Technique d’indexation et RI

Plan:

- Introduction
- Objectif
- Apercu du Projet
- Parcours du Code

Introduction:

- Le projet consiste à écrire un programme en langage Python permettant de réaliser des tâches précises sur un corpus de documents

Objectif:

- L'indexation des documents , definir les terms et calculer leur (tf) et (pg) puis stocker les dans un fichier.
- récupérer une requête de l'utilisateur et calculer les poids de ses terms

Apercu du Projet:

- Le pojet fait une extraction de chaque document , recuperer ses terms et calculer le (tf) puis il les stocker dans un dossier.Apres,

  il calcul le (pg) de chaque term et stocker dans un autre dossier.

  Enfin, l'utilisateur ecrit une requete et le system calcul les poids de chaque term de la requete et l'afficher

Parcours du Code:

```python
def extract_document(corpus):
    document=""
    i=0
    while(i<len(corpus)-1):

        if(corpus[i]=="#" and document!=""):
            corpus=corpus[i:]
            return document,corpus
        if(corpus[i]!="#"):
            document+=corpus[i]
        i+=1
        if(corpus[i]==""):
            break
```

```python
def extract_term(end_word):
    word=""
    term=[]
    i=0;j=0
    while(i<(len(document))):
        if((document[i] in end_word) |(document[i]==len(document)-1)):
            if(not(word in stopliste)):
                term.append(word)
            word=""
        else:
            word+=document[i]
        i=i+1
    return term
```

Dans cet etape on fait l'extraction des documents puis les terms

```python
def term_count(df,trm):
    for i in range(len(df)):
        c=0
        for j in range(len(trm)):
            if(df[i]==trm[j]):
                c+=1
        df_count[i]=c
    return df_count
```

```python
def write_file_terms(term,min_freq,max_freq,f1,tf,n):
    df=[]

    i=0
    while(i<len(term)):
        f1 = open("terms.txt","a")
        if(min_freq<term_freq(term[i],term)<max_freq):
            ch=term[i]+" "+"document:"+str(n)+" " +"terme :"+str(term_freq(term[i],term))+"\n"
            f1.write(ch)
            df.append(term[i])
            tf.append(term_freq(term[i],term))
            remove_occ(term[i])
        i+=1

    f1.close
    return df
```

Dans cet etape on fait calculer les (tf) puis on les stocker dans le fichier terme.txt

```python
def document_freq(df):
    d_df={}
    for i in range (len(df)):
        d_df[df[i]]=term_freq(df[i],df)
        remove_occ(df[i])
    return d_df
```

```python
def pond_global(pd,num_documents):
    idf = 1/(math.log10(num_documents / (pd) ))
    return idf
```

```python
pd_g=[0]*len(df0)
i=0
for i in range (len(df0)):
    f2= open("pond_global.txt","a")
    pd_g[i]=pond_global(pd[i],num_documents)
    ch=df0[i]+" "+" " +"pond_global : "+str(pd_g)+"\n"
    f2.write(ch)
```

Dans cet etape on fait calculer les (pg) et les stocker dans un fichier pond_global.txt

```
ch=""
rq=[]
req=str(input("saisir requete:")).lower()
f1= open("terms.txt","r")
f2= open("pond_global.txt","r")
req=req+" "
for i in range(len(req)):
    if(req[i]==" "):
        if(not(ch in stopliste)):
            rq.append(ch+" ")
        ch=""
    else:
        ch+=req[i]
```

```
for i in range(len(rq)):
    k=0
    with open("terms.txt","r") as file:
        for line in file:
            line1=line.lower()

            if (rq[i] in line1 and rq[i][0]==line1[0]):
                print(line1)
                for j in range(len(df0)):
                    if (df0[j]+" "==rq[i]):
                        print(rq[i]," poids: ",pd_g[j]*tf[k])
            k+=1
```

Dans cet etape l'utilisateur va saisir la requete et on va extract les terms de la requete puis faire une recherche de (tf) et (pg) ,enfin on calcul le poid et afficher la resultat

Merci a votre attention