Projet de RIW

Antoine Apollis, Marine Sobas et Paul Viossat

L'objectif du projet est le développement d'un moteur de recherche pour la collection de documents "Stanford". Celle-ci comporte des documents sous différents formats (pdf, php, html..) en rapport avec les enseignants dispensés à l'université. Pour développer l'outil, nous avons comparé différents modèles, les plus classiques comme les modèles booléen ou vectoriel, mais aussi des méthodes plus sophistiquées comme le modèle de langue. Dans ce rapport, nous décrivons la démarche adoptée pour construire un tel outil.

Choix des modèles

L'enjeu fondamental du projet est le choix d'un modèle adapté. Dans cette partie, nous expliquons nos critères de choix après avoir présenté les modèles mobilisés. Nous avons mis en place des recherches booléennes et vectorielles, ainsi qu'un modèle de langue. Comme nous n'avions pas de corpus d'entraînement, nous avons évité les modèles qui reposent sur l'apprentissage.

Recherche booléenne

Nous avons implémenté la recherche booléenne à l'aide d'un index inversé, ce qui donne des résultats corrects sur les requêtes d'évaluation, bien que pas parfaits, notamment sur la requête 4 (« very cool »), qui ne renvoie **aucun** résultat.

Cependant, le modèle booléen ne détermine pas un classement entre les résultats ce qui est problématique dans le cadre d'une recherche parmi un corpus de documents divers.

Modèle vectoriel

Le modèle vectoriel que nous avons mis en place permet de déterminer un rang parmi les documents pertinents. Nous avons mis en place différents types de pondération. Les documents sont pondérés par :

- la fréquence
- le TF IDF normalisé

- le TF IDF logarithmique
- le TF IDF normalisé logarithmique

Les requêtes sont pondérées par la fréquence.

On obtient ainsi des classements de documents. Il est facile d'obtenir un recall de 1, mais il est bien plus compliqué d'avoir des résultats précis.

Cependant, le modèle vectoriel suppose que les termes sont indépendants les uns des autres, ce qui est très simplificateur. C'est pourquoi, l'usage d'un modèle de langue permet d'améliorer nos performances.

Modèle de langue

Nous avons également implémenté un modèle de langue assez simple utilisant un filtrage de Dirichlet. Ce modèle se base sur la probabilité que la requête soit générée par un document et classe les documents par ordre décroissant de probabilité.

Le filtrage de Dirichlet permet d'éviter des probabilités nulles quand un mot absent du vocabulaire du document est présent dans la requête en les substituant par les probabilités d'apparition du mot dans le modèle de langue du corpus, modulées par un facteur.

Le facteur est modulé par un paramètre arbitraire μ et la taille du document.

Comparaison des modèles

Afin de comparer les modèles, nous avons calculé la mean average precision pour 11 valeurs interpolées. Les résultats sont les suivant :

	Mean average precision, K=11
Modèles vectoriel : Fréquence	0.568
Modèles vectoriel : TF IDF normalisé	0.557
Modèles vectoriel : TF IDF logarithmique	0.313

Modèles vectoriel :TF IDF normalisé logarithmique	0.582
Modèle booléen	0.551
Modèle de langue	0.165

La pondération TF IDF normalisé et logarithmique pour le modèle vectoriel semble être la méthode la plus performante sur les quatre requêtes.

Structure de données

Pour indexer la collection, nous avons opté pour un index inversé de position, stockant la **racine** des mots (en utilisant l'algorithme de Porter). Cet index est stocké sur le disque après la première génération, et occupe 459 Mo.

Analyse statistique de la collection

La collection comporte 98 998 documents, et 18 244 627 mots, après retrait des mots vides. Un document compte en moyenne 184 mots. Le plus long document contient 963 mots, le plus court n'en contient aucun après transformation. L'écart type du nombre de mots par document est de 127, ce qui montre que les documents sont globalement assez courts .

Le vocabulaire de la collection comporte 162 076 mots. Un document dispose en moyenne de 26 termes différents. La fréquence maximale d'un terme dans un document est de 27. Un mot apparaît en moyenne 2,6 fois par document.

Conclusion

A travers ce projet nous avons pu mettre en oeuvre différentes méthodes de recherche d'information dans un corpus. Nous avons pu découvrir les enjeux de cette

discipline, que ce soit au niveau de la préparation des données, du choix de l'algorithme utilisé ou encore de l'évaluation des résultats.

Nos résultats nous montrent que des modèles simples donnent de bons résultats comparé à des modèles plus avancés qui demandent plus de temps de réglage des hyper paramètres. Nous n'avons pas pu inclure de page ranking dans nos modèles car nous n'avions pas à disposition les liens entre les pages. Il aurait été intéressant de compléter nos modèles avec cette méthode.