

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

ANS:

1. I used Solr Server, HTTP&PHP Server that I have already configured, and all files on the server that I have used in the exercise 4. I added the suggestion component to myexample core's solrconfig.xml file according to the instruction and then reloaded the core.
2. I downloaded tika-app.jar from Apache-Tika's website. I created a script parsed_script.sh to parse all .html document that I am assigned to(New York Times website). I collected all of the parsed context into a file parsed.txt.
3. I downloaded PHP's spellcorrector component from <http://www.phpclasses.org/package/4859-PHP-Suggest-corrected-spelling-text-in-pure-PHP.html#download> and placed it into my server's directory. I change the SpellCorrector.php to read the training file from parsed.txt. I modified query.php to call out this component. The component compiled as list vocabulary necessary for performing Spell Checking as its first run and stored the relevant information in serialized_dictionary.txt. I placed this file in the server's directory.
4. I modified PHP's Solr client library, Service.php, enabling it to make a HTTP request to the /suggest service on Solr's server.
5. I created a file suggest_service.php that call to Solr server to request a list of suggested words. This file will be called by query.php via AJAX communication to retrieve a list of suggestion without reloading the search page.
6. I created a file funtions.php that contains helper functions that retrieve a snippet from a specified file and add text decoration.
7. I moved all of the news files into the server's directory.
8. I modified query.php to link all of the functionality together; Spell correction, Suggestion, Snippet. I modified the page layout. I used SublimeText as my IDE. I tested, debugged, and corrected the code to ensure that the system works according to what the instruction asked.
9. For detailed implementation of Spelling Correction, I rely on the suggested file, SpellCorrector.php. Basically, when user submits a search term, the system will tokenize the search term into multiple terms(if any), and call SpellCorrector::correct(...) to check whether each of the terms is valid or not. If at least one of them is not recognized by the SpellCorrector, the system will generate a new suggested search query by combining all of the corrected terms.
10. For detailed implementation of Autocomplete/ Suggestion, as user enters the search query, the system will tokenize the search term into multiple terms(if any) and make a request for the suggestion of the most recent term that user is currently typing. Basically, query.php will send AJAX message to suggest_service.php. Suggest_service.php will make a HTTP request to /suggest service on Solr server via PHP Solr's client library. Suggest_service.php obtains the result message and parse it back to query.php. Query.php use javascript to deserialize this message and parse them to HTML content. The dropdown suggestion list is implemented by incorporating jQueryUI library.
11. For detailed implementation of Snippet, as a link is generated, query.php will call a function getSnippet(...) in functions.php, sending in the search_term and doc_id. The function uses doc_id to obtain the relevant news file stored in the server and searches for the matching content using search_term and regular expression. The matching content is expanded to be centered around the keyword. The keyword is then decorated with a red color and put into the Snippet section.

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.

ANS:

Misspelled Terms:

1. Searched for 'hellx', the page suggested 'hell'. This completion is within 1 edit distance.

Search:

Did you mean: [hell](#)

2. Searched for 'hellx hello'. The completion is fixed for just the term 'hellx', which is not appeared in the stored dictionary.

Search:

Did you mean: [hell hello](#)

3. Similar to 1., but the fix happened to the character inside(not at the border) of the term.

Search:

Did you mean: [donald](#)

4. Similar to 1., but the completion is within 2 edit distance.

Search:

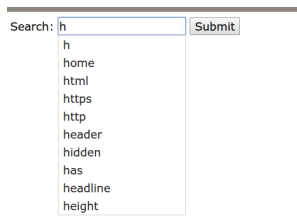
Did you mean: [trump](#)

5. Entered a completely nonsense query does not return any correction. This is as expected and there is no similar word within 2 edit distance.

Search:

Auto-Completion:

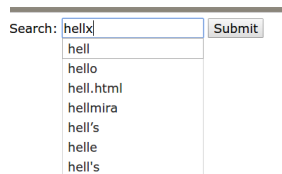
1. When user enters only one character for the term, the system displays a list of suggestion words comprised of (at most) 10 elements.



Search:

- h
- home
- html
- https
- http
- header
- hidden
- has
- headline
- height

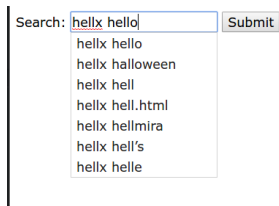
2. Similar to 1., but when the search term is larger than 1 character, the displayed elements are capped to at most 7.



Search:

- hell
- hello
- hell.html
- hellmira
- hell's
- helle
- hell's

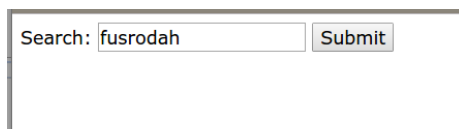
3. When user types in multiple words, the system only provides a suggestion of the latest typing term. As you can see, the system does not provide a suggestion for 'hellx'.



Search:

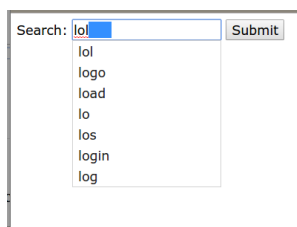
- hello
- halloween
- hell
- hell.html
- hellmira
- hell's
- helle

4. Entered a completely nonsense term, no suggestion provided.



Search:

5. The system does not send a request for a suggestion of a whitespace. The suggestion request will be sent when user enters a non-whitespace character.



Search:

- lol
- logo
- load
- lo
- los
- login
- log