### Assignment 1

# Building Inverted Positional Index and Answering Specialized Wildcard Queries

21st September 2020

**Due Date: 5th October 2020**

This assignment is on creating text corpus from html form data, building inverted positional index on that dataset and using them to answer wildcard queries. Please use python 3.x for this assignment as libraries like *nltk* will make many things easier (stop word removal and lemmatization). However, if you use any other language, you most probably have to design these modules yourselves which might not perform as good as *nltk* library in python.

## Task 1 (Loading data)

1. Download "earnings call transcripts" (ECT) indexed from 1 to 10,000 from this link using web scraper tool and save in a folder titled "ECT", which will be located in the same directory where the code is in. Tutorial (It also contains sample code), Code

## Task 2 (Building corpus)

1. For each transcript collected, remove additional structural information and create a nested dictionary with following keys and save the set of nested dictionaries as a corpus titled "ECTNestedDict" in the same directory. Tutorial Sample transcript
    a. Date
    b. Participants: List of the name of participants. For example, in sample transcript, the list contains all the names under "Company participants" and "Conference Call Participants"
    c. Presentation: It is a nested dictionary with key as speakers and value as their statements. For example, in sample transcript, example of the keys are "operator", "Caroline corner" etc and value contains the paragraph written below their name. More specifically, one such key value pair is as follows: *key: "operator", value: "Ladies and gentlemen, thank you for standing by. And welcome to the Acutus Medical Inc. Second Quarter 2020 Earnings Conference Call. At this time all participant lines are in a listen-only mode. After the speaker's presentation, there will be a question-and-answer session. [Operator Instructions] We ask that you please limit yourself to one question and one follow up. Please be advised that today's conference may be recorded. [Operator Instructions] I would now like to hand the conference over to your speaker today, Caroline Corner, Investor Relations. Please go ahead."*
    d. Questionnaire: It is a nested dictionary with key as serial of the question-answer and value is a nested dictionary of this form ("Speaker": the name of the person who is asking or answering, "Remark": corresponding remark). Hint: In the sample transcript, it starts under "Question-and-Answer Session" heading.

2. Build a text corpus titled "ECTText" from the set of collected transcripts where each transcript is regenerated as a text file by concatenating all the text information in the transcript. The ECTText should be created in the same folder with the code.

## Task 3 (Building Index)

1. Remove stop words, punctuation marks and perform lemmatization to generate tokens from the documents in the text corpus "ECTText". (use nltk library in python)
2. Build Inverted Positional Index (Dictionary with tokens as keys, and (file_name,positions) as postings)

## Task 4 (Answering wildcard queries with single * symbol)

Now write codes to use the built inverted positional index to answer different queries. The queries will be of different types as listed below. The code should take as argument the name of the query file where each line contains a query of the following type.

1. Wildcard queries with leading * symbol. For example, *mon
2. Wildcard queries with single trailing * symbol. For example, moo*
3. Wildcard queries with single *, occurring inside the string. For example, mo*n

## Important Instructions on How to write the code and How to submit

1. Submission format: You should submit a zip file , which contains four python codes for the given tasks specified (naming convention mentioned below). You should not submit any other file. Each code snippet will be executed automatically therefore please follow the naming convention. Also assume all the required folder and index are placed in the same folder with the code.
2. Naming the code file: The name of the zip file should be **ASSIGNMENT1_<ROLLNO>.zip** .The name of the code file should be in uppercase letters as below.
   **ASSIGNMENT1_<ROLLNO>_<Task_No>.py**
   e.g. :- For a student with roll no 17CS92R02, the code file name  for task 1 should be **"ASSIGNMENT1_17CS92R02_1.py".**
3. Checking code
   a. Code for tasks 1-3 should not take any argument and assume the required index or corpus is there in the same directory and it should produce all the deliverables in the same directory.
   b. Write code for task 4 which can take "query.txt" file as an argument as below.

```
$>> python code.py query.txt
```

   (query.txt file will contain many queries in the above mentioned format. There will be one query in each line. This file will remain unknown to you. Your program will be evaluated based on the results (precision & recall of the results), it produces for the queries in the above file.)

4. Saving the search results: Your code for task 4 should read the queries one by one and get the search results. At the end it should create a text file with results. The name of the results file should follow the below convention.
   **RESULTS1_<ROLLNO>.txt**
   e.g. :- **"RESULTS1_17CS92R02.txt"**
5. Python library restrictions: You can use python libraries like nltk, numpy, os, sys, collections, timeit, etc. However, you can't use libraries like lucene, elasticsearch, or any other search api. If your code is found to use any of such libraries, you will be awarded with zero marks for this assignment without any evaluation.
6. Plagiarism Rules: If your code matches (more than 60%) with another student's code, all those students whose codes match will be awarded with zero marks without any evaluation. Therefore, it is your responsibility to ensure you neither copy anyone's code nor anyone is able to copy your code.
7. Code error: If your code doesn't run or gives error while running, you will be awarded with zero mark.