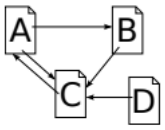
TME 4 - Algorithme de PageRank

Exercice 1 – Exercice de compréhension

On considère le graphe du TD :



Q 1.1 Stocker le graphe dans un dictionnaire (pour ressembler à votre structure d'indexation) et calculer de façon itérative les scores pour chacun des noeuds (on considérera $d = 0.85$ et $a_j = 1$). Rappel de la formule :

$$PR^t(j) = d \sum_{i \to j} \frac{PR^{t-1}(i)}{Out(i)} + (1-d)a_j \tag{1}$$

Exercice 2 – Projet

1 Indexation de documents structurés

Afin d'être à même de pouvoir considérer les relations entre documents du corpus considéré, les hyperliens présents dans ceux-ci doivent être extraits au moment de l'indexation de la collection. Il s'agit alors dans un premier temps de reprendre le processus d'indexation défini en TME1 pour y ajouter une prise en compte des relations entre documents. Selon l'architecture définie, il suffit de reprendre la méthode du parser spécifique à nos collections pour y inclure la considération des éléments de la balise .X et d'ajouter ces informations dans l'index. Chaque ligne de la balise .X contient plusieurs éléments. Seul le premier nous intéresse : il correspond à l'identifiant d'une page pointée par un lien du document courant.

Q 2.1 Ecrire le code permettant l'indexation des liens hypertexte.

Q 2.2 Implémenter la méthode `getHyperlinksTo` qui permettent de récupérer les documents qui citent un document donné en paramètre.

Q 2.3 Implémenter la méthode `getHyperlinksFrom` qui permettent de récupérer les documents cités par un document donné en paramètre.

Remarque : Pour le TP, vous pouvez tester sur une sous-collection (1000 premiers documents) pour aller plus vite.

2 Application de l'algorithme de PageRank

L'algorithme PageRank peut selon les cas, être appliqué au graphe de la collection entière (dans ce cas on pourra utiliser leurs scores combinés à des scores dépendant de la requête) ou bien être utilisés directement comme modèles de recherche appliqués à un graphe résultant d'une recherche préliminaire sur la base d'une requête.

Il s'agit ici de mettre en oeuvre cette deuxième approche pour définir des modèles de recherche documentaire basés sur ces algorithmes à base de marche aléatoire sur des graphes.

Q 2.4 Pour mettre en place ce genre de modèle, la première étape est alors la détermination d'un sous-graphe de documents candidats. Il s'agit alors pour chaque requête Q formulée par l'utilisateur de déterminer le sous-graphe $G_Q = (V_Q, E_Q) \subseteq G = (V, E)$ des documents candidats sur lequel nous allons lancer un algorithme de marche aléatoire. Cela se fait en deux temps :

- Initialisation de V_Q avec un ensemble S de documents "seeds" : les n premiers documents retournés par un modèle sur le contenu donné ;
- Pour chaque document $D \in S$, ajout dans V_Q de tous les documents pointés par D et de k documents choisis aléatoirement parmi ceux pointant vers D.

Trois paramètres sont donc à définir pour mettre en place le modèle :

- Le modèle de base permettant de récupérer les documents seeds ;
- Le paramètre n déterminant le nombre de documents seeds à considérer ;
- Le paramètre k déterminant le nombre de liens entrants à considérer pour chaque document seed.

Q 2.5 Une fois le sous-graphe extrait pour la requête de l'utilisateur, il s'agit d'appliquer l'algorithme de PageRank et ordonner les documents selon l'importance qui leur a été attribuée dans ce contexte.

3 Bonus

Q 2.6 Apprendre la valeur optimale du damping factor d en réalisant une cross-validation sur la MAP. (Comme pour l'optimisation des paramètres BM25, etc...).