PES UNIVERSITY

Department of Computer Science & Engineering
Session : Jan-May, 2023

UE20CS332 – ALGORITHMS FOR INTELLIGENCE WEB AND INFORMATION RETRIEVAL

HANDS-ON SESSION : 02

TITLE : UNDERSTANDING INVERTED INDEX AND POSITIONAL POSTINGS LIST

GOAL :

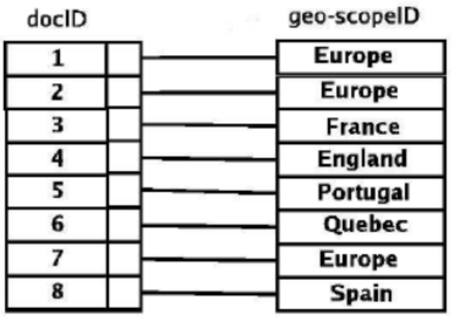
The goal of this hands-on session is to prepare an inverted index using a dictionary and postings list and put together a positional postings list.

CONCEPTS COVERED AND KEY TAKEAWAYS :

- Inverted index
  - Dictionary
  - Postings list
- Positional postings list

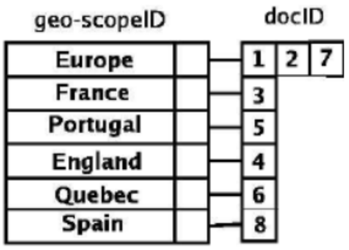
USEFUL POINTS AND LINKS :

- An inverted index is created by mapping each dictionary term to its corresponding postings list, which is a list of all the docIDs (document IDs) that it is present in.



docID	geo-scopeID
1	Europe
2	Europe
3	France
4	England
5	Portugal
6	Quebec
7	Europe
8	Spain

Forward Index



geo-scopeID	docID
Europe	1 2 7
France	3
Portugal	5
England	4
Quebec	6
Spain	8

Inverted Index

- Each postings list is sorted by the docID. This provides the basis for efficient query processing as we'll see later.
- A positional posting list is an extension of the regular posting list, where in addition to the document ID, it also stores the position of the word in the document. The format can be understood as : docID, [list_of_positions_the_word_is_present_at]
- Linguistic preprocessing tasks like stemming and lemmatization can be applied as and when necessary based on the preprocessing done on the query to the inverted index.
- Reading material :
    - A brief explanation of the inverted index
    - Introduction to inverted indexes

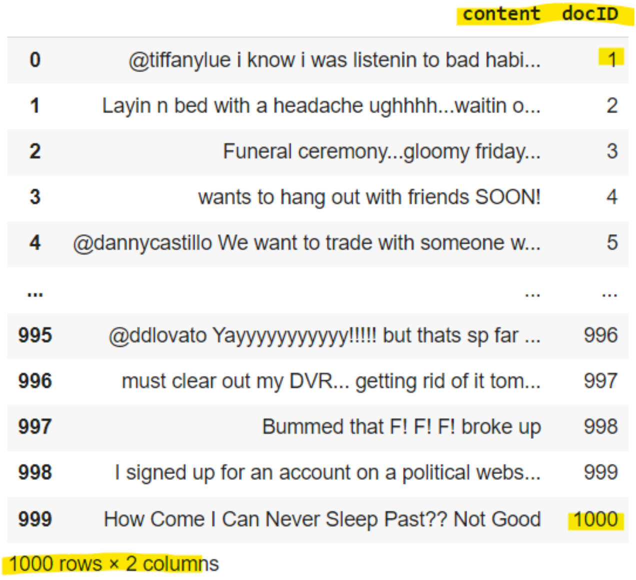
STEPS TO FOLLOW :

- Working with Kaggle : (SAME AS HANDS-ON SESSION 01)
    a. Follow the link Figure Eight Labelled Textual Dataset | Kaggle to download the dataset for this hands-on session.
    b. Download the csv file from the link mentioned above.
    c. To use the csv file as a dataframe, one of the following methods can be used :
        - Upload the csv file to the notebook working environment :

          df = pandas.read_csv("<filename>.csv")
        - Mention the path to access the csv file in the code :

          df = pandas.read_csv("<path_to_file>/<filename>.csv")
    d. The uploaded csv file is now in a dataframe format, ready to be used for the text processing tasks.
    e. The given dataset has 40,000 rows. Use the first 1000 rows for this hands-on session to avoid resource limitation. One of these methods can be used :
        - df = df.iloc[:1000] (OR)
        - df.drop(df.tail(30000).index, inplace = True)
- In this dataframe (named "df" in the above example), since the 'content' column is the focus of the task, remove all other columns using the following command :
    - df = pandas.DataFrame(df['content'])

- On the dataframe created, perform the following nlp text processing tasks on the text data in the column named “content” :
  a. Tokenization
  b. Case-folding
  c. Removal of punctuation marks, emoticons, HTML tags and links
  d. Convert numerical values to text (Ex: 10 -> “ten”)
  e. Stopword removal
- Now create a new column in the dataframe, called ‘docID’. The docID is a unique serial number (starting from 1) that can identify a document in a collection. The final dataframe looks as follows:



	content	docID
0	@tiffanylue i know i was listenin to bad habi...	1
1	Layin n bed with a headache ughhhh...waitin o...	2
2	Funeral ceremony...gloomy friday...	3
3	wants to hang out with friends SOON!	4
4	@dannycastillo We want to trade with someone w...	5
...	...	...
995	@ddlovato Yayyyyyyyyyyy!!!!! but thats sp far ...	996
996	must clear out my DVR... getting rid of it tom...	997
997	Bummed that F! F! F! broke up	998
998	I signed up for an account on a political webs...	999
999	How Come I Can Never Sleep Past?? Not Good	1000

1000 rows × 2 columns

This image is for reference ONLY, to understand the schema of the final dataframe. The preprocessing has NOT been shown in the above sample image. The values in the ‘content’ column MUST be preprocessed as mentioned in the previous step.

- Prepare the inverted index and print it as the output for this task.
  - Inverted index = Dictionary terms + Postings lists
- Extend the postings list prepared in the above step to make a positional postings list and print the same.
  - Refer to the reading material shared for theoretical understanding of inverted index and positional postings list.

NOTE :

- We require you to appropriately document your code using the Markdown feature for each different text processing task.The first cell must be a markdown cell which contains the following details :
    a. UE20CS332 : Algorithms For Intelligence Web And Information Retrieval
    b. SRN : PES1UG20CSXXX
    c. Name : Dua Lipa
    d. Section : X

SUBMISSION LINK FOR HANDS ON - 02:

https://forms.gle/ksPcb2Lb6hRRRGMh8

Format for evaluation: **PES1UG20XXX_HandsOn02.ipynb**

- **DEADLINE : END OF DAY**
- **ANY SUBMISSION POST THE DEADLINE WON'T BE CONSIDERED**
- **ENSURE THE FIRST CELL OF THE NOTEBOOK IS AS MENTIONED ABOVE**

TA CONTACT DETAILS:

- Abhay D A: abhayda2001@gmail.com
- Neha Angadi: nehaangadi19@gmail.com
- Priya S S: sspriya@pesu.pes.edu