Homework 5:

Adding Spell Checking, AutoComplete and Snippets to Your Search Engine

Objectives

- Experience using a third-party spell program
- Developing efficient methods for accomplishing autocomplete
- Implementing a simple snippet feature for search results

In the previous document (AutocompleteInSolr.pdf) you saw how to enhance the Solr program with spelling correction and an autocomplete (suggest) function. **In this exercise you are asked to use an external spelling correction program in conjunction with Solr and to enhance the autocomplete functionality of Solr**. In addition, the search results you return should include, in addition to a link to the resulting web page, a snippet that includes one or more of the query keywords. In the case of spelling correction, you may use an existing third-party program adapted to your downloaded files. In the case of autocomplete you will need to enhance your client program that communicates with Solr to deliver autocomplete suggestions to the web interface you created in an earlier homework. In the case of snippets, you will need to locate a string in the web page that includes the query terms and output that sentence along with the link results.

Description of the Exercise

Spelling Correction: in the class lecture you saw a complete spelling correction program developed by Peter Norvig. The program was written in Python. For this exercise you are welcome to use whatever third-party spelling program you wish, or you may even write your own. Since most of you wrote your previous homework client using PhP, you may want to adopt a version of Norvig's spelling program written in PhP and run it on your server. You can download the PhP version of Norvig's spelling corrector from here:

http://www.phpclasses.org/package/4859-PHP-Suggest-corrected-spelling-text-in-pure-PHP.html#download

(you will have to register at the site before being able to download the software, registration is free)

If you prefer to use Norvig's program in a different language, a wide variety of implementations can be found at the bottom of this page, http://norvig.com/spell-correct.html

You should make sure to enhance your spelling correction program with a set of terms that are specific to the news website that you are responsible for. You should make sure that common terms such as *climate, election, etc.*, and the terms used in the queries of homework #4 are handled. Norvig's spell correction program uses a text file(''big.txt") to get set of words to calculate edit distance. For this you should create your own "big.txt" for your specified news

website. You can use any parser (our suggestion - Apache Tika) and Instructions on using apache Tika for this purpose can be found here.

Autocomplete: for the autocomplete portion of the exercise, you will have to modify your client program, so it accepts single character insertions to the text box, and returns a list of completions/suggestions.

There are several ways to implement the autocomplete functionality while using Solr. One possible way is to use the FuzzyLookupFactory (https://lucene.apache.org/solr/guide/6_6/suggester.html) feature of Solr/Lucene. The FuzzyLookupFactory creates suggestions for misspelled words in fields. It assumes that what you're sending as the suggest.query parameter is the beginning of the suggestion. It will match terms in your index starting with the provided characters. So, if the query is "ca" it will return all the words starting with "ca", e.g. "california" and "carolina" etc. **For the first character and second character that is entered, some autocomplete suggestions should appear.**

For this to work you need to enable the suggest component as described in the tutorial but add some options.

Note: with respect to specific issues about how spelling corrections are displayed or how autocomplete corrections are displayed **you should imitate the way Google handles both**. For example, while typing in the search box, the top suggestions should automatically appear and be updated as the user keeps typing. The spellcheck suggestion should appear at the top of the retrieved results. If the word typed is correct no suggestion should appear at the top.

Snippets: To produce a snippet for each of the returned search results you can do the following: for each search result you should open the web page that is referred to. You should then look for a string match of the query terms with the web page. Return the first sentence that provides a match. If no match is found, then no snippet is returned. You may use any external library or use meta-data with query terms.

Note: For multiple term queries, please try 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, then no snippet is returned. Try to restrict to around 160 characters, trim if needed.

Demonstrating Your Solution

There will be an in-class demonstration where all students will demonstration their implementation. The demonstration will be on the last day of class. Graders will run a script designed to test your implementation. It will include a set of queries, so we can test autocomplete functionality. It will contain a set of misspellings so we can examine how well your spell correction program performs. And we will be looking at the snippets your program produces for the results returned from the queries.

Submission Instructions

There should be a report describing what you have done. This report should include:

1. Steps you followed to complete this assignment. Include the details of what tools and techniques you used to implement spelling correction and autocomplete.

2. Analysis of the results: In this you should provide FIVE examples of misspelled terms that are correctly handled by your spelling correction program. You should also provide FIVE examples of auto-completion.

3. Using the submit command you should provide a single .zip (CSCI572_HW5.zip) file which contain the following files

- your report (no more than 5 pages)
- the external spelling correction program that was used
- all source code that you wrote, most especially the code implementing the autocomplete functionality. (only your own code, please do not submit meta data files)

You are required to submit your results electronically to the csci572 account on SCF. Though there will be an actual in-class demonstration, you must also submit the above files electronically, by entering the following command from your Unix prompt:

submit -user csci572 -tag hw5 CSCI572_HW5.zip

Appendix

Note 1:

Suggested config change for making 'AND' as default instead of 'OR' for multi-word queries in solr:

Solr default boolean model uses OR instead of AND. So, if your query is "Elon Musk", then the result will match all pages which either have " Elon " OR " Musk " present and not the entire query " Elon Musk".To solve this problem, please do as following to set up the standard Query Parser Parameters:

In **solrconfig.xml** add this line:

<str name="q.op">AND</str>

within this tag: <requestHandler name="/select" class="solr.SearchHandler">

and Inside <lst> default tag within requestHandler tag: <lst name="defaults">

Remember to reload after editing. You won't face queries such as which word to choose in a multi-word query to search for snippet etc. with this.

FAQs

Q1. What should we display if an article already has a description meta information? Do we ignore it and use our own generated snippet or display both?
A. Use the generated one. It is acceptable to use the meta description also if it contains the query terms.

Q2. Can we use default spell checker for HW5?
A. You are not supposed to use default spell-checker for Hw5.

Q3. How to handle multi word queries?
A. To handle queries with two words, please handle each word separately.

Q4. Can we use solr's inbuilt auto-complete features?
A. Yes.

Q5. How should the spell correction and auto complete working look like.
A. Imitate googles auto complete and spell correction, your result should look like that

Q6. when using the php corrector and when loading big.txt, error log says allowed memory size exhausted.
A6. Add the following code, <? php ini_set ('memory_limit', '1024M')?>. This will solve it.

Q7. What are the grading guidelines for HW5?
A7. It's an in-class demo. So no grading guidelines will be published for Hw5.

Q8. Snippets for each of the result should display first sentence that contains the query. Does this mean we have to read all the result web pages and parse each of them?
A8. You need to check in all 10 documents and get all 10 snippets to print for 10 results. Also, you can do it in many ways. May not be the first sentence.

Q9. "If no match is found, then no snippet is returned." What does it mean?
A9. No need to print any snippet if no match is found.

Q10. Do we need to store user's history for suggestions?
A10. No need to store any user history to get this functionality.

Q11. Do we need to extract the data from all the 19000+ files into big.txt and do we have to avoid duplicates?
A11. Yes, you need to extract the content from all your files. Please Don't avoid duplicates, please read how Norvig's spell correction works, you will find why you need to have duplicates.

Q12. In hw4 it tells us to set the 'text' to have a type of "text general". However, in hw5 it says "text_en_splitting".
A12. You can leave as it is. It works just fine.

Q13. Solr exception**:** Java.lang.String cannot be cast to Java.lang.String
A13. please check whether you added suggest component as stated in document(at right place and with right tags).

Q14. How does big.txt look like?
A. You need to parse content of html files into big.txt. Please refer to http://norvig.com/big.txt.

Q15. Do we need to extract the data from all the 19000+ files into big.txt and do we have to avoid duplicates?
A. Yes, you need to extract the content from all your files. Please Don't avoid duplicates, please read how Norvig's spell correction works, If you de-duplicate, then you will defeat the purpose of using word frequency to estimate P(c), where c is the corrected spelling.

Q16. The document "SpellcheckandAutocompletioninSolr.pdf" is only for reference? For both spellcheck and autocompletion we don't use the solr internal functions? We both use external ones?
A. #1 for spell check: You use external program
#2 for Auto completion: You use the one inbuilt in solr.

Q17. If we search "Donad Trup", when we put "Donad", the spell check should show "Donald", what if we put "Donad Trup", what spell check should show? Should we combine each spell check result, like "Donald Trump", or just show "Donad Trump", only correct the correct word?

A. Please check for each word separately when you have multiple words in query. You need not get right spell check for every query you enter. Please have right big.txt will all text as per your news site. We will take care while grading.

Q18. Do we need to extract the data from all the 19000+ files into big.txt and do we have to avoid duplicates?
A. Yes, you need to extract the content from all your files. Please Don't avoid duplicates, please read how Norvig's spell correction works, you will find why you need to have duplicates.

Q19. Can we use the big.txt provided on the Norvig's website and add query terms from hw4 into it, or do we have to generate ourselves with Tika?
A. Please generate it. It doesn't make sense to load all the words from " The adventures of sherlock Holmes" in to memory. You might not get correct results too.

Q20. Can we remove the radio button and functionality of page rank here for hw5?

A. No. You can leave it as it was for Hw4.

Q21. I did not do the pagination function on my page. Can we just do top ten results?
A. Pagination is NOT a requirement for this exercise. Top 10 should suffice.