Text Mining Assignment

In this assignment we’re going to practice document similarity. Here’s what you need to do:

1. From Wikipedia’s List of musicians[1] page, pick five lists of musicians (e.g., List of big band musicians). You can pick any five you like but make sure that the list has the words “musicians” in it and that the list has at least 30 musicians listed.
2. Write a function: get_musicians(url) where url is the url of a list of musicians. The function should return a list of urls that point to articles on each musician in the list. You need to extract only musician URLs so be careful. (See note.)
3. Write a function that scrapes the text on a musician’s page and returns it as a text string. Save this string to a file. While we should be smarter about the scraping, for now just scrape all the text that is inside a <p> tag. Assuming that page_soup contains the entire bs4 object for the page, the following code should get you all the relevant text on the page

   ```
   all_text = ‘’
   for tag in page_soup.find_all(‘p’):
        all_text += tag.get_text()
   ```
4. The greater than 30*5 documents you’ve extracted is your “Learning Sample”. I suggest saving all the extracted text documents in a folder (you can then use the PlainTextCorpusReader to read them)
5. Use the Learning Sample to construct an LSI model.
6. Now go back to the Lists of musicians page and pick a sixth list (again, at least 30 members) and extract the same information from their Wikipedia articles. This is the Test Sample.
7. Use the LSI model to find the most similar musician from the learning sample for each musician in the test sample.
8. Print, in tabular form, the name of a musician from the test sample, and the name of the most similar musician from the learning sample

Turn in the following: The entire code. Make sure that everything is functionalized and saved either in a .ipynb or .py file. The TA should be able to enter a list of “list of musicians” url and call a single function to run the entire code and see the final print

[1] https://en.wikipedia.org/wiki/Lists_of_musicians

Note on extracting data from a List of Musicians

One of the problems with Wikipedia is that page structure is not always consistent. For example, if you go to a musicians list, like the List of bluegrass musicians, you'll see that each musician is in an <li> tag. Unfortunately, there are numerous other <li> tags that contain non-musician information. You have a couple of choices for how you go about extracting musicians:

1. Use the Contents section on the page. The Contents section contains navigational information. You could, for example, navigate to the header that contains "D" and then look ahead for a <ul> tag that contains the list of "D" musicians
2. Root out non-musicians by defining terms that shouldn't be in a musician link. Any link that contains words like "Category", "Help", "Special", etc. is not a candidate musician
3. Go to the musician page and see if it looks appropriate for a person. Does the "Talk" page of the linked page contain the phrase "biographies of living persons"? Does the Talk page of the linked page contain the words "Wikiproject Musicians"? Look for clues! Be creative!