Stockage

Accès et Recherche d'Information

1 Stockage

Les systèmes de recherche d'information stockent généralement le vocabulaire d'une collection en mémoire principale, le but étant de minimiser le temps de réponse à une requête (temps qui dépend grandement du nombre d'accès disque que ces systèmes effectuent lors de leur phase de recherche). La structure de données la plus simple pour stocker le vocabulaire est de trier d'abord ses termes par ordre alphabétique et de les stocker ensuite dans un tableau contenant des entrées de taille fixe. La taille des entrées est normalement choisie en fonction du nombre de caractères du terme le plus long. On alloue aussi un espace supplémentaire pour chaque terme afin de stocker son df et un espace pour le pointeur vers la liste des identifiants de documents contenant le terme. Cette structure est schématisée à la figure 1.

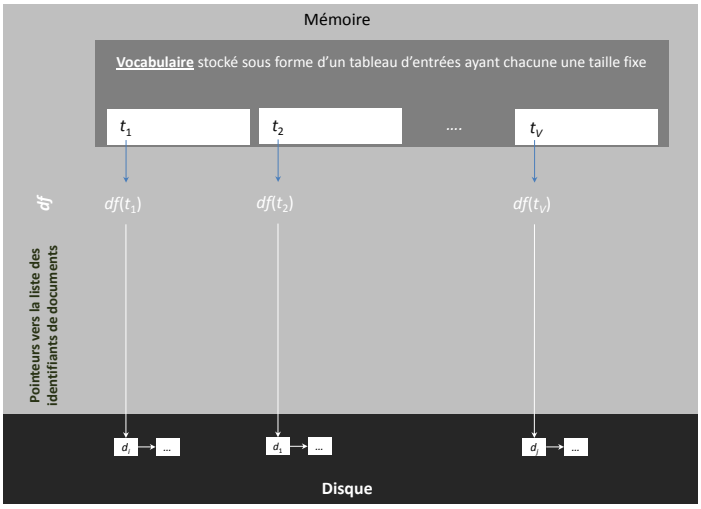

Figure 1: Stockage du vocabulaire dans un tableau de taille fixe pour chaque entrée.

1.1 Calculer la taille de l'espace nécessaire pour stocker le vocabulaire de la collection du Wikipédia français suivant le schéma précédent.Le terme le plus long de cette collec-

tion *Taumata-whakatangihanga-koauau-o-tamatea-turi-pukaka-piki-maungah-oronuku-pokai-whenuaki-tanatahu*
(nom d'une colline en Nouvelle-Zélande), est composé de 97 caractères. On suppose aussi qu'un caractère est codé sur 1 octet et que les df et les pointeurs sont respectivement codés sur 3 et 4 octets.

La longueur moyenne des termes de la collection de Wikipédia est de 9 caractères. Utiliser une entrée de taille fixe pour stocker chaque terme nous fait donc, en moyenne, perdre 88 octets par terme. Une solution simple serait de conserver le vocabulaire dans une longue chaîne de caractères. Pour séparer les termes entre eux, on utilise souvent un pointeur supplémentaire vers les termes comme illustré à la figure 2.

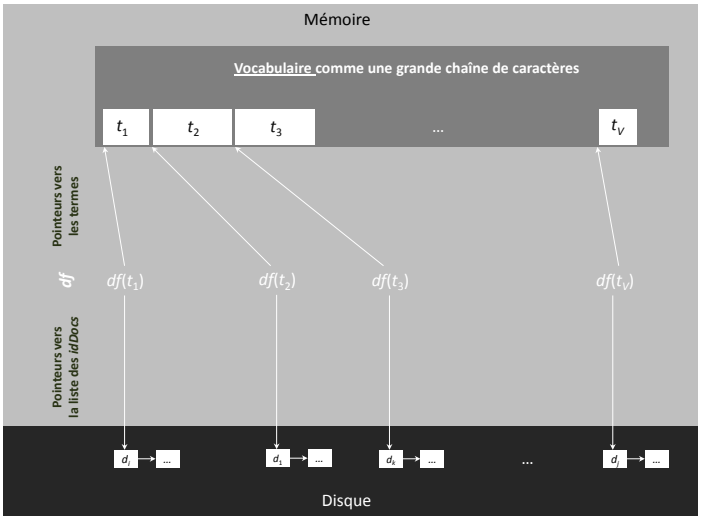

Figure 2: Stockage du vocabulaire dans une chaîne de caractères.

1.2 Quel est l'espace nécessaire pour stocker le vocabulaire suivant ce nouveau schéma ? On suppose que la taille de chaque pointeur de terme est de 3 octets.On peut aller plus loin en groupant les termes par bloc de taille fixe, k. L'avantage de ce groupement est qu'on aura moins de pointeurs de termes à stocker ($3 \times (k-1)$ octets de moins dans notre cas), mais il faudra néanmoins séparer les termes présents dans chaque bloc par un caractère spécial codé sur 1 octet.

1.3 Quel sera la taille en octets du vocabulaire si on préconise cette stratégie ? Par quel facteur aura-t-on comprimé le vocabulaire si on utilisait cette structure de données à la place de la première stratégie ?

2 Modèle Booléen

On considère une collection contenant les documents suivants:

d_1 : énergie hydraulique géothermie biomasse
d_2 : fossiles pétrole gaz énergie
d_3 : énergie nucléaire fission
d_4 : énergie conservation Joules physique

2.1 Quelle est la représentation par index inversé de cette collection?

2.2 Quels sont les résultats de recherche retournés pour les requêtes suivantes dans le cas où on utiliserait un moteur de recherche booléen?

q_1 : énergie
q_2 : énergie `SAUF` (pétrole `OU` hydraulique `OU` nucléaire)

3 Modèle vectoriel

On considère la collection de documents $\mathcal{C} = \{d_1, d_2, d_3, d_4, d_5, d_6\}$ suivante

- d_1 = (football corner)
- d_2 = (sport football penalty)
- d_3 = (sport sport coupe rugby collectif)
- d_4 = (sport coupe coupe volleyball)
- d_5 = (volleyball rugby coupe collectif)
- d_6 = (coupe sport sport)

3.1 Quel est le vocabulaire associé à la collection?

3.2 Donner les représentations vectorielles des documents suivant les pondérations binaire et fréquentielle.

3.3 Soit la requête suivante q = (football sport collectif sport coupe football sport). Calculer les scores produits scalaires entre les documents de la collection et la requête d'après les pondérations binaire et fréquentielle considérées.

4 Augmentation de la couverture des termes

On considère une requête q contenant les termes *OS, Jaguar* et trois documents de même taille d_1, d_2 et d_3 qui contiennent respectivement *Jaguar, Jaguar, Jungle, Jungle, Jungle*, et *Système d'exploitation, Jaguar, Mac, Système d'exploitation, Système d'exploitation*

et *Jaguar, Bentley, Mercedes, Jaguar, Jaguar.* On prend l'abréviation S.E. pour *Système d'exploitation* et on suppose que le vocabulaire associé est :

$$\mathcal{V} = \{bentley, jaguar, jungle, mac, mercedes, os, S.E.\}$$

4.1 Donner les vecteurs associés aux documents et à la requête. Dans le cas où on privilégie une représentation à base de tf, ordonner les documents par rapport à leur score *produit scalaire* avec la requête.*Jaguar* est un terme polysémique et on voit bien sur l'exemple précédent que si un terme polysémique d'une requête est répété plusieurs fois dans des documents traitants d'autres sujets que ce que l'on recherche, ces documents obtiendront un meilleur score que ceux traitant du sujet mais contenant moins d'occurrences de ce terme polysémique. Une solution est d'augmenter la couverture des termes du vocabulaire en prenant en compte, dans la représentation vectorielle des documents et de la requête, les termes synonymes des termes apparaissant dans les documents et la requête. Un moyen simple pour cela consiste à définir une matrice de similarité W entre les termes et de projeter les documents et la requête sur cette matrice avant de calculer leurs scores. Pour notre exemple, considérons la matrice de similarité entre termes suivante:

$$\begin{array}{c} \\ B \\ Ja \\ Ju \\ Mac \\ Mer \\ OS \\ S.E. \end{array} \begin{array}{c} \begin{array}{ccccccc} B & Ja & Ju & Mac & Mer & OS & S.E. \end{array} \\ \begin{pmatrix} 0,5 & 0,1 & 0 & 0 & 0,4 & 0 & 0 \\ 0,1 & 0,5 & 0,05 & 0,05 & 0,1 & 0,1 & 0,1 \\ 0 & 0,05 & 0,95 & 0 & 0 & 0 & 0 \\ 0 & 0,05 & 0 & 0,8 & 0 & 0,05 & 0,1 \\ 0,4 & 0,1 & 0 & 0 & 0,5 & 0 & 0 \\ 0 & 0,1 & 0 & 0,05 & 0 & 0,55 & 0,3 \\ 0 & 0,1 & 0 & 0,1 & 0 & 0,3 & 0,5 \end{pmatrix} \end{array}$$

4.2 Quelles sont les nouvelles représentations des documents d_1, d_2 et d_3 que de la requête q? Calculer les nouveaux scores produits scalaires entre ces documents et q et ordonner ces derniers par rapport à ces scores. Conclure.

4.3 Si on suppose que les termes qui apparaissent dans les mêmes documents avec les mêmes fréquences sont sémantiquement similaires, donner un moyen simple de calculer la matrice de similarité entre termes, W.