

# Documentation Utilisateur

## I- Fonctions Search Engine

Pour commencer, nous avons tout d'abord transformé le fichier csv en Dictionnaire avec la transcription en clé et l'url en valeur. (**csv\_to\_dict**)

Ensuite, nous avons créé 3 index différents ayant chacun une utilité :

\_ Le **premier (index\_words\_to\_url)** est un dictionnaire de mots en clé et leur url en valeur. On enlève tous les caractères non-alphanumériques dans les transcriptions d'abord, on les 'split' et enfin on les enlève les majuscules pour que le moteur de recherche soit insensible à la casse.

Puis pour chaque mot de la transcription nous l'ajoutons en clé de l'index et nous ajoutons l'url en valeur.

\_ Le **deuxième (index\_words\_to\_idf)** associe à chaque mot son score IDF.

\_ Le **troisième (index\_urls\_to\_words)** est en fait un dictionnaire ayant pour clé les urls et pour valeur un autre dictionnaire contenant les mots pour clé et le score TF pour valeur.

Puis, pour associer les Urls au score TF-IDF en fonction du ou des mots-clés entrés, nous avons créé une fonction préliminaire **find\_a\_word** qui prend les 3 index et un mot-clé et qui va retourner un dictionnaire avec les urls en clé et les scores en valeur.

Cette fonction va servir à la fonction **find\_some\_words** qui va retourner un dictionnaire avec en clé le mot-clé et en valeur le dictionnaire retourné par **find\_a\_word**. La fonction va d'abord récupérer les mots entrés en input puis les 'split' pour appeler la fonction **find\_a\_word** sur chaque mot.

Enfin, la fonction qui va permettre d'entrer plusieurs mots-clés et de prendre l'intersection des Urls des mots-clés entrés est la fonction **intersection\_of\_results**, il ne retourne

## II - Fonctions Interface Graphique

Concernant l'interface graphique, nous avons utilisé la bibliothèque graphique Tkinter. La première partie qui se trouve avant le code principal correspond à l'initialisation de l'interface graphique, de la barre de navigation latéral, de la création du label devant la zone de texte et la zone de texte où l'utilisateur va rentrer ses mots-clés :

```
##### Initialisation Interface Graphique #####

#Creation of the graphic interface
root = Tk()

#Scrollbar creation
scrollbar = Scrollbar(root)
scrollbar.pack( side = RIGHT, fill = Y )

#Label Creation
L1 = Label(root, text="Your keywords : ")
L1.pack( side = LEFT)

#Entry to enter and get the keywords
keywords = StringVar()
keywordInput = Entry(root, textvar=keywords, bd = 5)
keywordInput.pack(side = LEFT)
```

Ensuite, plusieurs fonctions vont permettre de faire le lien entre l'interface et les fonctions du moteur de recherche.

La première est **weblink** qui va vérifier que les résultats de la recherche sont des url et va permettre de cliquer dessus avec la bibliothèque *webbrowser*.

Plus encore, la fonction **printUrl** va permettre de supprimer chaque '\n' et d'afficher chaque résultat dans la ListBox qui est créée un peu plus tard.

Enfin la fonction **deleteUrl** va permettre de supprimer toutes les url dans la ListBox et de refaire une recherche sans relancer les index.

```
##### Fonction de l'Interface Graphique #####

#Check if it's an url and open the link when you click on it
def weblink(*args):
    index = lb.curselection()[0]
    item = lb.get(index)
    if 'https://' in item:
        webbrowser.open_new(item)

#Insert The url in the list
def printUrl():
    tupleUrlTFIDF = intersection_of_results(find_some_words(index1, index2, index3))
    listeUrl = []
    for tupleUrl in tupleUrlTFIDF :
        listeUrl.append(tupleUrl[0])

    for url in listeUrl :
        url = url.rstrip('\n')
        lb.insert(END, url)

#Clear the list and the input entry
def deleteUrl() :
    lb.delete(0, END)
    keywordInput.delete(0, END)
```

En toute fin de code, se trouve la fin de l'initialisation de l'interface graphique avec la création du bouton 'Enter' qui va lancer l'affichage des url. Le bouton 'Delete' qui va effacer les url et la création de la ListBox où toutes les urls vont se trouver.

```
##### Fin Initialisation Interface Graphique #####

#Creation of a button that will submit the keywords
button= Button(root, text="Enter", command= printUrl)
button.pack(side = LEFT)

buttonDelete = Button(root, text="Do another research", command= deleteUrl)
buttonDelete.pack(side=LEFT)

#Creation of a ListBox where all the urls will be
lb = Listbox(root, yscrollcommand = scrollbar.set)
lb.bind('<<ListboxSelect>>', weblink)
lb.pack(fill=BOTH, expand=True)

root.mainloop()
```

### III - Deuxième Version sans Interface Graphique

Nous avons créé une deuxième version exactement similaire à la première mais sans interface graphique, ce choix s'explique car la bibliothèque Tkinter ne fonctionnait pas sur un de nos deux. C'est pour cela que, par sécurité, nous avons créé une deuxième version.

La seule différence est qu'il y a une boucle de recherche dans la console, et ensuite cela marche exactement pareil. Il suffit de rentrer **exit** en mots-clés pour quitter la boucle d'interaction.

ATTENTION : Vous devez modifier le chemin ligne 14 afin de l'adapter à votre machine :  
`doc=open('/Users/julienesbt/Documents/Etudes/M1/BDA/TP/TP1/TED_transcripts.csv', 'r')`