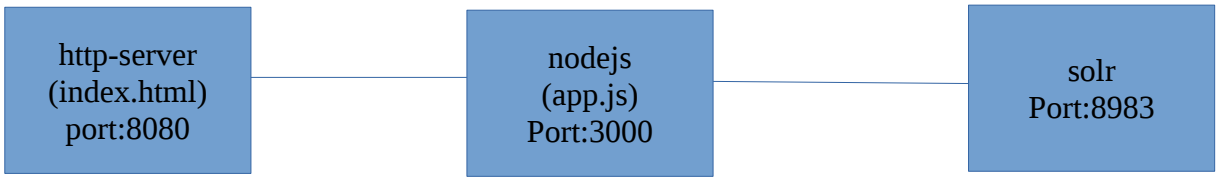
CSCI-572 Information Retrieval : Assignment-4-Report:
Name : Ravi Kiran Chadalawada
USC id : 9811634305
Websites : LAtimes and Huffingtonpost

URL of Youtube video : https://youtu.be/ZoaNXbKWhw0

Steps followed to complete the assignment

A. Auto Suggestion
1. Similar to assignment-3 , I used nodejs for this assignment as well. Following is the architecture.



Please install following modules to run my node.js application without any glitches.
Command to install : **npm install --save <module-name>**

a.express
b. fs
c. solr-client
d. deepcopy
e. unfluff

2. On top of my assignment-3, I started with getting system tuned for auto-suggestions first. I used **suggest** component of solr to achieve this. I followed instructions in offical **Reference** document in class website.

3.Once after adding that, I was getting suggestions only for single words with spaces. To make it work for multi word queries, I added logic of splitting words in a phrase and querying solr for individual word and passing results to my front-end from nodejs.

B. Spell Correction
4. Once after testing auto-suggestions feature, I started with my spell check feature. I used javascript implementation of peter norvig's algorithm,Jspell (**Reference**).

5. I got the data for learning phase by extracting content from given HTML files using Apache Tika HTML parser. I removed all duplicates and added it to existing big.txt given in Norvig's website.

6. In addition I added my own code to implementation as the one implemented it overwrote one of **Global** Javascript **object properties**. This will cause problem to all of my javascript objects in nodejs . I also tuned the implementation by adding few known spelling errors to handle few additional cases for better performance in present assignment.

7. Also as nodejs runs as an application, i.e. a seperate process, I read my whole big.txt in the start up of process and save the counts of all unique words in memory. So I didn't face any problem with nodejs server being slow like that in case of php.

8. With all the above measures I am able to get spell correction working to good extent for present test cases.

C. Snippets

9.Once after getting spell check working, I focused on final part of assignment. I.e getting snippets from html pages. For this I used a node module called "**unfluff**"(Reference) to extract different parts of html file in json format.

10. From the json formatted data returned by unfluff module, I used the content under 'text' key to split in to sentences. I iterated over each sentence and checked for my query term. If query term is present I added the sentence to my snippet string. I limited the content to 100 characters per each query term . For multi word query, I am splitting the query and checking for each word in sentences till I reach a limit of 300 characters.

D. Results

Following are five of the queries(as it is described to mention only five in description document) I used for checking spell correction:

1. NA**OT** : Corrected and got results for NATO
2. Do**J W**oens : Corrected and got results for Dow Jones
3. Po**ek**mon Go : Corrected and got results for Pokemon Go
4. Ca**il**fornia Wild Fires : Corrected and got results for California wild fires
5. Do**an**ld Trump : Corrected and got results for Donald Trump

Following are five of the queries(as it is described to mention only five in description document) I used for testing auto-completion. All five worked correctly as shown in video.

1. NATO
2. Dow jones
3. Rio olympics
4. Pokemon go
5. Brazil

In Video for the following queries I tried a different character interchange because of specified reasons..

For one letter interchange:

1. Harry Potter → Tried "Haryr Potter" as 3rd and 4th characters are same in both words.

2. Rio Olympics → Tried Rio olmypics as 3rd and 4th characters are same.

PS:

Credits to google and stackoverflow websites for helping me to get going with nodejs in specific and javascript in general, for this very first time.

Credits to Shine (https://stoi.wordpress.com/2012/1in2/31/jspell/) for implementing Peter Norvig's spell checker in javascript, which indeed I adopted in doing this assignment.