Moteurs de recherche

M2 IAD - Parcours Pro

TME 1 - Indexation

1 Introduction

L'objectif de ce TME est de réaliser l'indexation de corpus de documents textuels, en effectuant un ensemble de traitements standards. La performance d'un modèle de recherche, en terme de temps de réponse ou de pertinence des résultats, est fortement liée à l'étape d'indexation. Il est donc nécessaire de savoir effectuer le traitement du texte brut de collections réelles.
A la fin du TME, vous devez être capable de créer l'index inversé d'une collection et de fournir des réponses à des requêtes booléennes simples (par exemple composées uniquement de ET ou de OU). Vous pouvez programmer dansle langage de votre choix. Il est cependant fortement conseiller d'utiliser des langages efficaces pour le traitement de chaînes de caractères.

2 Présentation des collections de documents

L'ensemble des données est disponible à l'adresse suivante:
`http://ir.dcs.gla.ac.uk/resources/test_collections/cisi/`
ainsi que: `http://ir.dcs.gla.ac.uk/resources/test_collections/cacm/`.

2.1 Fichiers contenus dans ces bases

La base CISI contient quatre fichiers: `CISI.ALL` contient le corpus de document CISI (1460 documents); `CISI.QRY` contient 112 requêtes textuelles; `CISI.BLN` contient 35 requêtes booléennes qui sont des réécritures des 35 premières requêtes de `CISI.QRY`, et `CISI.REL` contient des jugements de pertinence pour les requêtes. Pour l'instant, vous n'avez à utiliser que le fichier `CISI.ALL`, les autres fichiers seront utilisés ultérieurement pour l'évaluation de vos modèles de recherche.
La base CACM contient 6 fichiers: `cacm.all` contient le texte des documents, `cite.info` contient des informations supplémentaires (que nous ignorerons), `common_words` est un anti-dictionnaire que vous pourrez utiliser (sur les bases CISI et CACM) lors des pré-

traitements; `query.text` contient 64 requêtes textuelles; `qrels.text` contient les jugements de pertinence correspondant. Enfin, le fichier `README` contient des information sur les fichiers précédents. Une fois encore, pour l'instant, uniquement le fichier `cacm.all` est utilisé.

2.2 Format des fichiers

Les fichiers des deux bases suivent le même format. Les textes des documents sont donnés dans les fichiers ".all". Dans ces fichiers, chaque document est séparé par une balise `.I` (suivie de l'identificant du document) qui contient le titre du documents (balise `.T`), les auteurs (balise `.A`), la date de publication (balise `.B`), l'abstract (balise `.W`), des mots-clefs (balise `.K`) ainsi que d'autres informations que nous ignorerons (en particulier des informations par rapport aux articles cités dans la balise `.X`). Certaines informations (par exemple abstract ou mots-clefs) peuvent ne pas être disponibles pour certains documents.

Bien que vous n'en ayez pas besoin pour l'instant, je vous donne aussi une description des formats des fichiers qui seront utilisés pour l'évaluation.

Les fichiers contenant les requêtes vectorielles (`CISI.QRY` et `query.text`) ont le format suivant: Chaque requête est précédée d'une balise `.I` (suivi de l'identifiant de la requête), le texte de la requête est contenu dans la balise `.W`, la balise `.N` donne l'auteur de la requête (à ignorer). Enfin, certaines requêtes sont données avec une balise `.A` qui représente des auteurs spécifiques pour les articles à renvoyer. Le contenu de cette balise peut être pris en compte.

Le fichier `CISI.BLN`, qui contient les requêtes booléennes pour la base CISI, a un format clair. Nous donnons ici un exemple de requête:

```
#q1= #and ('titles', #or ('automatically', 'retrieving', 'problems',
                          'concerns', 'descriptive',
                          'approximate', 'difficulties',
                          'content', 'relevance','articles'));
```

Chaque requête commence par `#q[identifiant de requête]=`, et finit par un point virgule. Deux requêtes ayant un identifiant identique entre les fichiers `CISI.BLN` et `CISI.QRY` correspondent à la même requête, et utilisent donc les mêmes jugements de pertinence.

3 Prétraitement de la collection

Pour l'instant, nous ne considérerons pas la sémantique des balises. Elles pourront cependant être utilisées ultérieurement. Nous allons considérer les prétraitements suivants pour l'établissement du dictionnaire:

- Pour chaque document, supprimez le contenu de toutes les balises sauf `.T`, `.A`, `.W` et `.B`. Supprimez ces balises de façon à ce que chaque document contienne tout le texte contenu dans ces balises sans distrinction de la balise d'origine.
- Considérez tous les caractères non alphanumériques comme des séparateurs de mots, puis supprimez les.

- Supprimez la casse en mettant tous les mots en minuscule.
- Supprimez les mots appartenant au fichier `common_words`.

Quels sont les avantages et les inconvénients de tels prétraitements pour la recherche ? Vous pourrez vous appuyer sur les exemples suivants:

- `mreames@cs.wisc.edu`
- `100,000.52 $`
- `C++`
- `Bill Gates`

4 Indexation

Vous pouvez maintenant effectuer l'indexation, et particulièrement l'index inversé. L'index inversé doit vous permettre de répondre rapidement à des requêtes booléennes simples. Vous ferez attention à utiliser les structures de données appropriées (pour le stockage du dictionnaire et de la matrice). En particulier, vous pourrez vous poser les questions suivantes:

- Quelle est la complexité de la recherche de l'identifiant d'un mot du dictionnaire ?
- Quelle est la mémoire utilisée pour stocker la collection, en fonction (1) du nombre de documents de la collection et (2) du nombre moyen de mots différents par document ?

Une fois l'indexation faite, vous pouvez la tester en effectuant quelques requêtes booléennes simples (conjonction ou disjonctions de quelques mots). Vous avez alors un premier moteur de recherche (très simple...).