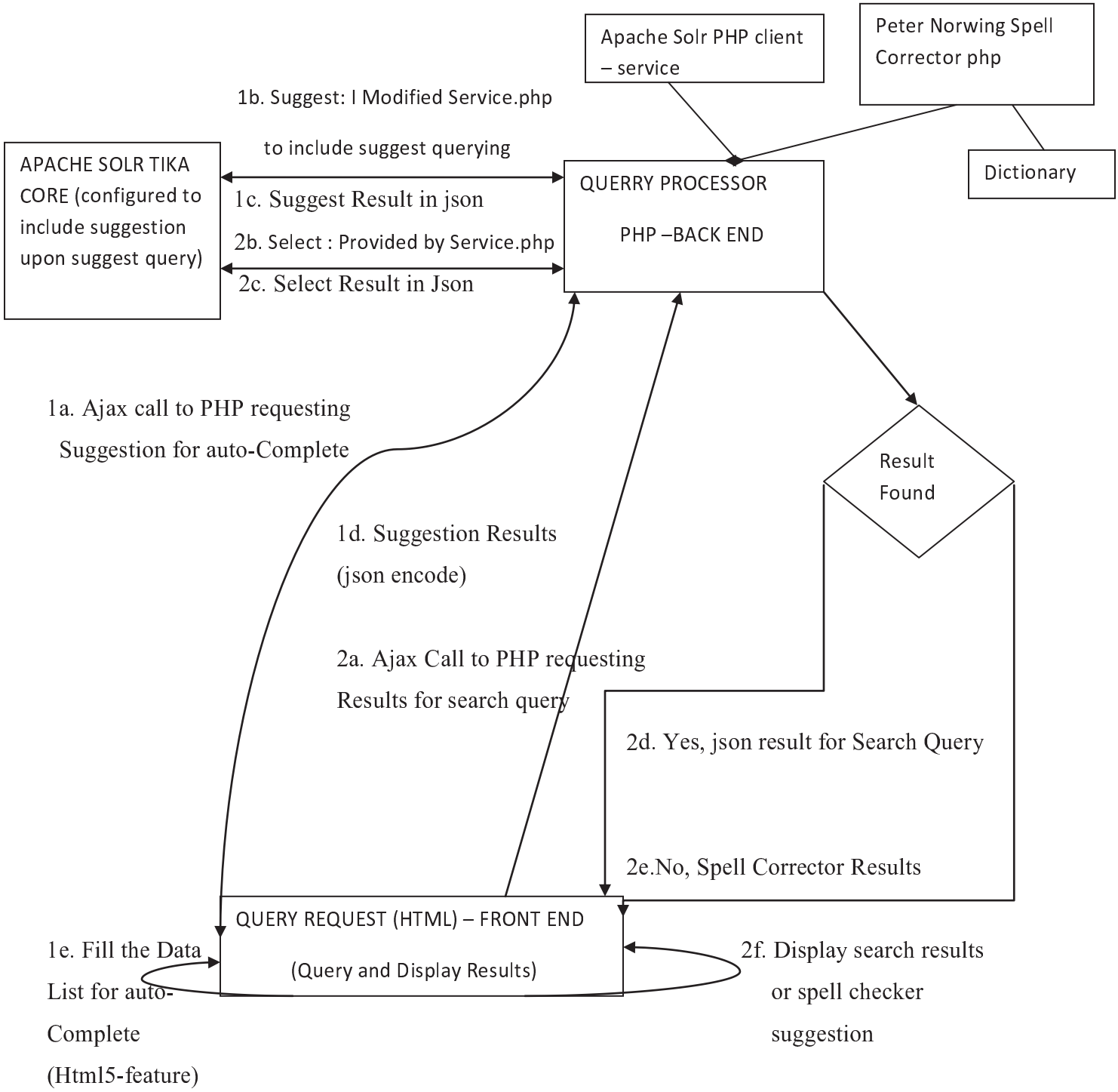
Spell Checker

- Peter Norwig's Spell Checker – PHP version as prescribed in the HW4 Description

Apache Solr Auto-Suggest

- Configured Solr to suggest words for auto-complete

WORKING PRINCIPLE:



I. STEPS FOLLOWED TO COMPLETE THE PROJECT

1. **Amendments Peter Norwig Spell Corrector**

   Firstly, I downloaded the PHP version of Peter Norwig Spell Corrector. It had couple of errors which I fixed - "isset(array[index])" and " memory limit" issues. Once this is done, I amended the "correct($word)" method in the spell corrector to read all the pages downloaded and stored in the "/PagesDownloaded" Directory and creates a Serialized Dictionary as well as Human Readable Dictionary out of it. This helped in fixing the memory limit to "1GB". Further this dictionary is used by the Spell Corrector to train the model. This one is included in the Query Processor.php for spell checking.

2. **Configuration of Apache Solr for Auto Suggest as described in the HW4 Description**

   "/suggest", is the query used for retrieving suggestion for a single word. Also, it is NOT the last array component of "/select" query, thereby de-coupling both of them.

   - **Stemming**
     - Added the "SnowballPorterFilterFactory" for stemming. The keyword attributes boosts the contribution of original words along with the stemmed words, as duplicate is created. Now, in order to remove extra duplicates, the "RemoveDuplicatesToken" is set as filter.
       Calucation, Calculating, Calculate, Calculator, Calculated are all stemmed to Calcul – enhances better searching (by matching more documents)
   - **Stop Words**
     - A lot of stop words such as "of","the","In","was", etc were fed to the stopwords.txt in the configuration folder. This prevents the stop words from being
   - **Html Tag Filters**
     - In order to exclude the html tags, i.e, text within "< >" , a "charFilter" is configured to inherit from the lucene analysis class "HtmlStipFilterFactory". For example <title style="border:non;"> TITLE </title>, will be stripped to TITLE.

3. **Amendments in Service.php – Apache-Solr-PHP-client**

   Added methods and attributes in Service.php to support querying using "/suggest" query.

   public function suggest($query, $params = array(), $method = self::METHOD_GET);

   _suggestURL and self::SUGGEST_SERVLET; this helps the PHP to query Solr using "suggest" as query field. The result of "/suggest" query is used for auto-complete.

   If the number of query words is greater than 4, then suggestion is requested only for the last word, otherwise suggestion is requested for each word split by space. For each word received from solr suggest, combination of them is created (in JavaScript) and the history of suggestion is maintained for that session, thereby enhancing auto-complete features. Also, any special character returned is stripped from the suggestions. Inclusion of stopwords prevents Apache Solr from suggestion "an, and, at, as, are, …" for the word "a", thereby forcing the Solr to suggest useful or related words such as "America", "abroad".

II Analysis of Results:

The default browser-query auto-complete and spell check is turned off. Thereby the results are from solr querying. Screen shots of auto-complete and spell check is attached with the report. Clicking upon the suggested spell check fetches the required results as seen from screen shots. **Screen shots** contains auto complete for " news publication", "gould", and "gould school", further spell check was done one "mstr program rquiremnts goudl school" and the return result was – Did you mean: "master program requirements gould school" as clearly seen from the screen shots. The results were closely resembling the better results as it was able to correct and suggest relevant details with better accuracy.

Autocomplete Example: "go" => go, gould, gould's, government, gouldslaw

Spell scheck Example: "mstr program rquiremnts goudl school" => "master program requirements gould school"

(Refer the screen shots for more examples)

NOTE: THE DEFAULT SEARCH WITH PAGERANK PRODUCES BETTER RESUTLS. THE PROJECT DEMO WAS TESTED WITH SOLR, WHICH MAY NOT PRODUCE THE DESIRED RESULTS IN TOP 10 FETCHES.