Dr. Nastaran FATEMI
Florian POULIN

Recherche de documents

Laboratoire Nº 3

A. Objectifs

Ce laboratoire a pour but de continuer le moteur d'indexation pour la collection CACM implémenté au laboratoire n°2 et de réaliser un outil de recherche plus performant.

Les points étudiés dans ce laboratoire seront :

1. Pondération des termes
2. Implémentation du modèle vectoriel

B. Références

Cours «Recherche d'Information Multimédia » de Nastaran Fatemi.

C. Rapport

A remettre : Vos sources et un rapport comportant les objectifs, la description des démarches adoptées, l'analyse des résultats obtenus et une conclusion personnelle.

A remettre : Au plus tard au début de la séance du 21 Novembre 2007.

D. Donnée

1. Mise à jour du framework

Pour réaliser ce laboratoire, vous devrez commencer par **mettre à jour certains fichiers du framework** distribué pour le laboratoire n°2. Les modifications à apporter sont les suivantes :

- Remplacer l'interface *Indexer* par la nouvelle version, fournie avec la donnée ;
- Remplacer la classe *CACMFeeder* par la nouvelle version, également fournie ;
- Remplacer l'interface *Retriever* par la nouvelle version, également fournie ;

La nouvelle interface *Indexer* définit une méthode supplémentaire, *finalizeIndexation()*, invoquée par le nouveau *CACMFeeder* lorsque le parsing de la collection CACM est terminé. La nouvelle interface *Retriever* contient une nouvelle méthode, *executeQuery(String)*. Les types de retour des méthodes de l'ancienne version ont été adaptés pour être conforme avec les pondérations introduites dans ce laboratoire.

2. Pondération des termes dans les documents

Dans ce laboratoire vos devrez **implémenter deux types de pondération** pour le contenu de la collection CACM. Pour chaque pondération, vous devez créer une nouvelle version du fichier d'indexation et du fichier inversé où les poids des termes sont déjà convertis.

- Pondération par fréquence normalisée

$$w(t_i) = freq_i / \max_j(freq_j)$$

- Pondération par tf*idf normalisée

$$tfidf(t_i) = \log_2(freq_i + 1) \cdot \log_2(N / n_i)$$
$$w(t_i) = tfidf(t_i) / \max_j(tfidf(t_j))$$

où $freq_i$ est la fréquence d'occurrences du terme t_i dans un document, N est le nombre de documents dans la collection, et n_i le nombre des documents qui contiennent le terme t_i, et $max_j( )$ est la valeur maximale de la pondération dans un document.

Comme dans le laboratoire précédent, il vous est demandé de **réaliser un export des index** (quatre au total) dans des fichiers texte en respectant le format habituel.

3. Modèle vectoriel

Vous devez implémenter la méthode *executeQuery( )* de la classe *CACMRetriever*, qui prend en paramètre une chaîne de caractère (requête) et qui retourne une liste de documents, associés à leur similarité avec la requête.

Pour ce faire, vous devez **implémenter le modèle vectoriel avec le calcul de similarité par cosinus**. La requête sera entrée sous une forme libre et sera traitée comme pour les documents (suppression des mots vides, ponctuation, etc.).

Votre programme doit donc accepter une requête et fournir une liste ordonnée de documents comme réponse (**en faisant apparaître les valeurs de similarité cosinus**). Il devra en outre laisser le choix à l'utilisateur sur le fichier d'index à utiliser pour la recherche (et donc quelle pondération utiliser).

E. Indications

- Pour la requête vectorielle, on garde simplement la fréquence brute des termes comme pondération (pas de normalisation !) ;
- Vous n'êtes pas obligés de dupliquer les index, vous pouvez stocker plusieurs pondérations dans un même index (ce n'est pas obligatoire) ;
- Pour le rapport, faites apparaître clairement les quatre points demandés. Construisez vos idées, soyez précis et soignez l'orthographe ;
- En cas de doute, n'hésitez pas à poser des questions à l'assistant ou au professeur.