Rapport du projet

Travail pour le cours « Acquisition,modélisation des connaissances»

Professeurs : Frédérique Segond / François Stuck

Siyu **Wang**

Natia **DAVITASHVILI**

Master 2 TAL

Institut National des Langues et Civilisations Orientales

2019-2020

Table des matières

Introduction

L'objet du projet final consiste à réaliser en python

- L'indexation 'incrémentale' d'un corpus bilingue français / anglais
- L'interrogation booléenne de ce corpus par mots-clefs

Alors, avant de commencer nous vous précisons que dans notre dossier vous trouverez: (nom de dossier souligné)

- corpusExamenCorrige
- documentsIndexes
- indexerDocuments.py
- incrementationIndex.py
- requirements.txt
- requetesCorpus.py
- traitementCommun.py
- readme.txt
- Rapport.pdf

L'importance de chaque éléments de cette liste sera évoquée dans les propos suivants.

Corpus

Pour effectuer des opérations nous avons utilisé le corpus nommé « corpusExamenCorrige » qui se trouve dans notre dossier « Projet». Il s'agit de 32 articles au format xml, parus dans le Monde Diplomatique entre 2001 et 2003. Ce corpus se divise en deux parties :

• initiaux : 30 textes à indexer dans un premier temps.
• complémentaires : 2 textes à ajouter à l'indexation précédente.

Nous avons 4 éléments xml qui structurent les articles avec : titre, auteur, texte, notes. L'indexation des documents concerne seulement les éléments « titre » et « texte », et puis l'indexation des termes traite tous les termes de tous les documents.

Installation

Pour installer les outils requis : (modules python: langdetect, six, et treetaggerwrapper), on passe la commande suivante :

pip3 install -r requirements.txt

Cela installera automatiquement tous les modules de la bonne version pour faire éxecuter le programme, et le module treetaggerwrapper permet d'utiliser le treetagger directement sans changer le langage de la programmation.

Première partie : Indexation incrémentale bilingue

Tout d'abord pour commencer l'indexation des documents xml dans le dossier, on lance le script :

python3 indexerDocuments.py <chemin de dossier> comme ceci :

```
python3 indexerDocuments.py corpusExamenCorrige/initiaux
```

Attention : Le script indexerDocuments.py ne doit être utilisé qu'une seul fois pour le corpus "initiaux", pour tout corpus complémentaire, il faut les incrémenter avec le script incrementationIndex.py .

Cela nous permettra de parcourir tous les fichiers et d'effectuer pour chacun d'eux le traitement pour obtenir plusieurs dictionnaires : un avec l'id d'article comme une clé et le titre du document comme une valeur, un avec l'id d'article comme une clé et un dictionnaire

des fréquences comme une valeur, et puis un (pour chaque langue) avec le terme comme clé et un dictionnaire d'article id et sa fréquence comme valeur.

Tous les fonctions communes utilisées par les 3 scripts indexerDocuments.py, incrementationIndex.py, et requetesCorpus.py sont stockées dans le script nommé traitementCummun.py comme un module pour effectuer : gestion de fichiers txt, traitement treetagger, lemmatisation, tokenisation, mise en minuscule, suppression de signes de ponctuations, lecture et écriture du format json, etc. Sachant que les jeux de tags sont différents selon les langues, nous avons utilisé donc langue.detect en python.

Alors on obtient des fichiers d'index au format json dans le dossier ./documentsIndexes. Pour l'incrémentation d'indexation d'un autre dossier on utilise le script suivant :

python3 incrementationIndex.py <chemin de dossier> comme ceci :

```
python3 incrementationIndex.py corpusExamenCorrige/initiaux
```

Si dans le corpus complémentaire il y a un fichier qui se trouve dans la liste des documents déjà indexés, on ne le traite pas pour éviter la répétition d'indexation; alors que s'il n'est pas dans la liste, on l'indexe et mettre à jour les fichiers json d'indexation. Dans le terminal il y aura des messages affichés si le document est indexé ou pas. C'est justement pour cet objetif qu'on a ce script.

En résultat on a créé des fichiers d'indexation de sortie en format .json.
Au total, nous avons 5 fichiers sauvegardés :

1) IndexDocs.json – un index de tous les documents avec leurs id et leurs titres
2) IndexTermesEN.json – un fichier d'indexation de termes anglais. Pour chaque terme, nous avons l'id du document qui contient ce terme et leur fréquence trouvée dans ce document
3) IndexTermesFR.json - un fichier d'indexation de termes français. Pour chaque terme, nous avons l'id du document qui contient ce terme et leur fréquence trouvé dans ce document

4) index_par_document_fr.json – un fichier d'indexation de document -> termes. Pour chaque id du document, nous listons tous les termes du document avec leur fréquence (français).
5) index_par_document_en.json – un fichier d'indexation de document -> termes. Pour chaque id du document, nous listons tous les termes du document avec leur fréquence (anglais).

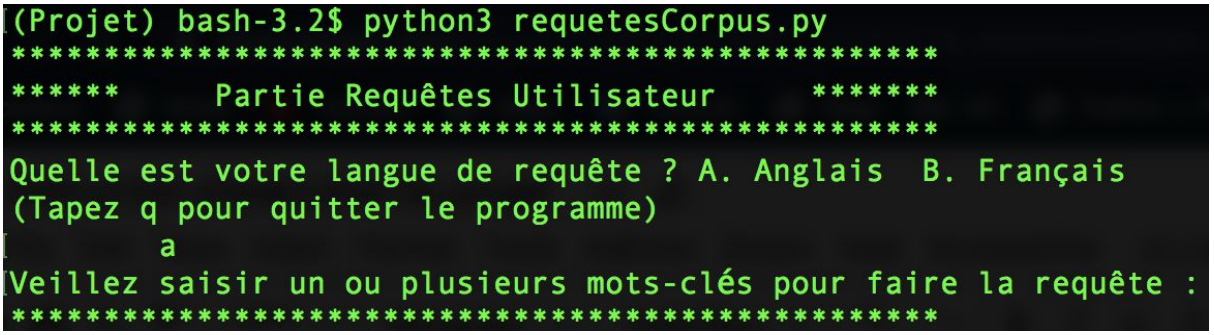
Les 5 fichiers se trouvent dans le dossier sous le nom « **documentsIndexes** » du dossier principal « Projet » .

Deuxième partie : Requête par mots-clés

Pour effectuer les requêtes on lance le script suivant :

python3 requetesCorpus.py

Et on saisi le choix de langue pour la requête, puis les mots clés:



Le résultat affiche d'abord le nombre de documents trouvé et puis, pour chaque document, son score et le titre du document.

Pour calculer le score de pertinence, nous avons calculé la fréquence absolue de termes, et nous avons pris en compte plusieurs cas pour les requêtes.

- S'il n'y pas de mot optionnel, et que nous avons seulement des mots obligatoires, alors dans ce cas-là, on calcule la fréquence de tous les termes et ensuite l'article avec la fréquence maximale nous donnera les meilleurs résultats.

Par exemple, la requête sans mot optionnel : *+france -italy*

```
**********************************************
Veillez saisir un ou plusieurs mots-clés pour faire la requête : +france -italy
**********************************************
------------------------------------------------
Nombre de documents trouvés : 5
ID : corpusExamenCorrige/initiaux/103060-article.txt     Score : 100     Titre : THE PRICE OF EU ACCESSION
ID : corpusExamenCorrige/initiaux/103433-article.txt     Score : 98      Titre : THREATS TO DISARMAMENT AND INTERNATIONAL SECURITY
ID : corpusExamenCorrige/initiaux/103200-article.txt     Score : 96      Titre : ISRAEL VERSUS PALESTINE VERSUS THE WORLD
ID : corpusExamenCorrige/initiaux/103342-article.txt     Score : 94      Titre : INSECURE, EXASPERATED AND BORED WITH THE OLD POLITICS
ID : corpusExamenCorrige/initiaux/103448-article.txt     Score : 92      Titre : THE FUTURE OF A MARGINALISED CONTINENT
**********************************************
```

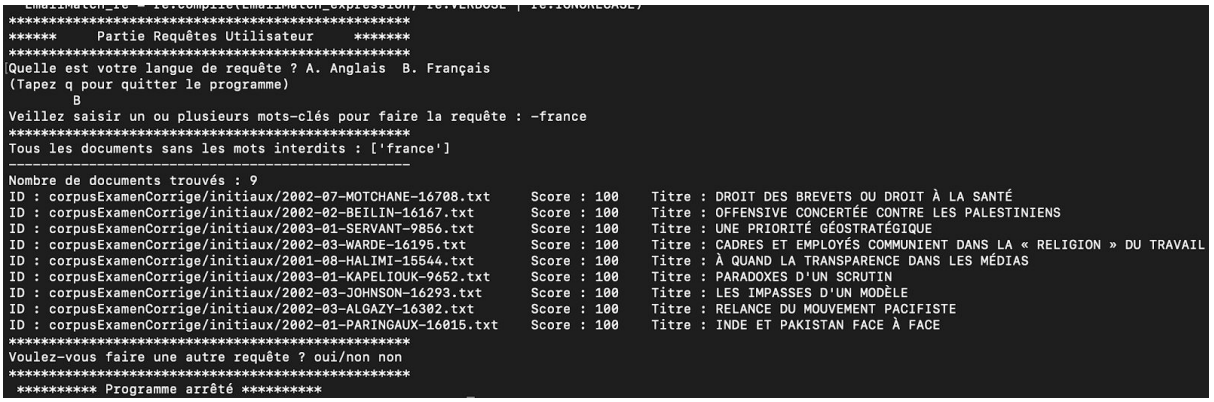
- Dans le cas où l'on a le mot optionnel, on calcule la fréquence des mots obligatoires et aussi les optionnels, mais on met d'avantage de mots obligatoires.

Et la requête avec mot optionnel : *+france italy*

```
**********************************************
Veillez saisir un ou plusieurs mots-clés pour faire la requête : +france italy
**********************************************
------------------------------------------------
Nombre de documents trouvés : 9
ID : corpusExamenCorrige/complémentaires/103687-article.txt      Score : 100     Titre : COLLATERAL DAMAGE FROM AN ILLEGAL WAR
ID : corpusExamenCorrige/initiaux/103674-article.txt     Score : 98      Titre : GLOBAL CRISIS OVER IRAQ
ID : corpusExamenCorrige/initiaux/103168-article.txt     Score : 96      Titre : MEDIA OWNERSHIP AND BIAS
ID : corpusExamenCorrige/initiaux/103060-article.txt     Score : 94      Titre : THE PRICE OF EU ACCESSION
ID : corpusExamenCorrige/initiaux/103167-article.txt     Score : 92      Titre : AMERICAN BURBS
ID : corpusExamenCorrige/initiaux/103433-article.txt     Score : 90      Titre : THREATS TO DISARMAMENT AND INTERNATIONAL SECURITY
ID : corpusExamenCorrige/initiaux/103200-article.txt     Score : 88      Titre : ISRAEL VERSUS PALESTINE VERSUS THE WORLD
ID : corpusExamenCorrige/initiaux/103342-article.txt     Score : 86      Titre : INSECURE, EXASPERATED AND BORED WITH THE OLD POLITICS
ID : corpusExamenCorrige/initiaux/103448-article.txt     Score : 84      Titre : THE FUTURE OF A MARGINALISED CONTINENT
**********************************************
```

- Nous avons aussi ajouté une autre fonctionnalité : si la requête ne contient que les mots négatifs, et évidemment dans ce cas-là il est impossible de calculer une pertinence variée, il renvoie donc tous les documents sans les mots interdits avec un score de 100. Donc il n'y a pas de différence de score.

Par exemple, passons la requête : *-france*



Celui-là va renvoyer vers toutes les requête sans *-france*, on peut dire que c'est une fonctionnalité pour pouvoir traiter tous les types de requêtes.

Conclusion

Enfin, nous pouvons dire que nous avons pu voir d'abord, l'indexation inversé et aussi 'incrémentale' -c'est-à-dire indexer une seule fois sans les doublons, de notre corpus bilingue français / anglais.

Ensuite nous avons réussi à faire un système de requêtes - interrogations booléennes par mots-clés, avec un ou plusieurs termes et conditions . Cela est très utile pour le milieu de TAL.