National University of Computer and Emerging Sciences, Lahore Campus

	Course:	**Information Retrieval**	**Course Code:**	**CS5040**
	Program:	**MS(CS)/MS(DS)**	**Semester:**	**Spring 2022**
			Total Marks:	**70**
	Deadline:	**06-Nov-2022**		
	Section:	**MCS-1A**		
	Assessment	**Assignment - 2**		

Instruction/Notes:

1. Make sure to read the instructions carefully before attempting them.
2. You are allowed to discuss with TA for general advice.
3. You must submit your own work on Google Classroom.
4. Plagiarism will be not tolerated. Penalty will be decided later on.
5. The code must be very clear, to-the point, and presentable. We will not just be looking for correct answers rather highest grades will be awarded to write-ups that demonstrate a clear understanding of the material. Write your solutions as if you were explaining your answer to a colleague. Style matters and will be a factor in the grade.

Tasks:

Read all .txt extension files and perform the following tasks:

1. Write a function named **wordList(doc)** in such a way that it takes a txt file as input argument and returns a list of words in your document. For example, for below paragraph

   Association for Computational Linguistics 6 th Applied Natural Language Processing Conference Proceedingsnof the Conference April 29--May 4, 2000 Seattle, Washington, USA ANLP 2000-PREFACE 131 papers were submitted

   It will return list of words like this

   ['Association', 'for', 'Computational', 'Linguistics', '6', 'th', 'Applied', 'Natural', 'Language', 'Processing', 'Conference',

'Proceedingsof', 'the', 'Conference', 'April', '29-- May', '4,', '2000', 'Seattle,', 'Washington,', 'USA', 'ANLP', '2000-PREFACE', '131', 'papers', 'were', 'submitted',]

2. Write a function named **removePuncs(wordList)** that takes list of words then iterate through this list. During iteration it do some processing on each word. Function should replace punctuation marks as well as \n. and check either this word in stopword on not? if it is in stopword then we didn't append this into resulting List. You must also handle case insensitivity of words. Output for above list should be like this.

   ['association', 'computational', 'linguistics', '6', 'th','applied', 'natural', 'language', 'processing', 'conference', 'proceedingsof', 'conference', 'april', '29may', '4', '2000', 'seattle', 'washington', 'usa', 'anlp', '2000preface', '131', 'papers', 'submitted']

3. Write a function named **termFrequencyInDoc(wordList)** which should take a list of words as input argument, and output a dictionary of words such that each word that appears in the document is key in the dictionary and it's value is term frequency
4. Write a function named **wordDocFre(dicList)** that takes list of dictionary as input argument, each dictionary in this list is the word that appears in the given document as keys and the no. of times the word appears as value. This function should construct a dictionary which has all the words that appear in the corpus as keys and no. of docs that contain this word as value.
5. Construct a function named **inverseDocFre(dicList,base)** that takes dictionary returned from wordDocFre functions above and outputs inverse document frequency of each word.
6. This function named **tfidf(docList)** takes list of documents it calls the function wordList to split the document in list of words and remove stopwords and punctuation marks from them, then calls **termFrequencyInDoc()** uses its output to create dictionary of vocabulary using the function **wordDocFre**(), it then should call **inverseDocFre**() function. It then outputs a list of dictionary, where each document corresponds to the dictionary, its words should be keys values should be tf-idf score.
7. Write a code for VSM and run the following queries, you must show the top 5 documents ranked according to the score. Function header should be like this vectorSpaceModel(query)
   1. LDA
   2. Topic modelling
   3. Generative models

4. Semantic relationships between terms
5. Natural Language Processing
6. Text Mining
7. Translation model
8. Learning procedures for the lexicon
9. Semantic evaluations
10. System results and combination

Good Luck!