Chapitre II. Fouille de données textuelles et recherche d'information

1. Introduction

Depuis l'invention des ordinateurs, les hommes sont à la recherche d'une manière efficace de gérer, stocker, diffuser ou de rechercher l'information. Plusieurs méthodes et techniques de gestion et de traitement d'information ont été développées. Aujourd'hui, nous pouvons estimer que nous sommes à un haut niveau d'informatisation grâce au développement et à la maîtrise des technologies de la communication et du traitement de l'information dont l'Internet est un exemple édifiant. Malgré cette évolution, la progression des moyens de recherche efficaces est encore insuffisante dans le domaine de l'information documentaire, plus spécifiquement dans celui du traitement et de la dissémination de l'information textuelle [Harrag, 2005]. Pour combler ces besoins générés par cette évolution rapide, une nouvelle industrie est en train de naître: la fouille de données textuelles ou «*Text Mining*». Cette dernière est l'outil le plus communément offert aux utilisateurs pour accéder à de grandes collections de données. C'est un ensemble de systèmes qui s'intéressent à l'analyse et au traitement des informations textuelles. Dans cette perspective, le système qui s'occupe des problèmes d'indexation automatique et de recherche d'information est connu sous le nom de Système de Recherche d'Information (SRI).

La deuxième section de ce chapitre traite les concepts, les méthodes et les applications liés au domaine de la fouille de données textuelles. La troisième section est totalement consacrée à la description du processus de la RI et à la définition les notions d'indexation, d'appariement et de reformulation de requêtes. Dans cette section, nous passons en revue les principaux modèles de RI. Nous décrivons la reformulation de requêtes et nous présentons les diverses mesures utilisées pour évaluer ces modèles et systèmes. La quatrième section est réservée à la conclusion du chapitre.

2. Fouille de données textuelles

L'accès à l'information textuelle a motivé depuis de nombreuses années les travaux de chercheurs issus de différentes communautés, comme les linguistes, les informaticiens et les statisticiens. Ces dernières années, ce domaine a connu une évolution rapide, avec en particulier le développement de grandes bases de données textuelles et du web. En effet, 80 % des informations conservées par les organismes sont sous forme de textes tels que les fiches de centres d'appel, les e-mails, les enquêtes de satisfaction, les lettres de réclamation, etc.). La fouille de données textuelles est la branche de la fouille de données qui offre des moyens capables de sélectionner, d'analyser, et d'extraire les formations textuelles non structurées en langage naturel.

2.1. Définition de la fouille de données textuelles

La fouille de données textuelles (ou forage de texte) que l'on peut traduire de l'anglais par *«Text Mining»* nous permet de déterminer le sens d'un texte sans nécessairement en lire tout le contenu dans le but de découvrir des informations cachées ou prendre automatiquement la bonne décision. D'une manière plus précise, la fouille de données textuelles désigne l'ensemble des techniques et des méthodes destinées au traitement automatique de données textuelles non structurées, disponibles sous forme informatique, en assez grande quantité. Il s'agit de les organiser et de les structurer afin d'en dégager des thématiques, des relations dans une perspective d'analyse non littéraire rapide.

Schématiquement, on peut énoncer :
Fouille De Texte = Linguistique + Fouille de Données

2.2. Processus de la fouille de données textuelles

Le processus de fouille de textes reprend les étapes du processus de fouille de données et il en ajoute d'autres pour les adapter à son objectif tel que le type de données à analyser (données non structurées ou semi-structurées). Quelques systèmes de fouille de données textuelles ont pour objectif de structurer le contenu des textes en découvrant des modèles pour les décrire. Ils se basent sur l'hypothèse d'une catégorisation *a priori* où il s'agit d'un prétraitement manuel des textes afin d'en extraire un certain nombre d'attributs comme les mots-clés ou les URL. Une fois les attributs extraits, les méthodes classiques de la fouille de données, telles que l'analyse statistique, les règles d'association, etc., sont appliquées. Le processus de fouille de textes est schématisé dans la figure 2.1.

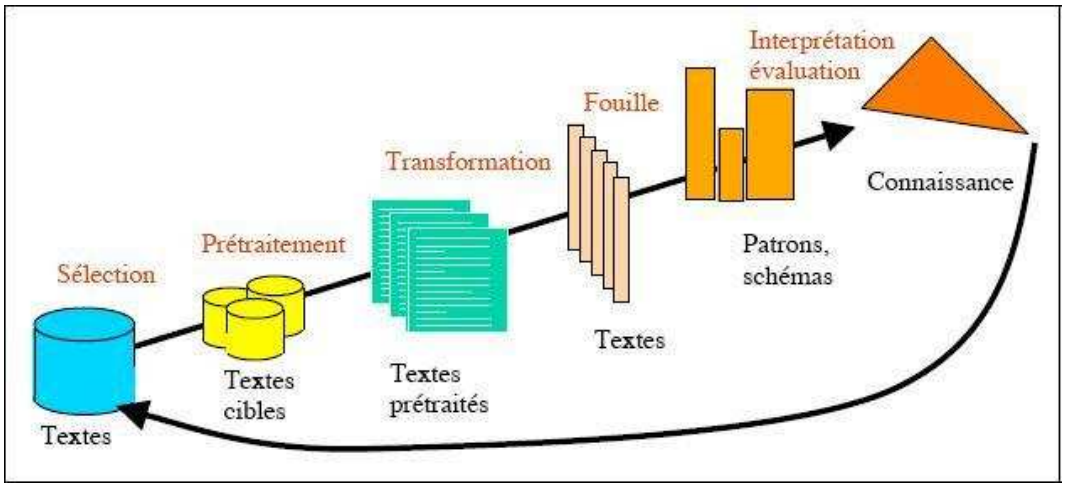

Fig. 2.1. Processus de la fouille de données textuelles [Cherfi, 2004]

2.3. Objectifs de la fouille de données textuelles

La fouille de données textuelle peut être utilisée en particulier dans les cas suivants:

- Pour mieux comprendre le positionnement d'un discours, d'une thèse, d'un communiqué.
- Pour appréhender les thèmes récurrents qui sont associés à une activité, une entreprise ou des concurrents.
- Pour mesurer les points faibles et les points forts dans une revue de presse.
- Pour comparer des textes sur un même thème afin d'en déterminer les points communs ou au contraire de distinguer les différences stylistiques.
- Pour créer automatiquement des répertoires de sites Web ou emails associés à des thématiques.
- Pour quantifier un texte ou les parties d'un texte pour en extraire les structures signifiantes les plus fortes telles que le résumé automatique et la segmentation thématique.
- Pour établir des liens entre les termes et les documents utilisés dans l'indexation.
- Pour établir des règles de classification automatique de documents (classification supervisée ou non supervisée).

2.4. Méthodes de la fouille de données textuelles

On distingue deux types de méthodes :

- Les méthodes prédictives cherchant des règles permettant d'affecter automatiquement un document à un thème, parmi plusieurs thèmes prédéfinis.
- Les méthodes descriptives permettent la recherche des thèmes abordés dans un ensemble de documents, sans connaître à l'avance ces thèmes. Une classification du corpus s'effectue selon des thèmes non prédéfinis et peut être suivie d'une extraction automatique des mots-clés ou une segmentation thématique de textes qui permet de structurer les documents suivant leurs thèmes.

2.5. Domaines d'applications de la fouille de données textuelles

Les applications de la fouille de données textuelles sont multiples : d'une simple indexation pour les moteurs de recherche à l'extraction de connaissances dans des documents non structurés. La liste suivante illustre les applications les plus connues de la fouille de données textuelles:

2.5.1. Le traitement automatique de la langue « TAL»

Depuis une vingtaine d'années, et avec la généralisation de l'outil informatique et d'internet, les applications du TAL au sens large du terme se multiplient dans les disciplines philologiques. Le TAL est une discipline à la frontière de la linguistique et de l'informatique. Le TAL concerne l'application de programmes et techniques informatiques à tous les aspects du langage humain. On distingue donc usuellement cinq niveaux linguistiques de profondeurs successifs d'analyse automatique des langues: segmentation de phrase «analyse lexicale», décodage acoustico-phonétique ou «traitement de parole», analyse morphologique, analyse sémantique et analyse pragmatique.

2.5.2. L'extraction d'information

Il est possible d'envisager des systèmes permettant l'identification précise de l'information pertinente. Ceci peut être effectué à l'intérieur d'un corpus homogènes où chaque texte est censé contenir des informations dont le type est prédéfini (problématique typique de l'extraction d'information). Ou d'une manière plus large, dans des corpus hétérogènes directement interrogés par des questions posées par l'utilisateur, le système devra renvoyer des passages de texte répondant précisément à ces questions (problématique de type question/réponse).

L'extraction d'information «RI» consiste en l'alimentation d'une base de données structurée à partir de données exprimées en langage naturel. Il s'agit de détecter dans le texte en langage naturel les mots correspondant à chaque champ de la base de données. L'analyse est donc locale et l'extraction de l'information est plus complexe, car elle nécessite d'effectuer une analyse lexicale et une analyse morpho-syntaxique pour reconnaître les constituants du texte (phrases, mots, verbes, adjectifs) qui permettent de détecter les phrases pertinentes pour l'extraction [Stéphane, 2005].

2.5.3. Le filtrage et la structuration des documents

Beaucoup de gestionnaires de courriers électroniques sont maintenant livrés avec un filtre anti-spam. Il existe aussi des logiciels anti-spam qui s'interfacent entre le serveur de courrier et le gestionnaire de courrier. Le système mondial d'interception des communications privées et publiques *Echelon* est un exemple d'utilisation militaire et économique de la fouille de textes. De nombreuses méthodes, essentiellement issues du domaine de traitement automatique de la langue, permettent de pallier ce problème. La plus couramment utilisée consiste à rechercher, et si nécessaire à filtrer, les mots-clés ou les phrases-clés des documents, et éventuellement les

relations existant entre ces divers éléments clés. Mais, il est également possible de faire des résumés de documents ou d'effectuer une catégorisation ou un clustering [Sanjuan, 2004].

3. Recherche d'information

La recherche d'information «RI» s'intéresse aux documents dans leur globalité et aux thèmes qu'ils abordent, pour comparer les documents et détecter des typologies. Elle cherche à détecter tous les thèmes présents. L'analyse est donc globale, en se plaçant au niveau du document, de nombreuses techniques permettent l'interrogation de larges corpus de données *via* une indexation de ces derniers, puis le calcul d'une mesure de similarité adéquate permettra de renvoyer les documents les plus pertinents pour une requête donnée. La section suivante sera totalement consacrée à la description des systèmes de recherche d'information. Le reste de ce chapitre sera totalement consacrée à la description des concepts, modèles et techniques liés à la recherche d'information.

3.1. Le processus de RI

Le processus de recherche d'information est le processus qui permet de mettre en relation l'ensemble des informations disponibles d'une part et les besoins de l'utilisateur d'une autre part. L'expression de ces besoins se fait par le biais de requêtes. Ces requêtes seront envoyées au système de recherche d'information afin d'extraire les documents pertinents répondant au besoin de l'utilisateur. Cette notion de pertinence est fortement subjective, car elle dépend de l'utilisateur, donc très difficile à automatiser.

Le processus de recherche d'information comprend plusieurs concepts :

- La collection de documents ou corpus;
- Le besoin en information ;
- La fonction d'indexation ;
- La fonction d'appariement requête-document ;
- La fonction de modification de requête qui se traduit généralement par un mécanisme de reformulation de requêtes.

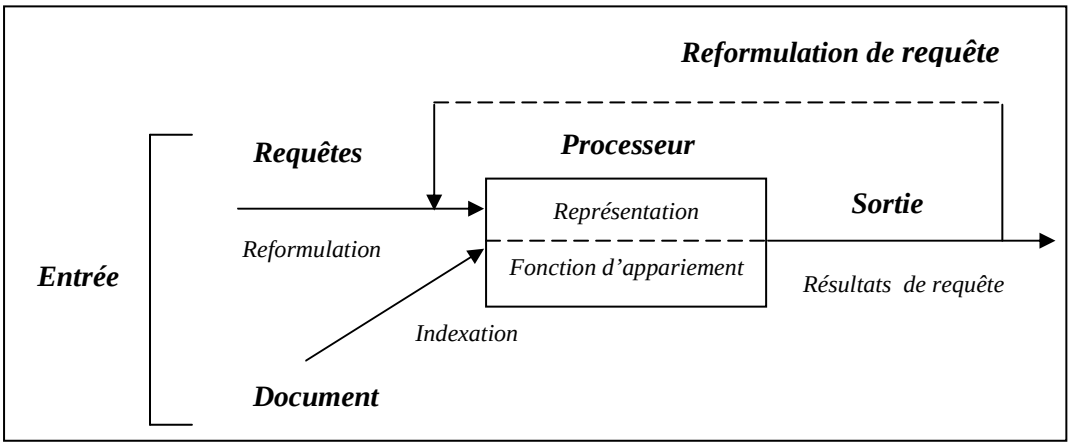

Fig. 2.2. Un système de recherche d'information typique [Rijsbergen, 1979].

3.1.1. Collection de documents (corpus)

Un corpus de documents est un ensemble de granules documentaires qui peuvent être des documents entiers ou bien des parties de documents. Dans la RI traditionnelle, l'unité d'information utilisée et recherchée lors du processus de recherche est le document.

3.1.2. Besoin en information

Ce besoin est l'expression mentale de ce que l'utilisateur recherche. L'expression d'un besoin se fait par une requête qui permet l'interrogation d'un système de recherche d'information. L'interrogation d'un SRI peut prendre plusieurs formes, à savoir : interrogation en langage naturel, interrogation en langage booléen ou interrogation graphique. Selon [Kleinberg, 1999], il existe trois différentes formes de requêtes:

- Requêtes spécifiques,
- Requêtes larges,
- Requêtes par similarité.

D'une manière générale, les requêtes sont composées d'un ensemble de mots clés. Ces mots clés peuvent être reliés entre eux par des opérateurs booléens, comme ils peuvent être aussi organisés sous forme d'expressions. Un autre type de requêtes est celui spécifique à la RI structurée. Ce dernier type prend en compte en plus de l'information textuelle des contraintes de structure.

3.1.3. La fonction d'indexation

L'indexation est le processus permettant de créer une représentation des documents et des requêtes facilement manipulable par un système de recherche d'information. Elle consiste à analyser les documents afin d'extraire un ensemble de mots clés servant comme descripteurs des documents. Il existe trois types d'indexation:

- Indexation manuelle: L'extraction et le choix des descripteurs s'effectuent par un documentaliste ou un spécialiste du domaine.
- Indexation automatique: L'extraction et le choix des descripteurs s'effectuent d'une façon totalement automatisée.
- Indexation semi-automatique: L'extraction des descripteurs s'effectue par le système et le choix des descripteurs est laissé au spécialiste.

Une étude comparative entre l'indexation automatique et l'indexation manuelle a été réalisée par Anderson et Perez-Carballo [Anderson et Pérez-Carballo, 2001]. Le résultat de l'étude montre que les avantages et les inconvénients de chacune des deux méthodes d'indexation ont une tendance à s'équilibrer. Autrement dit, le choix de l'une ou de l'autre est en fonction du domaine, de la collection et de l'application considérés. Dans les sections suivantes nous allons décrire en détail les différentes étapes de l'indexation automatique : de l'extraction des mots à partir des documents jusqu'à la création de l'index.

a. L'analyse lexicale

L'étape de l'analyse lexicale permet d'extraire l'ensemble des termes appartenant à un document. Cette extraction est effectuée en tenant compte des espaces de séparation entre mots, des chiffres et des ponctuations. Un terme peut être un mot simple (pomme) ou un mot composé (pomme de terre). Cependant, en RI on utilise souvent les mots simples.

b. L'élimination des mots vides

Les mots vides sont des mots peu significatifs et porteurs de peu de sens, augmentant ainsi la taille de l'index et rendant la recherche plus lente. De ce fait, leur élimination est impérative. L'élimination des mots vides est importante dans la mesure où elle représente un facteur qui a une grande influence sur la précision de la recherche. En effet, le fait de ne pas éliminer les mots vides provoque inévitablement du bruit, c'est-a-dire les documents non pertinents à la requête. On distingue deux techniques pour éliminer les mots vides:

- L'utilisation d'une liste de mots vides.
- L'élimination des mots dépassant un certain nombre d'occurrences dans la collection.

c. La lemmatisation

La lemmatisation est l'opération qui consiste à réduire les formes fléchies des mots à leur racine grammaticale. Car un mot donné peut avoir plusieurs formes dans un texte dont le sens est presque similaire. Plusieurs méthodes de lemmatisation ont été proposées dans la littérature [Frakes, 1992], parmi lesquelles: la troncature, la méthode des n-grammes [Adamson et Boreham, 1974], les dictionnaires ou l'élimination des affixes (exemple : algorithme de Porter [Porter, 1980]). L'étape de lemmatisation permet d'éviter à l'utilisateur de faire introduire les différentes formes d'un mot (exemple : pluriel, singulier,…) lors d'une recherche. La lemmatisation permet ainsi d'augmenter le rappel, ce qui diminue en pratique le taux de précision. Ceci est dû à la perte de la sémantique originale du mot lors du passage à la forme finale (canonique ou fléchie).

d. La pondération des termes

Cette étape permet de mesurer l'importance d'un terme dans un document. Elle a pour but de trouver les meilleurs termes représentant le contenu d'un document. L'importance d'un terme est généralement mesurée par des méthodes statistiques (et quelques fois linguistiques). La plupart des formules de pondération proposées dans la littérature de RI se basent sur deux facteurs, à savoir la pondération locale et la pondération globale. La première quantifie la représentativité locale d'un terme dans le document (on parle de fréquence du terme ou *TF*: *Term Frequency*), et la deuxième quantifie la représentativité du terme vis-à-vis de la collection des documents (on parle de fréquence inverse de document ou *IDF*: *Inverse of Document Frequency*).

- *TF* (*Term Frequency*): cette mesure indique l'importance du terme dans le document. Cette importance est proportionnelle à la fréquence du terme. Plusieurs formules de pondération locale ont été proposées, parmi lesquelles : la fonction brute (nombre d'occurrences), la fonction binaire, la fonction logarithmique et la fonction normalisée.

- *IDF* (*Inverse of Document Frequency*): ce facteur mesure l'importance d'un terme dans toute la collection (pondération globale). Il traduit l'impact d'un terme selon son nombre d'apparitions dans la base documentaire. La formule qui exprime l'importance d'un terme dans sa collection peut être vue comme suit: *log(N/DF)*, où *DF* représente le nombre de documents contenant le terme et *N* représente le nombre total de documents de la base documentaire. La combinaison des deux mesures *(TF* et *IDF)* donne une bonne approximation de l'importance du terme dans le document, particulièrement dans les corpus de documents de tailles homogènes. Les fonctions de pondération sont souvent référencées sous le nom de *TF-IDF*. Un autre facteur a été proposé pour pallier aux effets négatifs de la taille des documents sur le facteur *TF*. Robertson [Robertson et al., 1994a] et [Singhal et al., 1996] proposent d'intégrer la taille des documents à la formule de pondération, ce facteur est appelé facteur de normalisation.

e. La création de l'index

L'index, défini comme étant la structure de stockage utilisée pour mémoriser les informations sélectionnées pour la représentation du texte. Cette structure permet de sélectionner, pour n'importe quel terme, tous les documents où il apparaît. Plusieurs solutions de stockage ont été proposées parmi lesquelles : les fichiers inverses (*inverted files*), les fichiers de signatures (*signature files*) et les tableaux de suffixes (*suffix arrays*).

La solution la plus utilisée actuellement est celle des fichiers inverses. Ces fichiers sont composés de deux éléments principaux: Le dictionnaire et le fichier *posting*. Le vocabulaire consiste en la liste de tous les mots distincts extraits du texte. Pour chaque mot est assigné l'ensemble des positions dans lesquelles ce dernier apparaît.

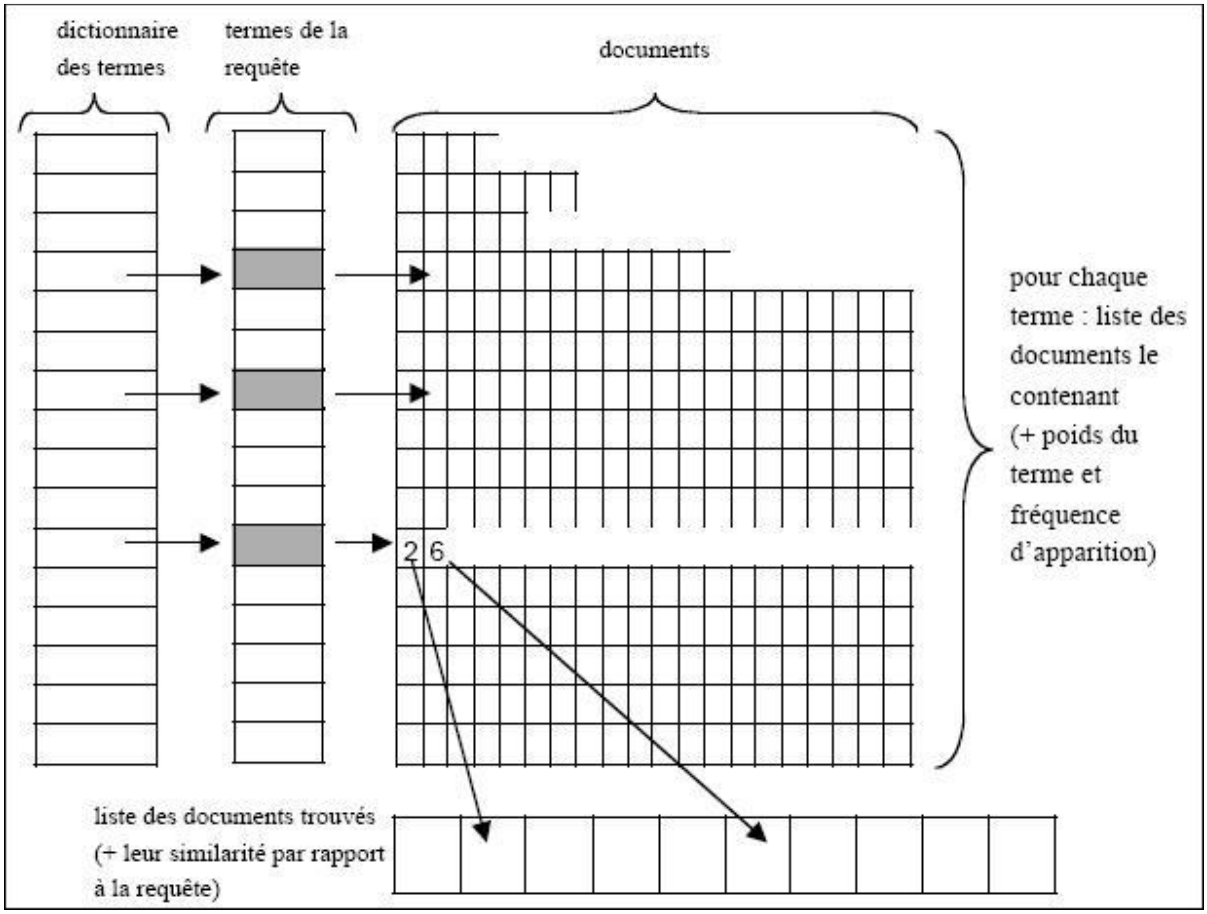

Fig. 2.3. Représentation d'un texte en fichier inverse [Bellot, 2000].

3.1.4. La fonction d'appariement requête-document

Cette fonction est définie afin de mesurer la pertinence d'un document vis-à-vis d'une requête. Elle est vue comme une probabilité ou une similarité vectorielle, notée *RSV(Q,d)* (*Retrieval Status Value*), où *Q* représente la requête et *d* représente un document. La fonction de similarité permet d'ordonner les documents renvoyés à l'utilisateur par ordre de pertinence.

3.1.5. La fonction de modification de requêtes

L'expression du besoin en information d'un utilisateur sous forme de requête est souvent une opération difficile. Par conséquent, les documents trouvés par la requête initiale peuvent ne pas accomplir le besoin en information de l'utilisateur. C'est pour cette raison que le système de recherche d'information fait appel à la fonction de modification de requêtes afin de corriger le chemin de la recherche. La reformulation de requêtes peut s'effectuer selon deux stratégies :

L'extension de la requête avec de nouveaux termes, et la repondération des termes de la requête initiale. La modification de la requête peut être manuelle, automatique ou bien semi-automatique. Cette fonction sera détaillée dans la section 3.3.

3.2. Les modèles de RI

Un modèle de RI a pour rôle de fournir une formalisation du processus de recherche d'information. Il doit accomplir plusieurs rôles dont le plus important est de fournir un cadre théorique pour la modélisation de la mesure de pertinence [Salton *et al.,* 1983]. On distingue trois principaux modèles :

- Les modèles ensemblistes
  - Modèle booléen
  - Modèle booléen étendu
  - Modèle des ensembles flous
- Les modèles algébriques
  - Modèle vectoriel
  - Modèle vectoriel généralisé
  - Modèle connexionniste
- Les modèles probabilistes
  - Modèle binaire
  - Modèle Bayesien
  - Modèle de langage

Les modèles ensemblistes reposent sur la théorie des ensembles. Dans ces modèles, les termes de la requête sont séparés par des opérateurs logiques: conjonction (ET), disjonction (OU) et négation (NON). Ces opérateurs permettent d'effectuer des opérations d'union «OU», d'intersection «ET» et de différence «NON» entre les ensembles de résultats associés à chaque terme.

Les modèles algébriques se basent sur la théorie algébrique. Dans ces modèles, la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance (ou similarité) dans un espace vectoriel.

Enfin, les modèles probabilistes se basent sur la théorie des probabilités. Pour ces modèles, la pertinence d'un document vis-à-vis d'une requête est vue comme une probabilité de pertinence document/requête.

Dans les sections suivantes, nous décrivons pour chacun de ces courants le modèle le plus représentatif (à savoir : le modèle booléen, le modèle vectoriel et le modèle probabiliste).

3.2.1. Le modèle booléen

Le modèle booléen est le premier modèle qui s'est imposé dans le monde de la recherche d'information. Il se base sur la manipulation des ensembles et l'algèbre de Boole. Dans ce modèle une requête est une expression logique composée de termes séparés par des opérateurs logiques (ET, OU et NON). Les poids des termes dans l'index sont binaires, c'est-à-dire que les termes sont présents ou absents du document ($wij \in \{0,1\}$). Le modèle booléen utilise l'appariement exact, c'est-à-dire qu'il ne permet de restituer que les documents appartenant à l'ensemble décrit par la requête. La similarité entre un document et une requête est définie par:

$$\begin{cases} RSV(q,d) = 1 \text{ si } d \text{ appartient à l'ensemble décrit par la requête} \\ 0 \end{cases} \quad (2.1)$$

Ainsi, un document est considéré dans le modèle booléen comme étant pertinent, ou bien non pertinent. Les résultats de la fonction de similarité ne permettent pas de renvoyer à l'utilisateur une liste ordonnée de documents. Ceci empêche le modèle d'avoir de bonnes performances.

3.2.2. Le modèle vectoriel

Le modèle vectoriel est l'un des modèles qui utilisent l'approche statistique. Il consiste à représenter les documents et les requêtes sous forme de vecteurs de termes pondérés. L'idée de base du modèle vectoriel est d'utiliser une représentation géométrique pour classer les documents par ordre de pertinence par rapport à une requête, c'est-à-dire que les documents et les requêtes sont représentés sous forme de vecteurs dans l'espace vectoriel engendré par les termes extraits de tous les documents de la collection. Cette idée a été développée par [Salton, 1970] dans leur projet *SMART* (*Salton's Magical Automatic Retriever of Text*). L'idée de base repose sur la mesure de similarité représentée par le produit scalaire.

Contrairement au modèle booléen où les termes de la requête doivent être reliés par des connecteurs logiques, le modèle vectoriel permet à l'utilisateur d'exprimer son besoin en information sous forme d'une liste de mots clés ou en langage naturel.

Formellement, dans le modèle vectoriel, la représentation d'un document est vue comme un vecteur $\vec{d}_j = \{w_{1,j}, w_{2,j}, \ldots, w_{n,j}\}$ où $w_{i,j}$ représente le poids des termes dans le document, n étant le nombre total de termes de l'index. La requête est aussi est vue comme un vecteur $\vec{q} = \{w_{1,q}, w_{2,q}, \ldots, w_{n,q}\}$. Une des plus simples mesures de similarité est celle du produit scalaire:

$$RSV(\vec{d}, \vec{q}) = \sum_{i=1}^{n} w_{i,j} * w_{i,q} \quad (2.2)$$

Cette mesure de similarité consiste à mesurer le nombre de termes partagés entre la requête et les documents, car les poids des termes sont binaires.

Plusieurs fonctions de similarité ont été proposées dans la littérature. Nous citons les fonctions les plus répandues: les mesures de Cosinus, Jaccard et Dice.

Mesure du Cosinus: $$Sim(D_j, Q_k) = \frac{\sum_{i=1}^{N}(wd_{ij}, wq_{ik})}{\sqrt{\sum_{i=1}^{N} wd_{ij}^2 * \sum_{i=1}^{N} wq_{ik}^2}} \quad (2.3)$$

Mesure du Jaccard: $$Sim(D_j, Q_k) = \frac{\sum_{i=1}^{N}(wd_{ij}, wq_{ik})}{\sum_{i=1}^{N} wd_{ij}^2 + \sum_{i=1}^{N} wq_{iq}^2 - \sum_{i=1}^{N}(wd_{ij}, wd_{ik})} \quad (2.4)$$

Mesure du Dice: $$Sim(D_j, Q_k) = 2 * \frac{\sum_{i=1}^{N}(wd_{ij}, wq_{ik})}{\sum_{i=1}^{N}(wd_{ij}^2, wd_{ik}^2)} \quad (2.5)$$

Les avantages du modèle vectoriel sont nombreux. Ce model permet la pondération des termes ; ce qui augmente les performances du système. Il permet de renvoyer des documents qui répondent approximativement à la requête et effectivement de trier les documents répondant à une requête. Les documents sont en effet restitués dans un ordre décroissant de leur degré de similarité avec la requête. Plus le degré de similarité d'un document est élevé, plus le document ressemble à la requête et plus il est susceptible d'être pertinent pour l'utilisateur.

Théoriquement, le modèle vectoriel présente le principal inconvénient lié à l'indépendance mutuelle des termes d'indexation. [Wong et al., 1985] ont proposé un modèle vectoriel généralisé (*Generalized Vector Space Model*) qui lève l'hypothèse d'indépendance des termes. Aujourd'hui le modèle vectoriel est le plus populaire en recherche d'information. Malgré sa simplicité, il donne de bons résultats par rapport aux autres méthodes d'ordonnancement des résultats.

3.2.3. Le modèle probabiliste

Le premier modèle probabiliste a été proposé par [Maron and Kuhns, 1960] et propose de modéliser le processus de sélection des documents dans un SRI en se basant sur la théorie des probabilités.

Le principe de base du modèle probabiliste consiste à présenter les résultats de recherche d'un SRI dans un ordre basé sur la probabilité de pertinence d'un document *vis-à-vis* d'une requête. [Robertson, 1977] résume ce critère d'ordre par le "principe de classement probabiliste", ou *PRP* (*Probability Ranking Principle*).

Dans le modèle probabiliste, répondre à une requête revient à spécifier les propriétés d'un certain ensemble de documents appelé: «ensemble de réponse idéal». Cet ensemble contient exactement les documents pertinents et aucun autre.

Au moment de la requête, les propriétés de l'ensemble idéal ne sont pas connues, donc il faut d'abord qu'il y ait une première tentative permettant de générer une première description probabiliste de cet ensemble. Ensuite, il faut une interaction avec l'utilisateur pour améliorer cette description probabiliste [Robertson, 1977]. Pour mesurer cette pertinence le modèle probabiliste se base sur la distribution des termes dans un échantillon représentatif de documents d'apprentissage. Les hypothèses posées sont les suivantes :

- La distribution des termes dans les documents pertinents est la même que leur distribution par rapport à la totalité des documents ;
- Les variables «document pertinent», «document non pertinent» sont indépendantes.

Le processus de recherche se traduit par le calcul du degré (ou probabilité) de pertinence d'un document *vis-à-vis* d'une requête. Deux probabilités conditionnelles sont utilisées dans le processus de décision :

- *P(wij/Pert)*: probabilité que le terme *ti* soit présent dans le document *dj* sachant que ce dernier est pertinent pour la requête.
- *P(wij/NonPert)*: probabilité que le terme *ti* soit présent dans le document *dj* sachant que ce dernier n'est pas pertinent pour la requête.

Le modèle probabiliste a été implémenté par Robertson et Walker [Robertson et al., 1994a] [Robertson et al., 1994b] dans le système Okapi.

3.2.4. Autres modèles

a. Le modèle booléen étendu

Le modèle booléen étendu, appelé aussi modèle *P_Norm*, a été introduit en 1983 par [Salton *et al.*, 1983]. Ce modèle étend le modèle booléen de base afin de supporter l'appariement approché en assignant des poids aux termes de la requête et des documents et en mesurant un score de pertinence. Le modèle booléen étendu interprète les opérateurs de l'équation de la requête comme des distances entre requêtes et documents.

Considérons un ensemble de termes $\{t_1, \ldots, t_N\}$ et soit wd_{ij} le poids du terme t_i dans le document D_j, où $D_j = (wd_{1j}, \ldots, wd_{Nj})$, avec $1 \leq i \leq N$ et $0 \leq wd_{ij} \leq 1$. La similarité entre le document D_j et une requête Q_k décrite sous une forme conjonctive ou disjonctive est donnée comme suit :

Opérateur OU: $$Sim(D_j, Q_k) = \left(\frac{\sum_{i=1}^{N} wq_{ik}^p \cdot wd_{ij}^p}{\sum_{i=1}^{N} wq_{ik}^p} \right)^{\frac{1}{p}} \quad (2.6)$$

Opérateur OU: $$\mathrm{Sim}(\mathrm{D_j}, \mathrm{Q_k}) = 1 - \left(\frac{\sum_{i=1}^{N} \mathrm{wq_{ik}^p} \cdot (1 - \mathrm{wd_{ij}^p})}{\sum_{i=1}^{N} \mathrm{wq_{ik}^p}}\right)^{\frac{1}{p}} \quad (2.7)$$

Où $p\ /0 \leq p \leq 1$ est une constante, et wq_{ik}^{p} le poids du terme t_i dans la requête Q_k. Le modèle booléen étendu est un modèle hybride qui inclut les propriétés des modèles ensembliste et algébrique, mais qui n'a pas été beaucoup utilisé par la suite, même s'il donne un cadre nouveau à la recherche d'information.

b. Le modèle vectoriel généralisé

[Wong *et al.*, 1985] ont proposé le modèle vectoriel généralisé dans le but de représenter les dépendances entre termes de l'index. Tous les modèles cités précédemment traitent les termes de l'index d'une manière indépendante. Ce modèle a essayé d'établir un cadre formel dans lequel les dépendances entre les termes peuvent être facilement représentées. Le modèle vectoriel généralisé est caractérisé par sa complexité et sa lenteur par rapport au modèle vectoriel classique.

c. Le modèle de langage

Les modèles de langage sont des modèles probabilistes. Ils désignent une fonction de probabilité qui assigne à chaque séquence de mots une probabilité. Leur objectif est de capter les régularités linguistiques d'une langue, en observant la distribution des mots, successions de mots, dans une langue donnée. Les modèles de langage sont basés sur l'hypothèse suivante: «un utilisateur en interaction avec un système de recherche fournit une requête en pensant à un ou plusieurs documents qu'il souhaite retrouver».

Dans un modèle de langage un document est considéré pertinent lorsque la requête de l'utilisateur ressemble à celle inférée par le document. Le principe est alors de chercher à estimer la probabilité qu'une requête soit inférée par le document [Boughanem *et al.*, 2004]. Cette probabilité (notée *P(Qk|D)*) sera utilisée pour ordonner la liste des résultats.

La mesure suivante permet d'ordonner les documents :

$$P(T_1, T_2, \ldots, T_n|D) = \prod_{i=1}^{n}((1 - \lambda_i)P(T_i) + \lambda_1\ P(T_i|D)) \quad (2.8)$$

Tels que $T_1, T_2, \ldots, T_n$ sont des variables aléatoires indépendantes représentant les termes de la requête, et λ_i estime l'importance du terme T_i de la requête. $P(T_i|D)$ représente la probabilité de pertinence du terme T_i dans le document D et $P(T_i)$ représente la probabilité que ce terme soit non important. Ces deux probabilités sont définies comme suit :

$$P(T_i|D) = \frac{tf(T_i|D)}{\sum_T tf(T,D)}, \text{ terme important} \quad (2.9)$$

$$P(T_i) = \frac{df(T_i)}{\sum_T df(T)}, \text{ terme sans importance} \quad (2.10)$$

Où $tf(T_i|D)$ est la fréquence du terme T_i dans le document D et $df(T)$ est le nombre de documents dans lesquels T apparaît.

3.3. La reformulation de requêtes

Le processus de la recherche d'information est incertain. Les utilisateurs peuvent avoir des idées moins que bien développées de ce qu'ils recherchent. Ils peuvent ne pas pouvoir exprimer un besoin en information en termes de requête appropriée. Les chercheurs ont conclu que bien que les utilisateurs aient souvent la difficulté d'exprimer leurs besoins informationnels avec précision, ils pourraient identifier l'information utile quand elle leur sera présentée. C'est-à-dire, qu'une fois que le système présente un premier ensemble de documents, ils peuvent facilement différencier entre les documents qui contiennent de l'information utile et ceux qui ne la contiennent pas. C'est ce qu'on appelle communément la réinjection de pertinence.

3.3.1. La réinjection de pertinence

La réinjection de pertinence (*relevance feedback*) est l'une des techniques de modification de requêtes les plus utiles dans le domaine de la recherche d'information. Cette méthode est mise en pratique quand l'utilisateur doit améliorer la requête qu'il a formulée au système de recherche d'information parce que les documents trouvés à l'étape initiale de la recherche ne répondent pas de manière pertinente aux besoins en information de l'utilisateur. La technique fonctionne comme suit:

- L'utilisateur soumet une requête au SRI, qui produit une liste ordonnée des documents selon leurs degrés correspondants de pertinence à la requête.

- L'utilisateur examine cette liste triée et détermine quels sont les documents pertinents et non pertinents.

- Avec cette information, le SRI modifie la requête initiale, en donnant plus d'importance aux termes apparaissant dans les documents pertinents, et en affaiblissant la force de ceux qui appartiennent aux non-pertinents.

- Ce processus est répété jusqu'à ce que l'utilisateur soit complètement satisfait de l'ensemble des documents trouvés.

3.4. Evaluation des systèmes de Recherche d'Information

Le domaine de la RI n'est pas une science exacte, car les approches font en sorte que la pertinence du système soit la plus proche possible de celle de l'utilisateur. Même si le temps de réponse et l'espace utilisé pour le stockage d'informations sont plus ou moins importants dans l'évaluation des SRI, la qualité des résultats renvoyés (appelée aussi efficacité) par un système reste le critère le plus important.

Cette évaluation permet de comparer les SRI entre eux. Les mesures d'évaluation doivent être effectuées pour les différents SRI sur les mêmes bases documentaires afin de rendre valable cette comparaison. Pour cela, plusieurs campagnes d'évaluation ont été créées. L'évaluation des SRI repose généralement sur trois éléments principaux:

- Une collection de documents de test;
- Des requêtes de test;
- Une liste des documents pertinents pour chaque requête.

Nous décrivons ci-dessous les mesures d'évaluation de SRI les plus courantes.

3.4.1. Rappel et précision

Les mesures de rappel et précision permettent d'évaluer la capacité d'un SRI à répondre aux deux objectifs principaux qui sont : retrouver tous les documents pertinents et rejeter tous les documents non pertinents. Afin de présenter ces deux mesures, nous introduisons le partitionnement de l'ensemble des documents restitués (noté B) par le SRI en deux sous-ensembles: un sous-ensemble de documents pertinents et un sous-ensemble de documents non pertinents (voir Figure 2.4).

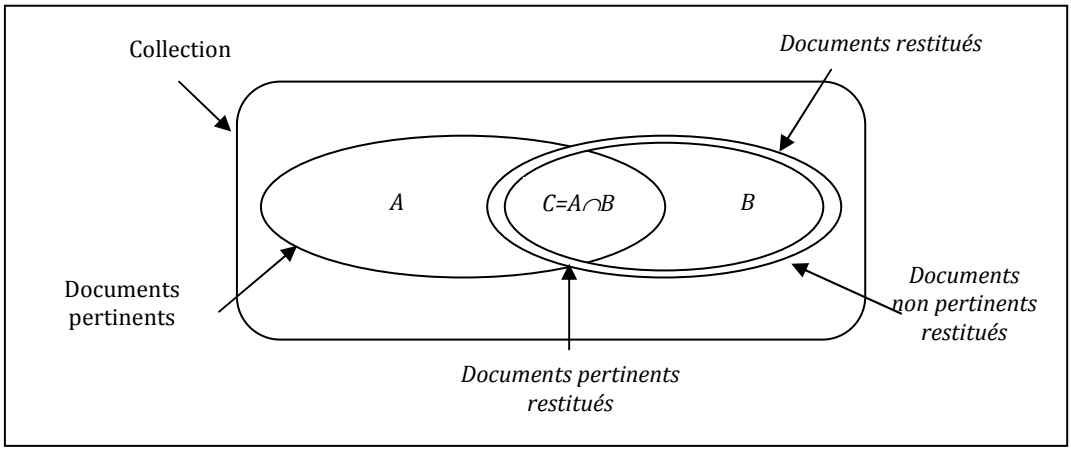

Fig. 2.4. Partition de la collection pour une requête [Mataoui, 2007]

Les taux de rappel et de précision sont définis comme suit:

Taux de rappel: le rappel mesure la capacité du système de retrouver tous les documents pertinents répondant à une requête. Autrement dit, il mesure la proportion de documents pertinents restitués par le système relativement à l'ensemble des documents pertinents contenus dans la collection. Il est exprimé par:

$$Rappel = \frac{|C|}{|A|} \qquad (2.11)$$

Taux de précision : la précision mesure la capacité du système à rejeter tous les documents non pertinents à une requête. Autrement dit, elle mesure la proportion de documents pertinents relativement à l'ensemble des documents restitués par le système. Elle est exprimée par :

$$Précision = \frac{|C|}{|B|} \qquad (2.12)$$

3.4.2. Courbes Rappel-Précision

Pour pouvoir examiner les résultats efficacement, on doit calculer les paires des mesures (taux de rappel, taux de précision) à chaque document restitué. Afin d'illustrer les calculs de rappel et de précision, nous donnons un exemple (Tableau 1) qui représente les résultats de recherche renvoyés pour une requête (Q_1) par deux systèmes (S_1, S_2) dans une collection contenant 10 documents pertinents. Les courbes de rappel-précision associées sont tracées sur la Figure 2.5.

	Système S1			Système S2		
Rang	Pertinent	rappel	précision	pertinent	rappel	précision
1	√	0.100	1.000	√	0.100	1.000
2	X	0.100	0.500	√	0.200	1.000
3	X	0.100	0.333	X	0.200	0.666
4	√	0.200	0.500	√	0.300	0.750
5	√	0.300	0.600	X	0.300	0.600
6	X	0.300	0.500	√	0.400	0.666
7	X	0.300	0.428	X	0.400	0.571
8	√	0.400	0.500	√	0.500	0.625
9	X	0.400	0.444	√	0.600	0.666
10	X	0.400	0.400	X	0.600	0.600

Tableau 2.1. Exemple de calcul de rappel et précision pour les systèmes S1 et S2

Nous pouvons remarquer que pour un même point de rappel correspondent plusieurs valeurs de précision. Une manière de rendre plus simple la lecture des courbes est de ne représenter que la précision calculée à chaque point de rappel.

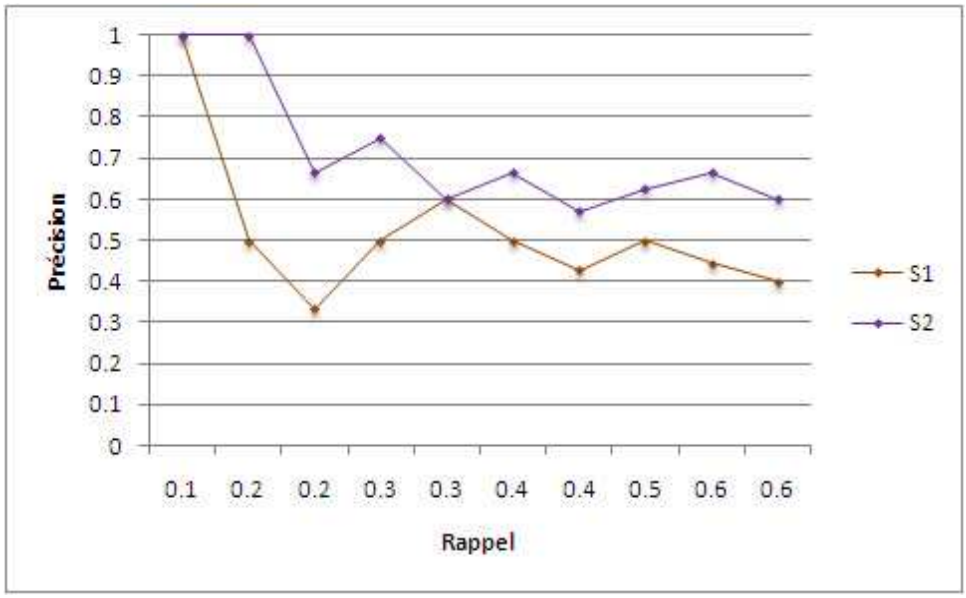

Fig. 2.5. Courbes de rappel-précision pour les systèmes S_1 et S_2. [Mataoui, 2007]

On dit qu'un système est parfait s'il ne restitue que les documents pertinents, avec un rappel et une précision de 100%. Dans la pratique, les deux mesures varient inversement, la précision diminue au fur et à mesure que le rappel augmente. Ceci signifie que la courbe rappel-précision est décroissante. Un système S_1 est dit plus performant qu'un système S_2 si sa courbe de rappel-précision est élevée par rapport à celle de S_2. Donc de la figure ci-dessus nous pouvons déduire que le système S_2 est plus performant que S_1.

5. Conclusion

La fouille de données textuelle est une discipline née en dehors de la statistique, dans la communauté de recherche documentaire et de l'intelligence artificielle dans le but de valoriser les bases de données textuelles. Elle offre des perspectives nouvelles pour la statistique et répond au défi du traitement des grands corpus de textes. Aujourd'hui la fouille de données textuelles est la branche de la statistique exploratoire qui cherche à découvrir des structures inconnues et utiles dans les textes [Gilbert, 2006]. Dans le présent chapitre nous avons exposé l'ensemble des méthodes et outils liés à la fouille de données textuelles. Nous avons aussi présenté les concepts de base de la recherche d'information à travers l'étude du processus de recherche d'information et des modèles de recherche. Chacun des modèles proposés tente de résoudre des problèmes inhérents à la recherche d'information. Nous avons aussi décrit l'évaluation des systèmes de recherche à travers l'introduction des mesures d'évaluation.

Plusieurs des notions introduites seront utilisées dans les chapitres suivants. La description générale de problématique RI permet de situer notre travail dans le cadre de l'accès à l'information basé sur l'application des nouvelles techniques de classification et de segmentation thématique des textes sur des corpus en langue arabe. L'étude porte principalement sur des corpus de traditions prophétiques «Hadith», pour des fins de découverte thématique. Dans le chapitre suivant, nous exposons les résultats expérimentaux de l'évaluation d'un système de fouille de textes et de recherche d'information dans le cas des corpus prophétiques.