Information Ritrieval Invert Index

Yasin Boloorchi

August 17, 2021

1 Introduction

One of the first steps of information retrieval is to make an invert index out of each document of the given data. this document is a report of homework 3 of the information retrieval class. in this homework, we are going to make an invert index of the "Opin Rank Car Dataset" which been given to us in the homework.

2 Dataset

In the start step of this homework, first, we download the dataset from the given URL and we take a look at it. this dataset includes 3 years of comments about a lot of cars. each year is separated in different folders. each year's folder contains a lot of files. each file name contains the year, mode, and the name of that car which is separated with an underline.

3 Implementation

3.1 Libraries

For this exercise, I choose to write my parser and for that, I used two built-in libraries of python in "my parser.py", which are subprocess and "re". for finding and reading each file I used "subprocess" that gives me the ability to find and move thru folders and find the path of each file and their names. and for parsing and finding the correct pattern in each file. further, in the "parse car dataset" I use JSON built-in library to convert invert index dictionary to JSON file and save it for further data analysis. in the "InvertIndex" code file I just used the "PorterStemmer" from "NLTK" for stemming tokenized words. other libraries that I use are "pandas", "numpy" and "matplotlib". "pandas" is for reading data and manage it the high speed and matplotlib is used for plot and drawing plots for better showing the statics of this dataset.

3.2 Parsing

My parser code is a module that I wrote for this exercise. this program can get a series of file paths in an array, parse them and finally return a python dictionary object of them. it's a simple program that opens each file in the given paths array and takes their name, then for each line in that file finds the comment and

all the tags in it. finally, it will make a key value for each comment (document) and save them with DocID and return them.

3.3 Pars Car Dataset

We can say that the "pars car dataset" is a part of the parsing process but we made another program so that we use "my parser" as an external module. this program is even simpler than "my parser", all that it does is to find file paths and give it to "my parser", and make a JSON file from the returned parsed dictionary.

3.4 InvertIndex

In the last python code "InvertIndex", we made an Invert index file with the parsed file. this program's work is a little bit complicated. in this code, for each key in the parsed dictionary, we read the values. it takes 'TEXT' and 'FAVORITE' tags and merges them, then for each word in that merged string, we check the inverted index that we made. if the word wasn't in invert index, then it will and the word and also the document id to invert index; if the word was in invert index, then it will increase the Document frequency, and then finally if it was in invert index and also in the current document, it will increase Text frequency in each document. In the end, it will create a json file and save the invert index dictionary to it.

4 Inverte Index Data Structure

The data structure that is used for the invert index is a python dictionary. in this dictionary, each word is a key to its values. for each word, we save "DocIDs", "DocTF" and DF. DocID is a dictionary of Documents that this word was in them. DocTF is another dictionary in the word dictionary that its keys are DocIDs and their values are the number of times that this word was used in them. the last value in each word dictionary is DF (Document Frequency) that is the number of documents that the word was in them.

5 Analysis

At last, all of our analysis has been done in a jupyter notebook and will be with this document. on that jupyter notebook, you can see all the required analysis and their code.