DIJOUD Fanny
210821

IS NLP deliverable Homework

28/01/2022

Table des matières

I. Problem to solve

This homework is about keywords' research/requests.

The users' issue is: Users have a folder with PDF documents. Users want to know what their PDF documents, that are into a folder, are about without reading all of them. Especially, users want to know if documents are linked to a specific word.

To respond to this users' issue, we need to create a program that have two fonctionalities. First, the program has to extract some keywords from PDF and saves them into a file. Moreover the program has to highlight that a specific word is linked to some specific PDF. Second, the program has to respond to a user keywords' request by saying if some PDF are linked to the seeked keyword.

II. Experiment(s) done

The program uses package 'tm' that is in the NLP package.

II.1 Create

For the « create » functionality (means the functionality that creates keywords and saves them in a file) :

The program takes one argument: the path of the folder which contains all the PDF documents.

The first idear was to check and to analyse each PDF individually to find keywords by using some tokenization stuff. But It looks more efficient to user a corpus analyse. Indeed, with the corpus, the user can choose the variable called “keyPresentNbDocument” to have more freedom about the way keywords are selected. Indeed the default value of “keyPresentNbDocument” is number of documents divides by 2 (line 23 code). That means that keywords need to be in at least half of documents of the corpus. If “keyPresentNbDocument” is 1 that means that each document are treated as individual. If the user wants to change this value and puts “keyPresentNbDocument” as 2. That means that keywords have to be in at least two documents of the corpus. So keywords are more specific to the corpus. Like that, user can have keywords for each PDF or keywords for a corpus.

Of course, keywords are cleaned with removePunctuation, stopwords, tolower, stemming and removeNumbers.

After that, we have a list a keywords. To clean it more, I added a minimal occurrence. Indeed, the user can change the variable called “keyOccurrence” which is the occurrence of a keyword. This varible is set to 10 by default (line 18 code).

Finally, keywords are saved into a xlsx file: keywords are lines and PDF names are columns, numbers are occurrence. The file is into the same folder that PDF are in.

II.2 Search request

For the « find » functionality (means the functionality that searches keywords into saved ones) :

The program takes two arguments: the path of the folder which contains all the PDF documents and the keyword.

First, the seeked keyword has to be cleaned to be easily found. So we use tolower, stemDocument, removeNumbers. Then, cleaned keyword is searched in the xlsx file. If keyword is in it, the program prints all columns’ name (means PDF’s name) where occurency is higher than 0 in the keyword’s line.

III. Analysis of results

The relevance of saved keywords depends on the chosen variable “keyOccurrence”: if “keyOccurrence” is too low, some keywords will not be relevant. But if the value is too high, maybe there will be no keywords available.

IV. Github

https://github.com/DIJOUDFanny/UPM-Intelligent_systems-NLP.git