REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'étude
En vue de l'obtention du diplôme de Master en informatique

Option : Conduite de Projets Informatiques

Thème :

Un modèle de langue pour une recherche d'information personnalisée

Réalisé par :
HADDAD Lakri
LAZIB Samia

Proposé et dirigé par :
F. ACHEMOUKH

Promotion : 2014/2015

REMERCIEMENTS

C’est à « ALLAH », que nous adressons toute notre gratitude en premier lieu pour nous avoir gratifiées de la lumière du savoir.

Nous adressons ensuite nos remerciements, les plus respectueux et les plus sincères à notre promotrice madame Farida Achemoukh pour avoir accepté de nous encadrer, nous soutenir et nous orienter tout au long de la réalisation de ce mémoire.

Nos remerciements vont également à l’honorable jury qui a consenti à juger notre travail.

Nous remercions aussi tous ceux qui ont

Contribué de prés ou de loin pour la réalisation de ce travail.

Liste Des Figures Et Tableaux

Liste des figures

Liste des tableaux

Sommaire

Sommaire

Chapitre 2 : la recherche d'information personnalisée

Chapitre 3 : le modèle et la recherche d'information personnalisée

Introduction générale

La recherche d'information est une activité dont la finalité est de localiser et de délivrer des granules documentaires à un utilisateur en fonction de son besoin en informations. Elle propose des outils, appelés systèmes de recherche d'information (SRI), dont l'objectif est de capitaliser un volume important d'information et d'offrir des moyens permettant de localiser les informations pertinentes relatives à un besoin en information d'un utilisateur exprimé à travers une requête.

Aujourd'hui avec L'explosion du volume d'informations disponibles sous des formats hétérogènes produites par des sources d'informations distribuées ainsi que une multiplication des utilisateurs, la recherche de l'information est devenue une tâche fastidieuse. Plusieurs outils d'accès à l'information (moteurs de recherche, systèmes de recommandation) ont été développés pour aider l'utilisateur dans cette tâche.

Tous les outils développés dans ce sens ont pour objectif de faciliter et d'accélérer l'accès à l'information pertinente. Il s'agit de s'assurer entre autre que les résultats obtenus sont compréhensibles par l'utilisateur, qu'ils correspondent aux buts et aux préférences de ce dernier. L'utilisateur, en réponse à cette avancée, est devenu plus exigeant quant aux résultats retournés par les systèmes de RI. La personnalisation tente de répondre à ces exigences en ayant pour objectif principal l'amélioration des résultats retournés à l'utilisateur en fonction de sa perception et de ses intérêts ainsi que de ses préférences.

L'objectif fondamental d'un système de recherche d'information basée sur le profil utilisateur est de retourner, à partir d'une collection de documents, les éléments qui sont pertinents à un besoin en information exprimé par l'utilisateur à travers une requête. La sélection des seuls documents intéressants un utilisateur se fait sur la base des données collectées sur l'utilisateur appelées profils et de la représentation des documents sous formes d'un index. La modélisation de ce profil est la clé de réussite de tout le processus de recherche dont la finalité est de satisfaire l'utilisateur en quête d'information en ne rapportant pour lui que les documents pertinents susceptibles de l'intéresser. Pour cela plusieurs modèles ont été proposés qui ont pour rôle de fournir une formalisation du processus de RI et un cadre théorique pour la modélisation de la mesure de pertinence. Il existe un grand nombre de modèles de RI textuelle développé dans la littérature. Parmi ces modèle les modèle probabilistes.

Notre travail s'inscrit principalement dans le contexte de la recherche d'information personnalisée et notre objectif est d'intégrer le profil utilisateur dans le système de recherche d'information et pour modéliser notre système nous avons opté pour les modèles de langue.

Pour ce faire, nous avons organisé notre mémoire en quatre chapitres, le premier aborde les concepts de base de la recherche d'information classique tout en présentons les techniques d'indexations, les différents modèles de RI, la pondération ainsi que l'évaluation des performances d'un SRI.

Le deuxième chapitre présente la personnalisation et le profil utilisateur qui est la principale base d'amélioration des performances des résultats des systèmes de recherche d'information.

Le troisième chapitre est consacré à la définition du modèle de langue, la modélisation de langue en linguistique informatique, les techniques de lissage et les modèles de langue en recherche d'information et un exemple de modèle de langue appliquées à la recherche d'information contextuelle.

Le dernier chapitre sera consacré à représenter notre contribution qui consiste en la proposition d'un modèle de langue pour la recherche d'information personnalisée suivi par un exemple illustratif.

La Recherche d'Information Classique

Introduction

La recherche d'information (RI) traite de la représentation, du stockage, de l'organisation et de l'accès à l'information. Le but d'un système de recherche d'information (SRI) est de retrouver, parmi une collection de documents préalablement stockée, les documents qui répondent aux besoins des utilisateurs, exprimés sous forme de requêtes. Pour cela, un SRI met en œuvre un ensemble de processus de sélection des documents pertinents pour la requête.

1.1 Historique de la recherche d'information

Le domaine de recherche d'information remonte au début des années 1950, peu après l'invention des ordinateurs. Comme plusieurs autres domaines informatiques, les pionniers de l'époque étaient enthousiastes à utiliser l'ordinateur pour automatiser la recherche des informations, qui dépassaient la capacité humaine : il y avait une explosion d'information après la deuxième guerre mondiale.

Le nom de « recherche d'information » (information retrieval) fut donné par Calvin N.Mooers en 1948 pour la première fois quand il travaillait sur son mémoire de maîtrise. La première conférence dédiée à ce thème – International Conférence on Scientific Information - s'est tenue en 1958 à Washington. On y comptait les pionniers du domaine, notamment, Cyril Cleverdon, Brian Campbell Vickery, Peter Luhn, etc.

Les premiers problèmes qui intéressaient les chercheurs portaient sur l'indexation des documents afin de les retrouver. Déjà à la « International Conference on Scientific Information », Luhn avait fait une démonstration de son système d'indexation KWIC qui sélectionnait les indexes selon la fréquence des mots dans les documents, et filtrait des mots vides de sens en employant des « stoplistes ». C'est à cette période que le domaine de RI est né[1].

[1] http://www.iro.umontreal.ca/~nie/IFT6255/historique-RI.html

1.2 Définitions et notions de base sur la recherche d'information:

1.2.1. Définition de la recherche d'information

D'après (l'AFNOR, 79), La recherche d'information (RI) est un ensemble de méthodes et de procédures ayant pour objet d'extraire d'une collection de documents, les informations voulues. Dans un sens plus large, la RI est toute opération ayant pour but la collecte, la recherche et l'exploitation de l'information en réponse à une question sur un sujet précis.

1.2.2 Définition d'un système de recherche d'information(SRI) :

Un système de recherche d'information est un système qui permet de retourner les **documents pertinents** à une requête formulée par l'utilisateur, à partir d'une collection de documents volumineuse ceci pour satisfaire son besoin en information.

Dans cette définition, on distingue quelques notions clés de la RI à savoir : **documents, collection de documents, requête, pertinence, besoins d'information.**

1.2.2.1. Collection de documents:

La collection de documents (ou le fond documentaire, corpus,…) constitue l'ensemble des informations exploitables et accessibles par le SRI. Elle est constituée d'un ensemble de documents.

1.2.2.2. Document:

Le document constitue l'information élémentaire d'une collection de documents. Il représente dans un SRI l'élément objectif. Dans le cadre de la RI traditionnelle, c'est du texte libre, qui peut être caractérisé selon trois vues :

- **La vue présentation** : c'est la mise en forme d'un document texte (entêtes, paragraphes, alignement…) ;
- **La vue logique** : qui présente la structure logique d'un document, elle porte des informations sur la structure ;
- **La vue de contenu** : qui se focalise sur le sens ou la sémantique d'un document représentée le plus souvent par un ensemble de mots.

Dans les SRI traditionnels, la vue de contenu est l'unique intérêt, puisque les utilisateurs forment leurs requêtes en se fixant comme objectif le contenu textuel des documents : c'est d'ailleurs la raison même de l'utilisation de tels systèmes.

1.2.2.3. Besoin en informations:

Le besoin en informations est souvent assimilé au besoin de l'utilisateur. Il est exprimé par une requête.

Trois types de besoins utilisateurs ont été définis par [**Ingwersen, 92]** :

- **Besoin vérificatif** : l'utilisateur cherche à vérifier le texte avec les données connues. Il recherche une donnée particulière et sait souvent comment y accéder (par exemple, chercher un document ayant une adresse connue sur le web).
- **Besoin thématique connu** : l'utilisateur cherche à clarifier, à revoir ou à trouver de nouvelles informations dans un sujet et un domaine connus (il est possible que le besoin s'affine au cours de la recherche).
- **Besoin thématique inconnu** : l'utilisateur cherche de nouveaux concepts ou de nouvelles relations hors des sujets ou domaines qui lui sont familiers.

1.2.2.4. Requête:

La requête constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le SRI et l'utilisateur.

Divers types de langages d'interrogation sont proposés dans la littérature pour formuler la requête. On en retrouve le langage naturel, booléen ou graphique.

1.2.2.5. Pertinence

La pertinence est une notion fondamentale et cruciale dans le domaine de la RI. Elle est l'objet de tout système de recherche d'information.

Définir cette notion complexe n'est pas simple car elle est une notion vague, à voir le nombre de définitions autour d'elle. **[Saracevic, 70]** les a regroupées dans un ensemble, nous citons quelques unes :

- La correspondance entre un document et une requête, une mesure d'informativité ;
- Un degré de relation (chevauchement) entre le document et la requête ;
- Un degré de la surprise qu'apporte un document, en rapport avec le besoin de l'utilisateur.
- Une mesure d'utilité du document retourné par le SRI à l'utilisateur.
- …etc.

Analyser ces définitions nous amène à remarquer que la pertinence est difficile à automatiser, car elle est fortement subjective, c'est-à-dire dépendante de l'utilisateur. Les travaux menés autour de la notion de la pertinence ont montré qu'elle n'est pas une relation isolée entre un document et une requête, elle fait appel aussi au contexte de jugement, ainsi qu'à l'utilisateur lui même. **[Denos, 97]** fait remarquer qu'il existe une distance plus au moins grande entre les résultats d'un système de RI et les jugements de pertinence de l'utilisateur.

On distingue la pertinence système de la pertinence utilisateur :

- **La pertinence système** : c'est l'ensemble des principes qui sous-entendent la fonction de correspondance dans un système de recherche d'information. c.-à-d. c'est l'évaluation par le SRI, de l'équivalence entre des documents et une requête.
- **La pertinence utilisateur** : quant à elle, correspond à l'ensemble des jugements de pertinence que produit l'utilisateur du système de RI.

Le but de SRI est alors de faire correspondre au mieux la pertinence système avec la pertinence utilisateur.

1.3. Architecture générale d'un SRI :

L'architecture générale d'un SRI est représentée dans la figure suivante:

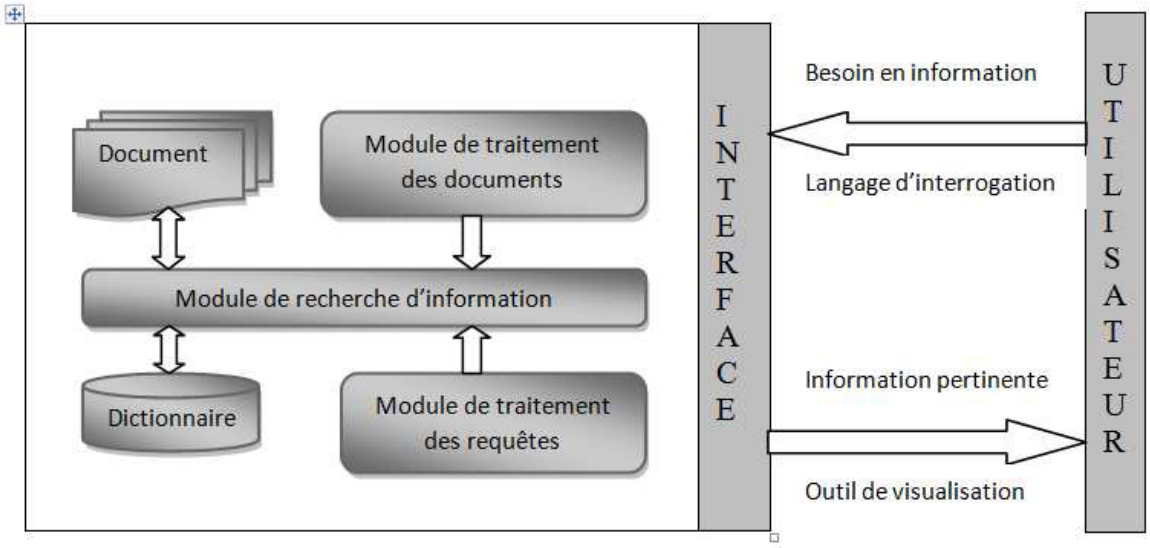

Figure 1.1: Architecture générale d'un SRI [Tamine, 98].

Dans cette figure, on distingue les éléments de base suivant :

1.3.1 L'interface :

Assure la communication entre la base documentaire et l'utilisateur. Elle doit être ergonomique et conviviale pour faciliter l'accès à l'information. La mise en œuvre de langages d'interrogation et outils de visualisation pour l'expression des requêtes d'une part et pour la visualisation de l'information pertinente d'autre part est nécessaire pour la communication entre le SRI et l'utilisateur.

1.3.1.1Langage d'interrogation :

Plusieurs langages ont été mis au point dans les SRI, les plus répondus sont les suivant :

- **Langage booléen** : L'utilisateur exprime sa requête sous forme de termes reliés par des operateurs de la logique booléenne. Comme le cas des systèmes LEXIS, STAIRS. Ce type d'interrogation est assez strict imposant une syntaxe difficilement accessible à un large public, et les requêtes sont de plus en plus complexes avec le nombre d'operateur utilisés. Les résultats de la recherche

dépendent de l'ordre des operateurs dans la requête, ainsi seuls les documents répondants à la requête sont restitués.

- **Langage naturel** : L'utilisateur exprime sa requête en langage libre, ce qui permet une utilisation généralisée des SRI, ça n'exige pas la connaissance d'une syntaxe pour formuler la requête comme est le cas avec les langages booléen. Cependant le traitement de ces requêtes ambiguës pour le système nécessite la mise en œuvre de mécanismes élaborés pour les traduire en mot clés sans perte de signification.
- **Langage graphique** : Avec ce langage, une interface d'aide à la formulation des requêtes est proposée à l'utilisateur. En effet, une vue d'ensemble de la base d'information est donné à l'utilisateur pour lui faciliter la formulation de sa requête.

1.3.1.2 Outil de visualisation :

Les outils de visualisation dans les SRI offrent la possibilité de consulter l'intégralité des documents. Les documents retournés sont présentés à l'utilisateur sous une forme qui lui permet de consulter l'information pertinente. On distingue différentes forme de présentation des résultats, dont les principales sont :

- **Présentation du document intégrale :** le système présente les documents intégraux ordonnés par ordre décroissant de ressemblance avec la requête.
- **Présentation de passage de document :** le système présente des unités d'information au lieu de documents entiers.
- **Présentation d'un identifiant de document :** le système retourne une liste d'identifiants de documents à l'utilisateur, qui peut alors visualiser le contenu de chaque document en sélectionnant son identifiant.

1.3.2 Module de traitement des documents et requêtes :

Le module de traitement des documents s'occupe de la représentation interne des documents, leur organisation et leur stockage. Le module de traitement des requêtes permet de représenter les requêtes sous un formalisme prédisposant à la recherche.

1.3.3 Module de recherche d'information :

Ce module calcule le degré de correspondance de la représentation interne du document à celle de la requête et retourne les documents jugés pertinents**.**

1.3.4 La base documentaire :

Contient un nombre important de documents. Son contenu diffère d'une base à une autre selon le domaine d'application du SRI. Principalement, on distingue deux types de bases documentaires :

1.3.4.1 Les référothéques **:** Constituées d'un ensemble d'enregistrements faisant référence aux documents dans lesquels se trouve l'information intégrale.

1.3.4.2 Les bibliothèques **:** composés de texte intégral de documents.

1.3.5 Dictionnaire :

Comprend les mots clé du domaine de la base documentaire et les mots nécessaires aux traitements des requêtes.

1.4 Processus de la recherche d'information

L'objectif fondamental d'un processus de RI est de sélectionner les documents "les plus proches" du besoin en information de l'utilisateur décrit par une requête. Pour cela, le système de recherche regroupe un ensemble de méthodes et de procédures permettant la gestion des collections de documents, stockés sous forme d'une représentation intermédiaire, permettant de refléter aussi fidèlement que possible leurs contenus sémantiques.

L'interrogation de la collection de documents à l'aide d'une requête nécessite la représentation de cette dernière sous une forme unifiée compatible avec celles des documents. Ces fonctionnalités sont représentés à travers le processus global de la RI, communément nommé **processus en U** et schématiquement représenté par la figure 1.2.

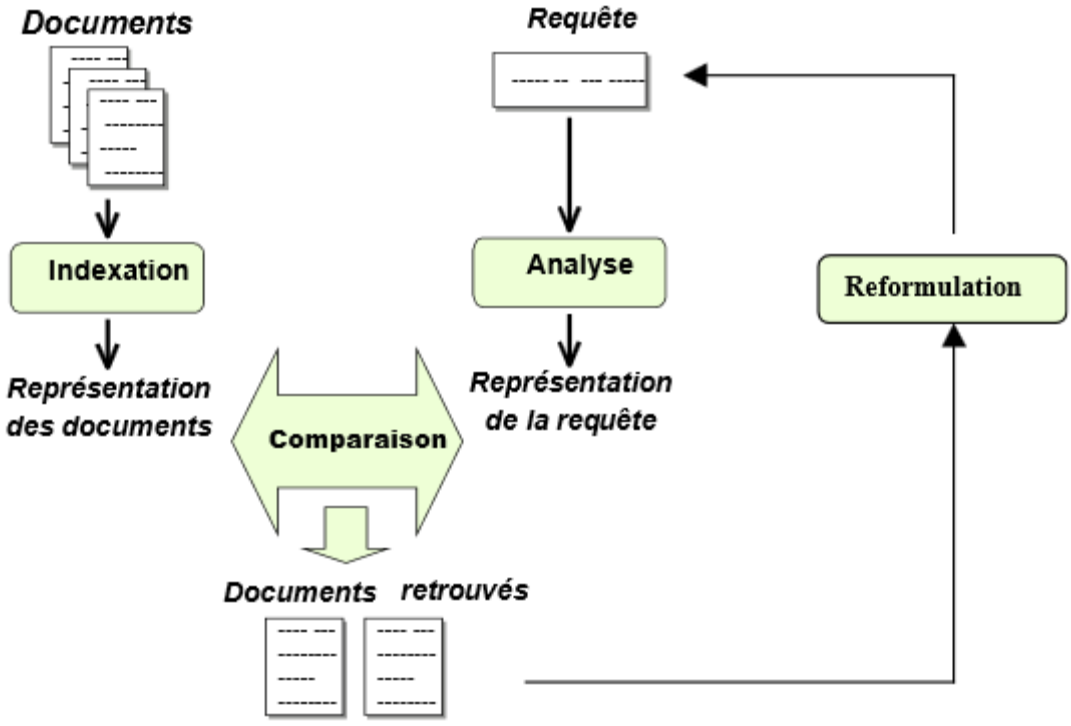

Figure 1.2 : Processus en U de la RI

Ce processus consiste en deux principales phases: l'indexation et l'interrogation.

1.4.1 L'indexation

1.4.1.1 Définition de l'indexation

Un SRI gère les différentes collections de documents en les organisant sous forme d'une représentation intermédiaire permettant de refléter aussi fidèlement que possible leur contenu sémantique. L'interrogation de ce fond documentaire à l'aide d'une requête nécessite également la représentation de cette dernière sous une forme compatible avec celle des documents. Ce processus de conversion est appelé indexation (également appelé analyse pour la requête).

L'indexation est une étape très importante dans le processus de RI. Elle consiste à déterminer et extraire les termes représentatifs du contenu d'un document ou d'une requête, qui couvrent au mieux leur contenu sémantique. La qualité de la recherche dépend en grande partie de la qualité de l'indexation.

Le résultat de l'indexation constitue, ce que l'on nomme le descripteur du document ou de requête. Ce dernier est souvent une liste de termes ou groupe de

termes significatifs pour l'unité textuelle correspondante, généralement assortis de poids représentant leur degré de représentativité du contenu sémantique de l'unité qu'ils décrivent.

Les descripteurs des documents (mots, groupe de mots) sont rangés dans une structure appelée dictionnaire constituant le langage d'indexation. Un groupe de mots est à priori sémantiquement plus riche que les mots qui le composent pris séparément. Cet argument conduit à ne pas considérer simplement les mots simples comme unités de base dans le langage d'indexation mais également des groupes de mots. Ce groupe de mots forme ce que l'on appelle un thesaurus.

1.4.1.2 types d'indexation:

L'indexation peut se faire selon trois types différents : manuel, automatique ou semi-automatique **[AMI 08, BAZ 05]:**

a) **Indexation manuelle** : Dans l'indexation manuelle, les documents sont analysés par un opérateur humain, généralement un expert du domaine. Après lecture du document, l'expert définit une liste de descripteurs qui vont refléter le contenu de ce dernier. Le jugement humai intervient indéniablement dans ce mode bien qu'il se caractérise par sa cohérence, sa profondeur et la qualité des indexes constitués. L'inconvénient majeur de ce type d'indexation est qu'il nécessite d'énormes moyens en terme d'effort intellectuel, de temps et de nombre de personnes intervenants rendu encore plus difficile par l'accroissement du volume de documents à traiter. C'est pourquoi les spécialistes de ce domaine recours à l'indexation automatique

b) **Indexation automatique** : L'indexation automatique traite les documents de façon nettement plus rapide que l'approche précédente. Ce qui lui a value d'être l'approche la plus utilisée. L'indexation automatique passe par plusieurs étapes : extraction automatique des descripteurs, utilisation d'un anti-dictionnaire pour éliminer les mots outils (ou mots vides tel que les mots grammaticaux, les signes de ponctuation..., etc), la lemmatisation (ou la radicalisation), le repérage de groupes de mots, et la pondération des mots avant de créer l'index. La première approche KWIC ou Keyword in Context fut

introduite par Luhn en 1958 à International Conference on Scientific Information (ICSI).

c) **Indexation semi-automatique (mixte) :** c'est une combinaison des deux méthodes précédentes. Les termes sont extraits de façon automatique, mais le choix final reste au spécialiste du domaine ou au documentaliste pour établir les relations entre les mots clés et choisir les termes significatifs.

Le choix du type d'indexation repose essentiellement sur le volume des masses documentaires à traiter. La qualité de l'indexation quant à elle se mesure à travers plusieurs critères notamment:

- l'exhaustivité qui revient à décrire le document de la manière la plus complète possible.
- la spécificité qui a pour but d'assurer une différenciation et une discrimination entre les descriptions des documents afin de bien pouvoir les distinguer.
- la sélectivité qui représente les degrés d'intérêt des informations retenues pour l'utilisateur.
- l'uniformité qui assure que pour un même document, différents indexeurs produiront le même indexe.

1.4.1.3 Le processus d'indexation :

L'indexation est un processus qui opère sur plusieurs niveaux, on en site les suivants:

- **Niveau du découpage**: à ce niveau le système extrait les unités lexicales communément appelées token.
- **Niveau morphologique**: consiste à reconnaitre les mots en éliminant les accents, unifiant la case et élimination des mots vides tels que les propositions et les pronoms personnels...
- **Niveau lexical**: consiste à réduire les mots à leur forme canonique. Ce procédé se fait en deux étapes: - la troncature ou racinisation qui revient à supprimer les préfixes et suffixes des mots. - la lemmatisation qui va remplacer le mot par sa forme canonique (lemme).

- **Niveau syntaxique:** consiste à extraire des groupes de mots ou des mots composés en se basant sur la grammaire de la langue.
- **Niveau sémantique**: consiste à regrouper les mots avec selon leurs sens en définissant leurs synonymes. des thesaurus et des ontologies peuvent être utilisé à ce niveau la.
- **Niveau pragmatique** consiste à analyser le langage naturel par la reconnaissance du monde réel. ce niveau n'a pas encore fait objet d'automatisation.

1.4.1.4 Pondération des termes

Cette étape est généralement basée sur des formules de pondération qui affecte à chaque terme un degré d'importance (une valeur de discrimination) dans le document où il apparait **[Hlaoua, 07].**

Il existe un grand nombre de formules de pondération. La plus parts d'entre elles se basent sur des aspects statistiques. Elles tirent leur origine des lois de Zipf et de la conjecture de Luhn.

- **Loi de Zipf:**

La loi de Zipf est une loi empirique énoncée en 1949 par G.K Zipf **[Zipf, 1949]**. Selon Zipf, les mots dans les documents ne s'organisent pas de manière aléatoire mais suivant une loi inversement proportionnelle à leur rang. Le rang d'un mot est sa position dans la liste décroissante des fréquences des mots du corpus. Ainsi, la fréquence du second mot le plus fréquent dans le corpus est la moitié de celle du premier, la fréquence du troisième mot le plus fréquent, son tiers, etc. Formellement, cette loi s'exprime par la probabilité d'apparition du nième mot le plus fréquent dans une collection de n'importe quelle langue est approximativement inversement proportionnelle à n (rang), soit : $P\ (n)\ =\ C\ N\ /\ n$

On en déduit: fréquence * rang = Constante

Dans le domaine de la recherche d'information, la loi de Zipf est utilisée pour déterminer les mots qui représentent au mieux le contenu d'un document. Pour cela, un autre concept est introduit, il s'agit de la conjecture de Luhn.

- **Conjecture de Luhn:**

La conjecture de Luhn est basée sur la loi de Zipf. Elle mesure l'informativité d'un document de la façon suivante :

• Les termes de rang faible (très fréquents) ne sont pas pertinents

• Les termes de rang élevés (très rares) ne sont pas pertinents

• Les descripteurs pertinents sont les termes de rang intermédiaire !

A : mots très fréquents, peu intéressants

C : mots peu fréquents, peu intéressants

B : mots intéressants

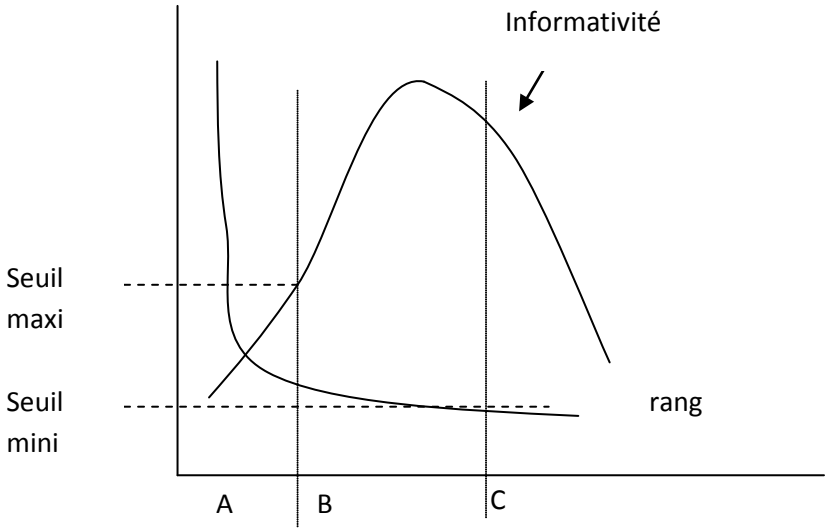

Figure 1.3 La Conjecture de Luhn

Voici un algorithme simple pour extraire et sélectionner les mots :

- Extraire les mots du corpus
- Éliminer les mots-outils (anti-dictionnaire)
- Lemmatiser (en anglais, algorithme de Porter) ;
- raciniser (déclinaisons morphologiques, représentation uniforme)
- Fixer un seuil haut et un seuil bas : on ne garde que les mots se situant entre les 2 seuils.

❖ **Pondération en tf*idf:**

Le schéma de pondérations tf*idf **[Jones, 1972]** combine un facteur de pondérations local tf quantifiant la représentativité locale d'un terme dans le document

et le second facteur de pondérations globale idf mesurant la représentativité globale du terme vis à vis de la collection du document.

- **Pondérations locale tf:**

Indique l'importance du terme dans le document. Les fonctions de pondérations locales les plus utilisées sont les suivantes:

- **Fonction brute de tf_{ij} (term frequency)** : correspond au nombre d'occurrences du terme t_i dans le document D_j.
- **Fonction binaire** : elle vaut 1 si la fréquence d'occurrence du terme dans le document est supérieure ou égale à 1, et 0 sinon.
- **Fonction logarithmique** : combine tf_{ij} avec un logarithme, elle est donnée par : $\alpha + \log(tf_{ij})$, où α est une constante. Cette fonction a pour but d'atténuer les effets de larges différences entre les fréquences d'occurrence des termes dans le document.
- **Fonction normalisée** : permet de réduire les différences entre les valeurs associées aux termes du document. Elle est donnée par la formule suivante :

$$0{,}5 + 0{,}5 + \frac{tf_{ij}}{\max\limits_{t_i \in D_j}(tf_{ij})} \qquad \textbf{(1.1)}$$

Où $\max\limits_{t_i \in D_j}(tf_{ij})$ est la plus grande valeur des termes du document D_j. Dont la plus connue est basée sur deux facteurs: fréquence de terme (TF) et fréquence inverse de document (IDF), définies dans ce qui suit **[Dahak, 06]:**

- **fréquence de terme (TF)**

La fréquence du terme (term frequency) est simplement le nombre d'occurrences de ce terme dans le document considéré. L'idée sous-jacente est que plus un terme est fréquent dans ce document, plus il est important dans la description de celui-ci. Soit le document d et le terme t, alors la fréquence TF du terme dans le document est souvent utilisée directement ou exprimée selon l'une des formules suivantes :

$$tf = \log\big(f(t,d)\big)$$

$$tf = \log(f(t,d) + 1)$$

$$tf = f(t,d)/\max(f(t,d)) \qquad \textbf{(1.2)}$$

- **fréquence inverse de document (IDF)**

La fréquence inverse du document (inverse document frequency) est une mesure de l'importance du terme dans l'ensemble du corpus. Elle consiste à calculer le logarithme de l'inverse de la proportion de documents du corpus qui contiennent le terme. Cette mesure est exprimée selon l'une des formules suivantes:

$$idf = \log\frac{|N|}{n}$$

$$idf = \log\frac{|N-n|}{n} \qquad \textbf{(1.3)}$$

Où n est la proportion des documents contenant le terme et N le nombre total de documents dans la collection. La fonction de pondération de la forme TF-IDF consiste à multiplier les deux mesures TF et IDF comme suit:

$$tf * idf = log(1 + tf) + \log\frac{|N|}{n} \qquad \textbf{(1.4)}$$

1.4.2 Interrogation :

L'interrogation est la deuxième phase du processus d'interaction d'un utilisateur avec le SRI. Une fois les documents sont représentés sous forme d'index, le système calcule la pertinence de chaque document vis-à-vis de la requête utilisateur selon une mesure de correspondance du modèle de RI et retourne les résultats à l'utilisateur.

1.4.2.1 L'appariement document-requête

Ce processus permet de mesurer la pertinence d'un document vis-à-vis d'une requête. De manière générale, à chaque réception d'une requête, le système crée une représentation de la requête qui soit similaire à celle des documents, puis calcule un score de correspondance entre la représentation de chaque document et celle de la requête. Le processus d'appariement est étroitement lié au processus d'indexation et de pondération des termes. Il existe deux méthodes d'appariement:

- **Appariement exacte** : « exact match retrieval »: Le résultat est une liste de documents respectant exactement la requête spécifiée avec des critères précis. Les documents retournés ne sont pas triés.
- **Appariement approché** : « best match retrieval » : Le résultat est une liste de documents censés être pertinents pour la requête. Les documents retournés sont triés selon leurs scores de pertinence vis-à-vis de la requête.

*1.4.2.2 La reformulation de la requête***:**

La reformulation automatique de requête permet de générer une requête plus adéquate à la recherche d'information dans l'environnement du SRI, que celle initialement formulée par l'utilisateur. Son principe est de modifier la requête de l'utilisateur par ajout de termes significatifs et/ou par ré-estimation de leur poids.

1.5 Les modèles de la recherche d'information :

Le système de recherche d'information définit une méthode d'appariement entre la représentation des documents (après le processus d'indexation) et la représentation de la requête afin de déterminer le degré de correspondance (similarité), cette méthode correspond au modèle de recherche d'information .Le modèle de recherche détermine alors le comportement clé d'un SRI. On peut distinguer trois grandes classes de modèles regroupés selon les fondements mathématiques sur lesquels ils se basent. Ils sont illustrés dans la figure 1.4.

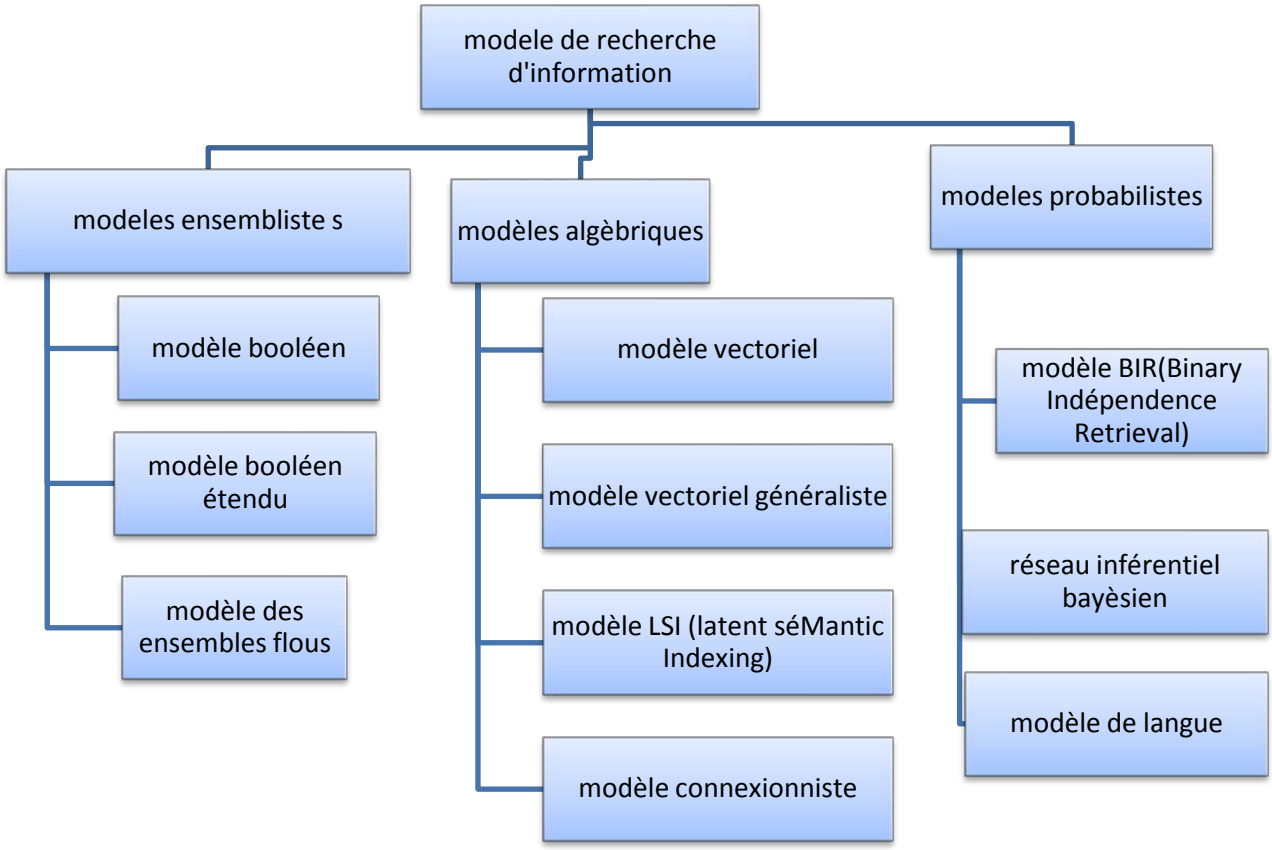

Figure 1. 4 : les trois principaux modèles de la RI [Baeza-Yates 99]

Un modèle de RI est une abstraction du processus de recherche d'information, il se distingue par le principe d'appariement **[Boughanem,06].** Un modèle de RI a pour rôle de fournir une formalisation du processus de RI et un cadre théorique pour la modélisation de la mesure de pertinence. Il existe un grand nombre de modèles de RI textuelle développées dans la littérature. Ces modèles ont en commun le vocabulaire d'indexation basé sur le formalisme mots clés et différent principalement par le modèle d'appariement entre requête-document. Nous présentons ici les modèles les plus couramment utilisés pour la RI.

A. **Les modèles ensemblistes :** Dont le représentant le plus connu est le modèle booléen. Dans ces modèles, des opérateurs logiques (OR, AND, NOT) séparent les termes de la requête et permettent d'effectuer des opérations d'union, d'intersection et de différence entre les ensembles de résultats associés à chaque terme. On distingue :

A.1 Modèle booléen : Dans ce modèle, un document est représenté comme la conjonction de l'ensemble des termes qui le composent. Une requête est quant à elle

considéré comme une expression logique dont les termes sont reliés par les opérateurs de conjonction(^), disjonction (v) ou de négation (¬).

La pertinence entre le document Dj et la requête Q (notée RSV (Dj, Q)) se calcule alors de la manière suivante :

- Si la requête contient un seul terme, soit $Qk = t$ (avec t : un terme) :
  $RSV\ (Dj, Q) = 1\ si\ t \in Dj$, 0 sinon
- si la requête contient deux termes reliés par l'opérateur (^), soit
  $Q = t1\ ^\wedge\ t2$
  $RSV\ (Dj, Q) = 1$ si $RSV\ (Dj, t1) = 1$ et $RSV\ (dj, t2) = 1$, 0 sinon
- si la requête contient deux termes reliés par l'opérateur (v), soit
  $Q = t1\ v\ t2$
  $RSV(dj, Q) = 1\ si\ RSV(Dj, t1) = 1\ ou\ RSV(Dj, t2) = 1$, 0 sinon
- si la requête est composée de la négation d'un seul terme, soit $Q = \neg\ t$:
  $RSV\ (Dj, Q) = 1\ si\ t \notin Dj$, 0 sinon

✓ **Avantages :**

- Simple à appréhender.
- Efficace si l'utilisateur maîtrise parfaitement le langage de requêtes.

✓ **Inconvénients :**

- Modélisation assez pauvre de la notion de pertinence. Cette dernière repose en effet sur un critère exclusivement binaire : un document est soit pertinent, soit non pertinent. Ce modèle ne prend pas non plus en considération la pondération des termes : un mot a un poids égal à 1 s'il appartient au document, 0 sinon.
- Les résultats retournés à l'utilisateur ne peuvent être classés : les documents retournés ont tous la même importance.
- Les documents qui ne contiennent pas tous les termes de la requête sont automatiquement considérés comme non pertinents.

A.2 le modèle booléen étendu (ou modèle P_Norm) : Le modèle booléen a été introduit en 1983 par Salton et Al **[Salton et Al, 1983].** Ce modèle étend le modèle booléen de base afin de supporter l'appariement approché, ceci 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 distances entre requêtes et documents.

Considérons un ensemble de termes $t1, \dots, tn$ et soit d_{ij} le poids du terme ti dans le document $Dj = (dij, \dots, dnj)$, avec $1 \leq i \leq N \; et \; 0 \leq dij \leq 1$. La similarité entre le document Dj et une requête Q décrite sous forme conjonctive ou disjonctive est donnée comme suit :

- Opérateur OR : $RSV\ (Dj, Q) = \left(\frac{\sum_{i=1}^{N} q_i^p d_{ij}^p}{\sum_{i=1}^{N} q_i^p}\right)^{1/p}$ **(1.5)**

- Opérateur AND : $RSV\ (Dj, Q) = 1 - \left(\frac{\sum_{i=1}^{N} (1 - d_{ij}^p}{\sum_{i=1}^{N} q_i^p}\right)^{1/p}$ **(1.6)**

Ou P une constante $0 \leq P \leq \infty$, et q_{ik} le poids du terme ti dans la requête Qk.

Remarque : lorsque p=1, les deux formules étant égales. En effet :

Opérateur AND : $RSV\ (Dj, Q) = 1 - \left(\frac{\sum_{i=1}^{N} (1 - d_{ij}^p}{\sum_{i=1}^{N} q_i^p}\right)^{1/p}$

$$1 - \left(\frac{\sum_{i=1}^{N} q_i}{\sum_{i=1}^{N} q_i^1} - \frac{\sum_{i=1}^{N} q_i * d_{ij}}{\sum_{i=1}^{N} q_i^1}\right) \quad = \frac{\sum_{i=1}^{N} q_i * d_{ij}}{\sum_{i=1}^{N} q_i^1} \qquad \textbf{(1.7)}$$

Il n'y a aucune distinction entre les deux connecteurs ET et OU. Par conséquent, la similarité entre les requêtes et les documents peut être calculée par le produit scalaire entre leurs termes pondérés.

La littérature rapporte, qu'aucune méthode formelle n'est proposée pour la détermination de la valeur du paramètre P **[Ponte, 1998].**

B. **Les modèles algébriques** : Dont le premier représentant a été le modèle vectoriel. Dans ces modèles, la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel.

B.1 modèle Vectoriel : Dans ce modèle, un document (ou une requête) est représenté par un vecteur de termes pondérés dans un espace à N dimensions, où N : représente le nombre des termes d'indexation dans la collection.

Un vecteur document Dj et un vecteur requête Q sont donc définis de la manière suivante :

$Dj = (d1j, d2j, d3j, \dots, dnj)$

$Q = (q1, q2, q3, \dots, qn)$

Avec dij : poids du terme ti dans le document Dj.

qi : poids du terme ti dans la requête Q.

Le mécanisme de mise en correspondance évalue la similarité entre les vecteurs documents et le vecteur requête. Les documents considérés comme les plus pertinents sont ceux dont le vecteur est le plus proche de celui de la requête, suivant une mesure de similarité définie au préalable.

Les principales mesures de similarité utilisées sont :

- Le produit scalaire : Sim (Q,D)=$\sum_{i=1}^{N} q_i * d_{ij}$ **(1.8)**
- Mesure de Jaccard : Sim (Q, D)= $\frac{\sum_{i=1}^{N} q_i * d_{ij}}{\sum_{i=1}^{N} q_i^2 + q_{ij}^2 - \sum_{i=1}^{N} q_i * d_{ij}}$ **(1.9)**
- La mesure cosinus : Sim (Q, D)=$\frac{\sum_{i=1}^{N} q_i * d_{ij}}{(\sum_{i=1}^{N} q_i^2)^{1/2} * (\sum_{i=1}^{N} q_{ij}^2)^{1/2}}$ **(1.10)**

✓ **Avantages**

- La pondération améliore les résultats de la recherche.
- Représentation uniforme des documents et requêtes
- Les mesures de similarité utilisées permettent d'ajouter à la notion de pertinence un degré d'approximation. Un document peut ainsi être considéré comme pertinent même s'il ne contient pas tous les termes de la requête.
- Le classement ordonné des résultats par ordre décroissant de pertinence.

✓ **Inconvénients**

- Ne prend pas en compte l'ordre des mots. Ce modèle offre la même représentation interne.
- Ce modèle suppose l'indépendance entre les termes d'indexation. Or ceci n'est pas toujours le cas : en effet, d'une part un concept est souvent représenté par un groupe de mots, d'autre part par les mots d'une langue entretiennent les uns avec les autres des relations de natures diverses (synonymie et de polysémie, etc.)

B.2 modèle connexionniste (basé sur les réseaux de neurones) : L'idée de base est que la RI est un processus associatif (elle va d'une simple comparaison des requêtes et des documents à des techniques associatives basées sur des associations de documents pour l'expansion de la réponse (sélection de nouveaux documents)) qui peut être représenté par les mécanismes de propagation d'activation des réseaux de neurones.

C. **Les modèles probabilistes** : Reposant 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.

Il s'agit plus précisément ici, de répondre à la question suivante :

Etant donné un document d et une requête q, quelle est la probabilité que d soit pertinent pour q ?

C-1 modèle probabiliste de base : Il a pour objectif de présenter le résultat de la recherche à l'utilisateur dans un ordre basé sur le rapport de la probabilité de pertinence d'un document pour une requête sur sa probabilité de non pertinence pour cette requête.

Soient donc :

P(R\D) : probabilité qu'un document D soit pertinent **(R).**

P(NR\D) : probabilité qu'un document D soit non pertinent **(NR).**

Le score de correspondance entre un document **(d)** et une requête **(q),** noté **RSV (D, Q)**, est donné par :

$$\textbf{RSV (D,Q)}=\frac{P(R\backslash Q)}{P(NR\backslash D)} \qquad \textbf{(1.11)}$$

Selon le théorème de Bayes, les deux probabilités : P(R\D) et P(NR\D) sont calculées de la manière suivante :

$$\textbf{P(R\D)} =\frac{\mathrm{P(R\backslash D)\ P(R)}}{P(D)} \text{ et } \mathrm{P(NR\backslash D)} =\frac{\mathrm{P(D\backslash NR)P(NR)}}{P(D)} \qquad \textbf{(1.12)}$$

Où :

P(D\R) : (resp. P(D\NR)) : représente la probabilité que le document D fasse partie de l'ensemble des documents pertinents R (resp. des documents non pertinents NR).

P(R) (resp. P(NR)) : est la probabilité qu'un document choisi au hasard soit pertinent (resp.non pertinent).

P(D) : correspond à la probabilité qu'un document soit choisi.

Après simplification, le calcul du score de correspondance entre un document et une requête peut être noté :

RSV(D, Q)$\approx \frac{P(D\backslash R)}{P(D\backslash NR)}$

La probabilité qu'un document soit pertinent s'appuie sur la probabilité de pertinence de ses termes.

Pour chaque terme t_i de la requête, on calcule la probabilité qu'un document qui contienne t_i soit pertinent (P ($t_{i=1}$\R)) ou non pertinent (P($t_{i=1}$\NR)).

- ✓ **Avantages**
  - Les documents jugés pertinents (sélectionnés) seront restitués dans l'ordre de leur pertinence.
- ✓ **Inconvénients**
  - Les calculs des probabilités sont complexes.
  - Pas de prise en compte des dépendances entre les termes.

C.2 Réseau bayésien : Les réseaux bayésiens on été utilisé en RI depuis les années 90. On distingue deux principaux types de réseaux, les réseaux d'inférence introduit par Turtle **[Turtle & Croft, 1990]** et les réseaux de croyance développé par **[Ribeiro-Neto et al., 1996].**

C-3 Modèle de langue : Le principe des approches utilisant un modèle de langue est différent des approches classiques en RI. En effet, plutôt que d'évaluer le degré de similarité des documents et requêtes, le modèle de langue considère que la pertinence d'un document pour une requête est en rapport avec la probabilité que la requête puisse être générée par le document **[PON & CRO, 1998].**

Formellement soit Md, le modèle de langue du document d ; la pertinence de D vis-à-vis d'une requête Q revient à estimer P(Q\Md), c'est-à-dire, la probabilité que la requête Q soit générée par Md. Etant donné une requête Q, cette pertinence est mesurée par :

RSV (D, Q) = $P(Q\backslash M_d) = \prod_{i=1}^{n} P(t_i\backslash D)$ **(1.13)**

Avec n : est le nombre de termes dans la requête.

t_i : un terme de la requête et 1<i<n.

Les documents retournés à l'utilisateur sont alors classés par ordre décroissant de la probabilité $P(Q\backslash M_d)$

1.6 Evaluation d'un SRI:

L'évaluation d'un système de recherche d'information peut être appréhendée selon deux aspects: un aspect efficience et un aspect efficacité. L'aspect efficience dépend de l'évaluation cognitive de l'utilisateur, tels que la facilité d'utilisation du système, rapidité d'accès, temps de réponse à une requête, présentation des résultats, etc. L'aspect efficacité concerne la capacité du système à sélectionner le maximum de documents pertinents et un minimum de documents non pertinents.

Pour évaluer et comparer des SRI, on utilise des mesures d'évaluation. Les mesures les plus utilisées par les compagnes d'évaluation sont : le rappel et la précision, le bruit et le silence. Pour ce faire, il est nécessaire de disposer d'une collection de test (corpus de test).

1.6.1 Corpus de test :

Dans un corpus de test, on retrouve :

- un ensemble de documents ;
- un ensemble de requêtes ;
- la liste de documents pertinents pour chaque requête.

Pour qu'un corpus de test soit significatif, il faut qu'il possède un nombre de documents assez élevé. Les premiers corpus de test développés dans les années 1970 renferment quelques milliers de documents. Les corpus de test plus récents à l'image de ceux de TREC, contiennent en général plus de 100 000 documents.

L'évaluation d'un système ne doit pas reposer seulement sur une requête. Pour avoir une évaluation assez objective, un ensemble de quelques dizaines de requêtes, traitant des sujets variés, est nécessaire.

L'évaluation du système doit tenir compte des réponses du système pour toutes ces requêtes.

Finalement pour établir les listes de documents pour toutes les requêtes, les utilisateurs (ou des testeurs simulant des utilisateurs) doivent examiner chaque document de la base de document, et juger s'il est pertinent. Après cet exercice, on connaît exactement quels documents sont pertinent pour chaque requête. Pour la construction d'un corpus de test, les jugements de pertinence constituent la tâche la plus difficile.

1.6.2 Précision et rappel : La comparaison de réponses d'un système pour une requête avec les réponses idéales nous permet d'évaluer les métriques suivantes :

- **Le taux de rappel (R)** : est une mesure du pourcentage des documents pertinents ayant été retrouvé parmi tous les documents pertinents dans la base (collection).

$$R = \frac{nombre\ \ de\ documents\ \ pertinents\ \ retrouv\ és}{nombre\ \ de\ documents\ \ pertinents\ \ dans\ la\ collection} \qquad \textbf{(1.14)}$$

- **Le taux de précision (P)** : calcul le pourcentage de documents pertinents retrouvés parmi tous les documents retrouvés par le système. Elle est calculée comme suit :

$$P = \frac{\text{nombre de documents pertinents retrouv és}}{\text{nombre total de documents retrouv és}} \qquad \textbf{(1.15)}$$

- **Le bruit** : il mesure le taux de documents non pertinents extrait par rapport à la totalité des documents extrait.

$$bruit = \frac{\text{nombre de documents non pertinents retrouv és}}{\text{nombre de documents retrouv és}} \qquad \textbf{(1.16)}$$

- **Le silence :** il mesure le taux de documents pertinents non extraits par rapport à la totalité des documents pertinents contenus dans le corpus.

$$silence = \frac{\text{nombre de documents pertinents non retrouvés}}{\text{nombre de documents pertinents}} \qquad \textbf{(1.17)}$$

Silence = 1- rappel

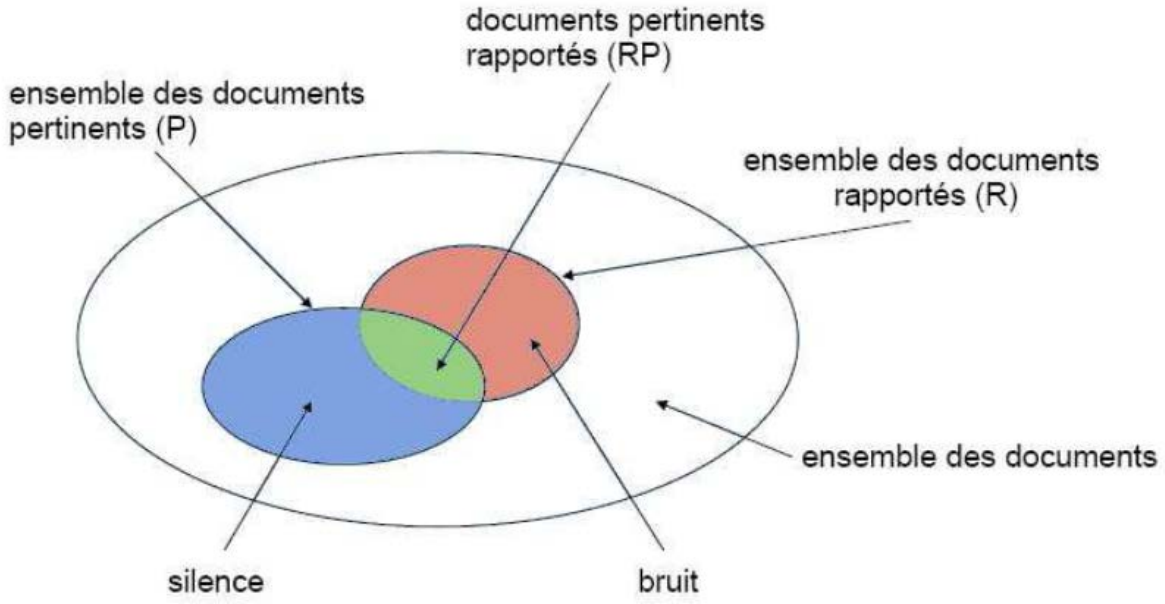

Figure 1.5 Exemple de rappel et de précision pour une requête

La figure illustre la précision et le rappel d'une requête d'une façon générale. Toutefois, seule une partie des documents restituée par le système est examinée par l'utilisateur. Dans ce cas, la paire des mesures (taux de rappel, taux de précision) est calculée a chaque point de rappel (document pertinent restitué). Il s'agit de considérer la liste ordonnée des documents évalués, de calculer pour chaque document sélectionné la précision et le rappel, puis exprimer en fonction des valeurs trouvées la précision en fonction du rappel. Idéalement, on voudrait qu'un système donne de bons taux de précision et de rappel en même temps.

Les mesures de précision-rappel ne sont pas statiques non plus (c'est-à-dire qu'un système n'a pas qu'une mesure de précision rappel). Le comportement d'un système peut varier en faveur de précision ou en faveur de rappel (en détriment de l'autre métrique). Ainsi, pour un système, on a une courbe de précision-rappel qui a en général la forme suivante :

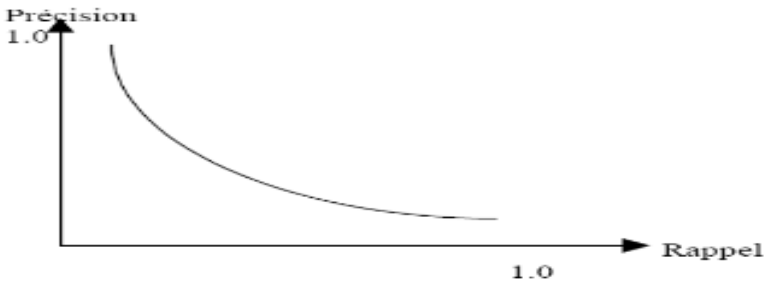

Figure 1.6 : courbe de précision-rappel

1.6.3 Comparaison de système et précision moyenne

Si on veut comparer deux systèmes de RI, il faut tester avec le même corpus de test (ou plusieurs corpus de test).

Un système dont la courbe dépasse celle d'un autre est considéré comme un meilleur système. Il arrive parfois que les deux courbes se croisent. Dans ce cas, il est difficile de dire quel système est meilleur. Pour résoudre ce problème, on utilise la précision moyenne comme une mesure de performance.

- **La précision moyenne** : est une moyenne de précision sur un ensemble de points de rappel. On utilise soit la précision moyenne sur 10 points de rappel (0.1,….,1.0), ou celle sur 11 points de rappel (0.0, 0.1, …,1.0). Cette dernière (11 point de rappel) est possible seulement avec la polarisation. La précision moyenne décrit bien la performance d'un système. C'est la mesure souvent utilisée en RI.

  Pour comparer deux systèmes ou deux méthodes, on utilise souvent l'amélioration relative qui est calculée comme suit:

  Amélioration de méthode 2 sur méthode 1

  = (performance de méthode 2 – performance méthode 1)/performance méthode1.

Conclusion

Dans ce chapitre nous avons présenté les notions de base de la recherche d'information en remontant à son histoire laquelle repose sur l'exploitation optimale des masses d'informations documentaires qui ne cessent de croitre, ce qui a conduit les spécialistes à développer des système de recherche d'information SRI. Nous avons introduit l'architecture globale d'un SRI qui illustre l'interaction de l'utilisateur avec ce dernier. Nous avons enchainé avec la présentation du processus de conception d'un SRI qui s'articule autour de trois axes essentiels à savoir: l'indexation l'interrogation et l'appariement document requête. Nous avons présenté les différents modèles de la RI qui explique comment est modélisé l'appariement document requête. Pour finir avec l'évaluation d'un SRI où sont présentés les différents critères et approches de mesure de performance.

La Recherche d'Information Personnalisée

Introduction

L'accès à une information pertinente, adaptée aux besoins et profil de l'utilisateur est un enjeu capital dans le contexte actuel caractérisé par une prolifération massive de ressources d'informations hétérogènes. Malgré les développements récents dans le domaine de la recherche d'information (RI), les résultats produits par un moteur de recherche sont en deçà des attentes des utilisateurs exploitant un moteur de recherche lors de leurs activités quotidiennes.

La personnalisation est une dimension qui permet la mise en œuvre de systèmes centrés utilisateurs en vue d'adapter son fonctionnement à son contexte précis. C'est pourquoi les travaux s'orientent actuellement vers l'adaptation du cycle de vie d'un processus d'accès à l'information à un utilisateur spécifique en vue de lui délivrer une information pertinente relativement à ses besoins précis, son contexte et ses préférences.

La recherche d'information tend principalement à modéliser l'utilisateur selon un profil, puis à l'intégrer dans la chaîne d'accès aux contenus, afin de mieux répondre à ses besoins spécifiques. Ces travaux s'inscrivent dans le cadre de la personnalisation de l'information.

2.1 La recherche d'informations personnalisée et le profil utilisateurs

2.1.1 Définition de la recherche d'informations personnalisée RIP

La personnalisation est une dimension qui permet la mise en œuvre de systèmes centrés utilisateurs, non dans le sens d'un utilisateur générique mais d'un utilisateur spécifique et ce, en vue d'adapter son fonctionnement a son contexte précis **[Zemirli, 08]**.
Elle implique une modélisation des besoins d'un utilisateur sous forme de profil décrivant ses centres d'intérêt, ses préférences et ses déférents contextes d'utilisation pour répondre de façon adaptée a un même type de requête émis par des utilisateurs de profils différent **[Bouzeghoub].**

La personnalisation de l'information consiste à fournir à un utilisateur une information pertinente correspondant à ses préférences et à ses besoins **[Abdou et al, 06].**

Pour Kostadinov [**Kostadinov, 03]**, la personnalisation de l'information se définit par un ensemble de préférences individuelles, par des ordonnancements de critères ou par des règles sémantiques spécifiques à chaque utilisateur ou communauté d'utilisateurs. Ces modes de spécification servent à décrire le centre d'intérêt de l'utilisateur, le niveau de qualité des données qu'il désire ou des modalités de présentation de ces données.

Le Gartner Group **[Janowski et al, 01]** définit la personnalisation comme « toute interaction avec l'utilisateur dans laquelle le message, l'offre ou le contenu a été taillé sur mesure pour un utilisateur ou groupe d'utilisateur spécifiques ».

Indépendamment de l'objectif applicatif visé, on identifie trois aspects à promouvoir dans les systèmes de recherche d'information personnalisée :

- Capacité à identifier l'intention conceptuelle de l'utilisateur,
- Possibilité d'exprimer des informations liées au contexte d'utilisation courant,
- Convivialité des interactions utilisateurs-système.

2.1.2 Profil utilisateur

Dans le but de personnaliser l'information, le système de recherche d'information personnalisée SRIP doit disposer des éléments clés ayant une incidence concrète sur la recherche en cours. Les éléments en question représentent les données contenues dans le profil utilisateur et les métadonnées des documents. En effet, l'utilisateur et le document ne sont pas assimilés seulement à des descripteurs exprimés à l'aide de mots comme c'est le cas dans les SRI classiques. Ils possèdent tous deux des caractéristiques propres.

Le concept de profil utilisateur a été introduit pour l'accès à l'information en premier dans les travaux de filtrage d'information **[Beklin, 92]**, pour décrire une structure représentative de l'utilisateur, en l'occurrence ses centres d'intérêts. Cette notion a ensuite été ré-exploitée en RI personnalisée pour former les composantes du contexte directement dépendantes de l'utilisateur : centres d'intérêts, préférences, domaines professionnels, expertise, etc.

On appelle profil utilisateur toute structure qui permet de modéliser et de stocker les données caractérisant l'utilisateur. Ces données représentent les centres d'intérêts, les préférences et les besoins en informations de l'utilisateur ou un groupe d'utilisateurs **[BK, 05], [ZTB, 05].**

Il convient de distinguer la notion de profil de la notion de requête. Un profil est défini comme une mise en équation du centre d'intérêt et des préférences de l'utilisateur, alors qu'une requête est l'expression d'un besoin circonstancié que l'utilisateur souhaite voir satisfait en tenant compte de son profil. Un profil a un caractère plus invariant que les requêtes même si le centre d'intérêt et les préférences de l'utilisateur peuvent légitimement évoluer.

Pour modéliser le profil, les questions à poser sont : comment les préférences d'utilisateur peuvent être obtenues et comment seront elles structurées **[Goecks ,00]** ?

En effet, le processus de modélisation du profil utilisateur requiert deux étapes **[Amato,99].** Il faut déterminer en premier le « Quoi » puis le « Comment » :

- Le «Quoi » : c'est la détermination de ce que doit représenter le contenu du profil. Quelles sont les informations pertinentes représentant au mieux les centres d'intérêts et les besoins de l'utilisateur ?
- Le « Comment » : c'est la détermination de la structure du profil et la/les technique(s) à utiliser pour la construction de ce profil.

2.2 Modélisation de l'utilisateur

La modélisation de l'utilisateur est une discipline de recherche datant des années 70 et évoquant en premier lieu les travaux d'Allen, Cohen et Perrault. La préoccupation majeure de cette discipline est d'améliorer la qualité des interactions homme-machine par inférence et prédiction des buts, préférences et contexte des utilisateurs à partir de faits observés.

Par la suite, les méthodes issues de la modélisation de l'utilisateur ont investi et continuent à investir de nombreux domaines portant sur la mise en œuvre de systèmes intelligents tels que les systèmes ayant recours à l'analyse de langage naturel, système d'aide à l'apprentissage, système hypermédia adaptatifs et tous les systèmes personnalisées de manière générale.

Indépendamment du domaine d'application, tout système mettant en œuvre des méthodes de modélisation de l'utilisateur inclut en partie les paquets d'informations suivant:

- **Des informations personnelles** associées à l'utilisateur telles que l'âge, le pays, la langue ;

- **Les préférences** : peuvent être de différents nivaux tels que préférences de forme (style de la page, longueur d'un document) et préférences de domaine permettant de cibler le centre d'intérêt de l'utilisateur ;
- **Historique de l'utilisateur** : les interactions passées de l'utilisateur représentent une source pour prédire ses intentions et lui recommander des objets.

Les approches et techniques de la modélisation utilisateur peuvent être basé sur des modèles simples ou complexes dépendant de l'objectif final ou domaine d'application du système ; un effort de standardisation pour la généralisation de tels systèmes afin de produire des interfaces a toutefois été mené et semble donner une meilleure portée au devenir des méthodes de modélisation de l'utilisateur **[Kob 01].**

En outre, ces méthodes peuvent être interactives ou implicites, peuvent avoir une portée d'adaptation sur une session d'utilisation du système avec des informations très générales sur l'utilisateur ou sur plusieurs sessions d'utilisations du système avec un modèle plus élaboré.

2.2.1 Les approches de modélisation de l'utilisateur

La construction du profil traduit un processus qui permet d'instancier sa représentation.

L'approche de construction dépend fortement de la représentation choisie pour le profil utilisateur : Les techniques utilisées par les systèmes différent selon qu'ils représentent le profil par un (des) vecteur(s) de termes ou par des classes (hiérarchiques ou pas).

Cependant la démarche de construction commune à tous les systèmes est la suivante : on commence par collecter des informations sur l'utilisateur à partir de sources d'informations diverses, puis on applique des techniques et des algorithmes pour apprendre à partir de ces informations le profil de l'utilisateur. La construction du profil s'effectue donc en deux étapes : (1) l'acquisition et la collecte des données utilisateur ; (2) puis la construction proprement dite du profil.

2.2.1.1 Acquisition des données utilisateurs

Cette phase consiste à collecter les informations pertinentes pour instancier le profil de l'utilisateur. Le processus d'acquisition des données de l'utilisateur implique différentes formes de diagnostic ou d'évaluation. Ce processus peut collecter ces informations soit

directement à partir de la machine de l'utilisateur (côté client) ou à partir de l'application (côté serveur). Ce processus d'acquisition peut être explicite et/ou implicite.

A. l'approche explicite consiste à obtenir les informations directement de l'utilisateur,

B. l'approche implicite, largement motivée par les travaux actuels dans le domaine, implique l'exploitation des données de comportement de l'utilisateur pour inférer son profil.

- **Approche explicite :**

Cette technique constitue une approche simple pour obtenir des informations sur l'utilisateur. On interroge directement l'utilisateur ou on lui demande par exemple de remplir des formulaires pour collecter les données personnelles et démographiques tels que sa date de naissance, son statut marital, son activité professionnelle et ses centres d'intérêts. Dans le cadre de l'accès personnalisé à l'information, l'approche explicite est assimilable au feedback explicite, largement utilisé dans les systèmes de filtrage et de reformulation de requête. En effet, l'utilisateur émet directement son jugement d'intérêt en donnant une valeur de pertinence sur une échelle graduée allant du moins intéressant au plus intéressant.

L'acquisition explicite a été largement utilisée dans les systèmes de e-commerce pour personnaliser l'interface des sites web en fonction des préférences des utilisateurs. Ces systèmes peuvent être considérés comme les premières approches de personnalisation.

- **Acquisition implicite**

L'acquisition implicite ou « feedback implicite» consiste à collecter les données de l'utilisateur en observant son comportement et en scrutant son activité. L'activité peut correspondre à :

- ✓ L'utilisation de moteur de recherche : requêtes et documents sélectionnés,
- ✓ La navigation sur le web : pages web consultées, liens sélectionnés,
- ✓ L'utilisation de diverses applications dans le contexte de sa recherche: les applications du bureau (tels que les produits MS Office), les outils de messagerie électronique, les éditeurs de texte, les fichiers logs,
- ✓ Consultation de bases de données ou des bases documentaires.

Le principal avantage de cette approche est qu'elle ne nécessite aucune implication directe de l'utilisateur, ni de temps passé à émettre des jugements, ni un effort d'attention

particulier lors de sa recherche. En effet, toute interaction de l'utilisateur avec le système est considérée comme une estimation de son jugement d'intérêts.

2.2.2 Les techniques de modélisation de l'utilisateur

La mise en œuvre naïve d'un modèle classique de recherche d'information suppose que l'utilisateur est complètement représenté par sa requête et que les résultats retournés pour une même requête sont identiques même si elle est exprimée par des utilisateurs différents. Les problèmes immédiats posés par une telle hypothèse sont notamment l'ambiguïté du sens des mots, l'impossibilité de sélectionner des sources opportunes et l'inintelligibilité des résultats. En outre, ces problèmes sont d'autant plus accentués que les requêtes sont courtes (≈2.29 mots par requête) et que les sources d'information sont volumineuses et hétérogènes. Les premières solutions apportées à ce type de problèmes et pouvant s'apparenter à la personnalisation, sont les techniques de reformulation de requêtes par injection de pertinence. Cependant, vu le contexte actuel lié au volume d'informations, ces techniques sont peu fiables. Les travaux s'orientent actuellement vers une définition plus large de l'utilisateur permettant de l'exploiter dans la chaîne d'accès à l'information. Cette section présente les notions ayant émergé de cette direction de travaux puis en rapporte les principales contributions portant sur la définition du profil de l'utilisateur.

Ces techniques sont basées essentiellement sur des modèles théoriques issus de la statistique soutenus par des heuristiques et des algorithmes appropriés. Citons parmi elles:

- **Modèle linéaire :** Le modèle linéaire a une structure simple. L'hypothèse de base est que la valeur de prédiction présumée et inconnue d'un objet cible du système est une combinaison linéaire des valeurs calculées à partir d'un comportement passé de l'utilisateur. le modèle linéaire peut être aisément combiné avec des techniques collaboratives où les valeurs connues sont issues de l'appréciation du groupe associé à l'utilisateur courant. **[RIW ; 1994]**

  Le modèle linéaire peut être aisément combiné avec des techniques collaboratives ou les valeurs connues sont issues de l'appréciation des membres des groupes associés à l'utilisateur courant.
- **Modèle Markovien** : Ce modèle est essentiellement basé sur l'hypothèse markovienne qui permet de représenter une séquence d'événements ultérieurs sur la base d'un nombre fixe d'événements antérieurs. Etant donnée la distribution de

probabilités d'occurrences des événements passés, la théorie markovienne offre alors des éléments pour calculer la probabilité d'occurrence des événements futurs.

- **Réseaux de neurones :** les réseaux de neurones sont des structures basées sur l'interconnexion de nœuds et un principe d'activation par propagation de signaux à travers les connexions depuis les entrées jusqu'aux sorties. La signification effective des nœuds, des connexions et des valeurs d'activation dépend du problème pour lequel est dédié le réseau. De manière générale, les réseaux de neurones sont destinés à résoudre des problèmes de décision non linéaires. Dans le cas de la modélisation utilisateur, l'entrée représente une situation ou faits observables à partir de l'utilisateur, les sorties représentent des objets cible du système avec des valeurs d'activation qui traduisent le degré de prédiction**.**
- **Classification :** les méthodes de classification permettent de partitionner un espace d'objets en classes de manière à réduire sa dimension. Les objets d'une même classe ont des propriétés partageables dont le degré de corrélation est calculé à l'aide de métrique basées sur la similarité. De nombreuses stratégies de classification sont proposées dans la littérature. Du point de vue de la modélisation utilisateur, les méthodes de classification permettent généralement d'identifier la classe de caractéristiques de l'utilisateur courant à partir d'informations dérivées de son comportement.
- **Induction de règles :** l'induction de règles consiste en l'apprentissage de règles de prédiction à partir d'un ensemble de faits observés. Contrairement aux méthodes de classification, les techniques d'induction requièrent, durant l'apprentissage du système, l'identité de la classe associée à chaque observation. Ces techniques produisent un ensemble de règles proprement dites ou des arbres de décision.
- **Réseaux bayésiens** : Les réseaux bayésiens **[Per ,88]** ont largement investi, ces dernières années, les travaux sur la modélisation utilisateur **[Jam, 96]**. Les réseaux bayésiens sont des graphes acycliques orientés où les nœuds correspondent à des variables aléatoires. Les nœuds sont interconnectés à l'aide de liens orientés qui représentent des liens de causalité entre nœuds parents et nœuds fils. A chaque nœud est associée une distribution de probabilités conditionnelles qui permet d'assigner au nœud, une valeur de probabilité dépendante de la combinaison des valeurs de

probabilités possibles des nœuds parents. Les réseaux bayésiens sont plus flexibles que les techniques précédentes, en ce sens qu'ils permettent de représenter explicitement les relations de causalité entre faits et d'émettre des prédictions sur de nombreux paramètres du système **[Zuc, 01].**

2.3 Représentation du profil utilisateur

Comme le contenu du profil dépend fortement de l'application qui l'exploite, on trouve dans la littérature trois principales approches pour représenter et structurer les profils des utilisateurs : représentation vectorielle, hiérarchique et multidimensionnelle.

2.3.1 Représentation vectorielle

Ce type de représentation s'appuie généralement sur le modèle vectoriel **[Salton, 71].** Le contenu du profil est constitué d'un ou de plusieurs vecteurs définis dans un espace de termes. Ces termes sont obtenus à partir de plusieurs sources d'informations concernant l'utilisateur. Les coordonnées des vecteurs correspondent aux poids associés aux termes retenus dans le profil.

L'utilisation de plusieurs vecteurs correspond à deux préoccupations : pouvoir prendre en compte des centres d'intérêt multiples et gérer leur évolution dans le temps. Cette représentation apporte l'avantage de la simplicité de mise en œuvre. Néanmoins, même si ces systèmes prennent en considération des centres d'intérêts multiples en utilisant plusieurs vecteurs, cette représentation manque de structuration. Cette représentation ne facilite ni l'interprétation ni la prise en compte des différents niveaux de généralités caractérisant l'utilisateur **[Bottraud, 04].** Il reste aussi à résoudre le problème de l'ordonnancement des préférences et des centres d'intérêts de l'utilisateur. En effet, ces derniers sont très variés et n'ont pas le même degré d'importance pour chaque utilisateur. Il faut donc modéliser le profil utilisateur de façon à prendre en considération l'ensemble des paramètres représentant l'utilisateur.

2.3.2 Représentation hiérarchique

De nombreuses approches ont été suivies pour améliorer d'avantage les performances des SRIP en structurant mieux les profils utilisateurs. La construction d'une hiérarchie de concepts ou d'une ontologie personnelle, plutôt qu'un ensemble de domaines indépendants

(suivant une direction proposée dans un contexte plus général par **Huhn [Huhn, 99])** offre une alternative intéressante à l'approche précédente.

Dans cette approche, la modélisation de l'utilisateur est fondée sur l'élaboration d'une ontologie personnelle. L'ensemble des caractéristiques de l'utilisateur est organisé dans une structure hiérarchique de concepts (catégories) où chaque catégorie représente la connaissance d'un domaine d'intérêt de l'utilisateur.

Le SRIP s'appuie sur la sélection dans une ontologie générale de nœuds estimés correspondre aux intérêts de l'utilisateur.

Ainsi, le rapport de généralisation /spécification existant naturellement dans ce genre de structure permet d'avoir une représentation plus réaliste du profil utilisateur.

Le premier à avoir utilisé une telle structure fut Pretschner **[Pretschner, 99]** dans le système OBIWAN. Il a proposé un modèle innovant pour la construction du profil utilisateur. Il s'appuie sur l'ontologie publique de Magellan[2]qui est composée d'approximativement 4.400 nœuds de concepts. Semblable à ce travail, on peut citer le système SmartPush **[Kurki 99] [Gauch 03].**

Bien que la représentation de ce profil d'utilisateur soit innovatrice, ces travaux ne se servent pas des caractéristiques de la structure hiérarchique (par exemple pour dédoubler ou fusionner des nœuds dans le profil d'utilisateur) pour capturer la dynamique des changements. De plus, la sémantique générale de cette hiérarchie n'est pas formellement indiquée; dans la plupart des cas, ils correspondent à une relation de généralisation/spécialisation.

2.3.3 Représentation multidimensionnelle

La représentation multidimensionnelle du profil s'inscrit dans une réflexion globale sur la personnalisation de l'information. En effet, le profil utilisateur est un élément clé dans le processus de recherche, la modélisation de l'utilisateur doit pouvoir capturer toutes les dimensions qui représentent l'utilisateur.

Une autre proposition faite par **Amato [Amato, 99]** consiste à représenter le contenu du profil utilisateur par un modèle structuré de dimensions (ou catégories) prédéfinis. C'est la première approche où les informations sont structurées et qui offre un modèle général.

[2] http://www. magellan.excite.com

Le modèle de profil contient cinq catégories :

1. catégorie de données personnelles,
2. catégorie de données de la source,
3. catégorie de données de livraison,
4. catégorie de données de comportement,
5. catégorie de données de sécurité.

La première catégorie Données personnelles contient toutes les informations concernant l'identité de l'utilisateur.

La deuxième « Données collectées» contient les informations nécessaires pour décrire les préférences et restrictions sur les documents. Elle est divisée en trois sous catégories : contenu (des informations sur le sujet du document, la langue, etc.) structure, (format, type, date de publication, dimensions, etc.), source (provenance, auteurs, éditeurs, etc.).

Dans la catégorie « Données de livraison », on trouve les informations sur la manière de transmettre des résultats à l'utilisateur. Ces informations sont regroupées selon deux sous catégories : moyen (mode de livraison par exemple email, fax téléphone, etc.) et moment (contient des informations temporelles sur le moment de livraison comme lors d'un changement, vers midi, entre 9h et 9h15, etc.).

Dans la catégorie « Données de comportement » se trouvent des enregistrements sur les interactions de l'utilisateur avec le système (URLs des pages visitées, documents lus et pertinence, etc.).

Enfin dans, la catégorie Données de sécurité, des informations sont données sur les conditions d'accès aux données du profil.

L'auteur a proposé ce modèle dans le cadre du développement d'un service avancé de librairie digitale (recherche et de livraison personnalisé de l'information sur le Web) : le système EURO gather service. Poursuivant la classification de **[Amato; 99], Kostadinov [Kostadinov ; 03]** propose un ensemble de dimensions ouvertes, capables d'accueillir la plupart des informations caractérisant un profil.

Il distingue principalement huit dimensions :

1. les données personnelles,
2. le centre d'intérêt,

3. l'ontologie du domaine,
4. la qualité attendue des résultats délivrés,
5. la customisation,
6. la sécurité et la confidentialité,
7. le retour de préférences (feedback),
8. les informations diverses.

Ces classes de données sont brièvement décrites dans ce qui suit :

- **Les données personnelles**

Les données personnelles sont la partie statique du profil. Elles comprennent l'identité civile de l'utilisateur (nom, prénom, numéro de sécurité sociale, etc.) ainsi que des données démographiques (âge, genre, adresse, situation familiale, nombre d'enfants, etc.)

- **Le centre d'intérêt**

Le centre d'intérêt exprime le domaine d'expertise de l'utilisateur. Il peut être défini par un ensemble de mots clés ou un ensemble d'expressions logiques (requêtes).

- **L'ontologie du domaine**

L'ontologie du domaine complète la définition du centre d'intérêts en explicitant la sémantique de certains termes ou de certains opérateurs employés par l'utilisateur dans son profil ou dans ses requêtes.

- **La qualité attendue**

La qualité est un des facteurs clés de la personnalisation ; elle permet d'exprimer des préférences extrinsèques comme l'origine de l'information, sa précision, sa fraîcheur, sa durée de validité, le temps nécessaire pour la produire ou la crédibilité de sa source. Les attributs de cette dimension expriment la qualité attendue ou espérée par l'utilisateur.

- **La customisation**

La customisation concerne d'abord tout ce qui est lié aux modalités de présentation des résultats en fonction de la plateforme, de la nature et du volume des informations délivrées, des préférences esthétiques ou visuelles de l'utilisateur.

- **La sécurité**

La sécurité est une dimension fondamentale du profil. Elle peut concerner les données que l'on interroge ou modifie les informations que l'on calcule, les requêtes

utilisateurs elles-mêmes ou les autres dimensions du profil. La sécurité du processus exprime la volonté de l'utilisateur de cacher un traitement qu'il effectue.

- **Le retour de préférences**

On désigne par ces termes ce qu'on appelle communément le 'feedback' de l'utilisateur. Cette dimension regroupe l'ensemble des informations collectées sur l'utilisateur.

- **Les informations diverses**

Certaines applications demandent des informations spécifiques ne pouvant être incluses dans aucune des dimensions précédentes comme par exemple la bande passante attribuée au gestionnaire du profil. Pour cette raison l'utilisateur a la possibilité de rajouter ce type de préférences dans la partie divers du profil et de décrire leurs utilisations.

Pour une application donnée, un utilisateur n'a pas besoin de toutes les dimensions ou sous dimensions ni de toutes les informations caractérisant une dimension. Un profil donné est donc une instanciation partielle de ce méta modèle en fonction des besoins de l'utilisateur, du type d'application et de l'environnement d'exécution de cette application. Les dimensions et sous dimensions définissant un profil ne sont pas indépendantes les unes des autres; elles peuvent être liées par des associations sémantiques qui caractérisent leurs dépendances ou leurs corrélations.

2. 4 Construction du profil

La construction du profil traduit un processus qui permet d'instancier sa représentation. Ce processus peut être explicite ou implicite. La construction explicite est basée sur une collecte d'information directement fournies par l'utilisateur via l'interface du système. La construction implicite, largement motivée par les travaux actuels dans le domaine, repose sur un procédé d'inférence du contexte et préférences de l'utilisateur via son comportement lors de l'utilisation du système ou d'autres applications quotidiennes. Les informations exploitées pour la construction sont généralement issues : **[Tam ,Bough , 05]**

2. 4.1 Directement de l'utilisateur (explicite) :

- Jugement explicite sur la pertinence des termes, documents,
- Définition de différents attributs : domaine d'intérêts, niveau, langue, etc.
- Sélection des thèmes, sites favoris.

2. 4.2 Indirectement de l'utilisateur (implicite) :

- Contenu des documents crées, consultés,
- liens explorés,
- durée de lecture des documents,
- dernières pages visitées,
- type d'application.

2.5 L'évolution du profil utilisateur

L'évolution des profils désigne leur adaptation à la variation des centres d'intérêt des utilisateurs qu'ils décrivent, et par conséquent, de leurs besoins en information au cours du temps. La phase d'évolution ne prend un sens que lorsque le profil a une structure pérenne, ce qui permet de distinguer les besoins à court terme, construits à partir de la session d'interaction courante, des besoins à long terme qui sont une réelle représentation des centres d'intérêt persistants de l'utilisateur. Peu de travaux ont exploré le problème de l'évolution du profil de l'utilisateur sous l'angle de la dimension temporelle (court terme, long terme).

Dans la plupart des systèmes d'accès personnalisé, l'évolution du profil consiste à adapter son contenu aux variations des besoins utilisateur en information :

- Dans le cas de systèmes à représentation ensembliste du profil, son évolution est exprimée par l'ajout de nouveaux vecteurs de termes extraits des documents correspondants aux centres d'intérêt détecté de l'utilisateur **[Billsus et Al, 1999].**
- Dans le cas d'une représentation hiérarchique, le système fait évoluer le contenu du profil en associant de nouveaux documents collectés aux classes similaires appropriées. L'adaptation de la structure du profil aux nouvelles classes s'effectue en mettant à jour les relations entre ces classes **[Moukas, 1997]**. Elle traduit dans **[Chen et al, 2002]** la notion de cycle de vie d'un centre d'intérêt, où on associe aux classes du profil utilisateur une valeur d'énergie, traduisant le degré d'importance d'un centre d'intérêt par rapport à un autre. Basée sur les valeurs d'énergie des catégories, la structure de profil peut être modulée pendant que les centres d'intérêt des utilisateurs changent.

- Dans **[Zemirli, 2008]** l'évolution de profil consiste en une évolution de la dimension centre d'intérêt basée sur une mesure de corrélation des rangs qui évalue le degré de changement entre contextes d'usage associés à des périodes successives.

2.6 Les systèmes de recherche d'information personnalisée (SRIP)

2.6.1 Définition des SRIP : [Zemirli, 03/04]

Un système de recherche d'informations personnalisé (SRIP) est un SRI qui intègre totalement l'utilisateur tout au long du processus de recherche. Il répond ainsi de manière personnelle aux besoins en informations de chaque utilisateur.

La RI personnalisée est une activité faisant intervenir deux entités : les caractéristiques de l'utilisateur communément appelés «profil de l'utilisateur » et les caractéristiques des documents appelés les « métadonnées des documents ».

Un SRI personnalisé inclut :

- Des modèles et algorithmes pour capturer et modéliser le but, les préférences et les centres d'intérêts de l'utilisateur ou un groupe d'utilisateur. Un modèle de profil est alors décrit et instancié.
- Une procédure de mise à jour du profil qui traduit son évolution dans le temps.
- Des mécanismes pour extraire les caractéristiques descriptives des documents à l'aide de métadonnées.

2.6.2 Architecture d'un SRI Personnalisé

Le processus de personnalisation s'effectue en incluant le profil utilisateur et le profil document dans l'ensemble des étapes du processus en U de la RI **(figure 2.1).** Cette personnalisation peut avoir lieu à différents niveaux du processus :

- ✓ **Lors de l'analyse de la requête**. Le SRIP intègre le profil utilisateur pour mieux cibler le besoin informationnel effectif de l'utilisateur.
- ✓ **Lors de l'indexation du corpus documentaire**. Le SRIP utilise des métadonnées des documents pour une meilleure représentation de la sémantique des documents.
- ✓ **Lors de l'appariement requête/document**. Le SRIP inclut également le profil utilisateur pour calculer la pertinence d'un document.
- ✓ **Lors de l'affichage des résultats.** Le SRIP restitue les documents selon la directive et les préférences incluses dans le profil utilisateur.

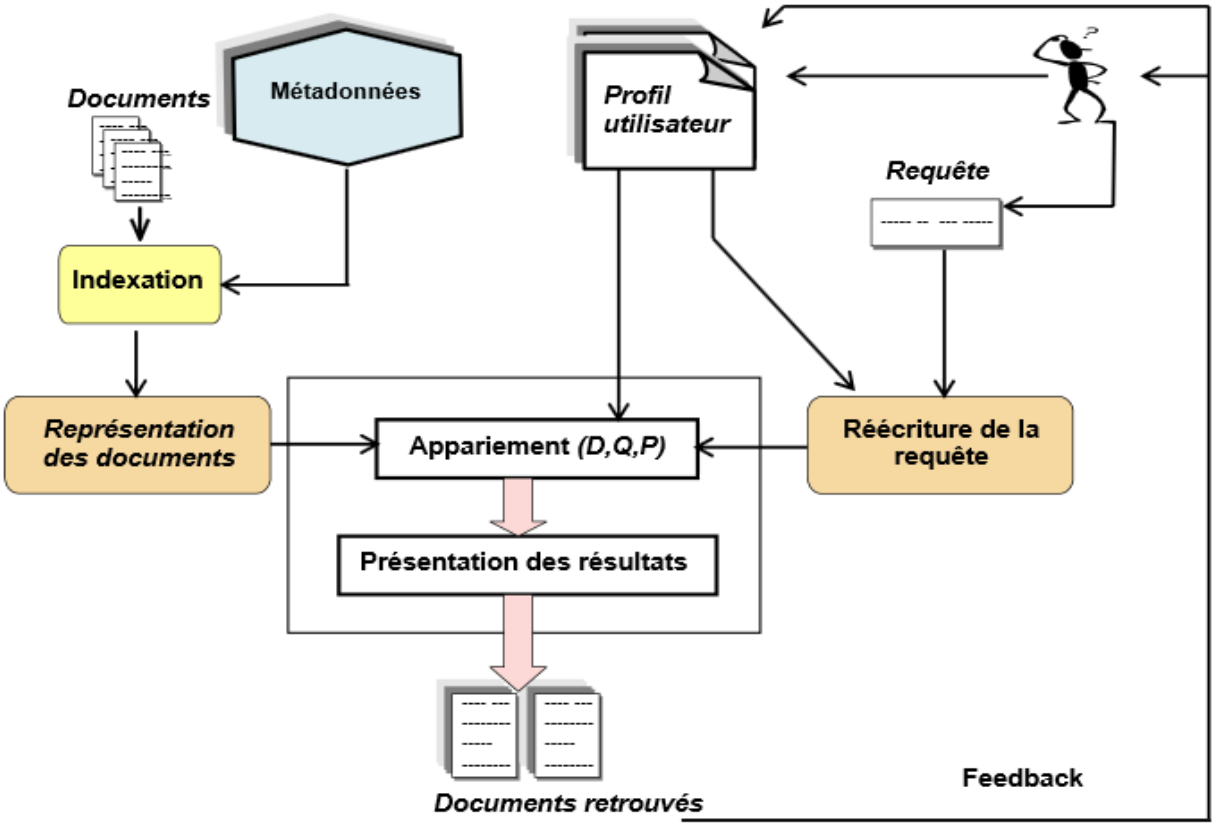

Figure 2.1 : Architecture générale d'un SRIP.

2.6.2.1 Intégration du profil utilisateur dans la phase d'évaluation de requête

D'après Zemirli, Ils ont posé que la requête initiale de l'utilisateur contient des valeurs de paramètres du profil utilisateur fournies directement par l'utilisateur. Le système peut déduire ainsi les données nécessaires pour instancier son profil. En choisissant une classe d'intérêts, l'utilisateur exprime son but de recherche et le système va extraire les documents correspondants et effectue la reformulation de la requête. La requête ainsi améliorée par les données du profil est nommée « requête profilée ».

2.6.2.2 Intégration du profil utilisateur dans la phase d'appariement document-requête

L'intégration du profil utilisateur dans cette phase consiste à modifier le calcul de la fonction d'appariement classique entre document-requête en fonction des centres d'intérêt.

Cette fonction correspond à une mesure de probabilité de pertinence d'un document sélectionné dans l'espace de recherche réduit, la requête profilée et le profil utilisateur.

2.6.2.3 Intégration du profil utilisateur dans la phase de présentation des résultats

Dans cette phase l'interaction entre le profil et le module de représentation du résultat se fait à travers des informations fournies par la catégorie de « customisation », le SRIP va déployer des mécanismes d'adaptation.

2.7 Objectifs des SRIP

L'objectif d'un SRIP est de délivrer une information pertinente en fonction des caractères spécifiques de l'utilisateur. La personnalisation de l'information peut être alors vue comme un processus de définition, construction, exploitation et évolutions des profils d'un utilisateur ou un groupes d'utilisateurs en vue de répondre de façon adaptée à un besoin en informations exprimé par un type de requête.

Ces systèmes peuvent être abordés selon deux points de vue orthogonaux **[Bottraud, 2004]** les domaines d'application et les technologies de base.

Les domaines d'application qui ont recours à la personnalisation de l'information sont nombreux : le commerce électronique, l'accès aux bibliothèques électroniques, les systèmes d'information mobiles, la configuration de logiciels...etc.

Selon les domaines, la personnalisation consistera en l'une ou plusieurs des taches suivantes :

- filtrer un flux d'informations entrant pour éliminer le bruit,
- guider la navigation dans un espace d'informations trop vaste,
- recommander un ensemble d'informations à l'utilisateur de manière plus intrusive (nouvelles offres par exemple),
- ajuster le résultat d'une requête selon une interface (ordre de présentation des résultats par exemple),
- adapter l'interaction à la situation de l'utilisateur (matérielle, géographique),

etc.

Les technologies qui permettent de supporter ces applications. On distingue entre autre Les systèmes de bases de données, les moteurs de recherche d'informations, les interfaces homme-machine.

Chacune de ces technologies a une offre différente en personnalisation :

- introduction de préférences dans les langages de requête,
- utilisation de reformulation des requêtes,

- pondération et ordonnancement des résultats des recherches, exploitation d'ontologie, etc.

2.8 L'évaluation du SRIP

Les méthodes d'évaluation largement adoptées en recherche d'information, sont empiriques (évaluation par observation expérimentale). En effet, les résultats obtenus sont issus de la comparaison de mesures et de métriques en termes de rappel et précision, sur les réponses fournies par le système relativement à celle issues des réponses attendues. Mais ces méthodes sont contestées en raison de la non prise en considération du contexte dans lequel s'effectue la recherche, et de la perception (estimation) de pertinence des utilisateurs dans ce même contexte.

Conclusion

Le profil utilisateur représente un élément fondamental dans le processus de la recherche d'informations. Il aide le système de recherche d'information à sélectionner les documents les plus pertinents et les plus fiables pour l'utilisateur. Il permet aussi de réduire le nombre de documents à renvoyer.

Dans ce chapitre nous avons présenté le profil utilisateur, comment le construire et son évolution dans le temps. Nous avons présenté aussi le système de recherche d'information personnalisé et son évaluation.

Modèle de langue et la recherche d'information personnalisée

Introduction

L'étude des modèles de recherche d'information a une longue histoire dans le domaine de la Recherche d'Information. Cette histoire continue toujours à s'illuminer de nouvelles approches.

En 1998, une nouvelle approche basée sur la modélisation statistique de langage a été proposée par Ponte et Croft **[Ponte et Croft, 98]** ; considérée comme une nouvelle alternative des variantes traditionnelle. Elle est appliquée avec succès en recherche d'information et étendue dans plusieurs publications **[Chengxiang Zhai.2001].** C'est ce que nous allons détailler dans ce chapitre.

3.1 L'approche modélisation de langage :

Une modélisation de langage est une mesure de probabilité *P(S)* pour toute séquence *S* de mots composant un corpus d'entraînement : $\sum_s P(S)$

Il est utile de comparer la modélisation de langage en recherche d'information à celle en linguistique informatique **[Ronald Rosenfeld]**. Ces deux domaines ont des frontières brouillées causées par leurs caractéristiques communes (traitement de langage), mais leur différence est dans la finalité de leur utilisation pour les modèles de langage**.**

3.2 Modélisation de langage en linguistique informatique :

L'idée fondamentale de la modélisation de langage en *linguistique informatique* est de mesurer la probabilité $P(H\ /S)$.

- S est une séquence de mots.
- H est une structure associée à *S*; détermine l'analyse syntaxique des mots.

Dans le but de déterminer l'existence d'une unité linguistique (mot, séquence de mots,…) dans un langage; une estimation de sa probabilité peut être obtenue en l'observant dans un corpus de ce langage ou bien d'estimer la probabilité de sa génération à partir d'un modèle associé à ce corpus.

Soit la séquence de mots m_n *(n varie de 1..i)* suivante : $S = m_1, m_2, ..., m_i$.

Quelque soit le modèle de langage utilisé, il tente de déterminer *la probabilité* de la séquence de mots *S*, en la décomposant en produit de probabilités conditionnelles de ses mots tout en se basant sur une approche probabiliste, donnée selon la formule :

$$P(S) = P(m1)\ x\ P(m2\ /\ m1)\ x \ldots\ x\ P(mi\ /m1, m2, \ldots, mi-1)$$

Donc: $P(S) = \prod_{i=1}^{i}(mi\ /m1,\ldots,mi-1)$ **(3.1)**

Où la probabilité conditionnelle d'un mot m_i de la séquence *S* dépend des m_{i-1} mots qui le précédent ; appelés *l'historique* h_i *(*$h_{i=}\{m_1, m_2,\ldots, m_{i-1}\}$*)* du mot m_i :

$$P(S) = \prod_{i=1}^{i} P\left(m_i/_{h_i}\right) = \prod_{i=1}^{i} \frac{P(m_1 \ldots m_{i-1} m_i)}{P(m_1 \ldots m_i)} \qquad \textbf{(3.2)}$$

Avec : *P ( )* la probabilité du nombre d'occurrences de la suite de mots dans le corpus de langage.

On estime donc la probabilité d'une unité linguistique par sa fréquence dans le corpus de langage ; ce qui est appelé l'estimation par *le maximum de vraisemblance (MLE).*

Le principal problème dans l'utilisation de ce modèle porte sur la longueur de l'historique considéré. Il est très difficile de calculer efficacement la probabilité *P* $(m_i/m_1 m_2 \ldots m_{i-1})$ en pratique car généralement, il n'existe pas de corpus de langage comprenant tous les historiques possibles pour tous les mots. On approche alors la probabilité en fonction d'un historique de taille réduite et fixe.

Il existe plusieurs techniques de modélisation statistique de langage **[Ronald Rosenfeld]:**

3.2.1 Techniques de modélisation de langage :

3.2.1.1 Modèle n-gramme :

Ce type de modèle constitue la base de la modélisation de langage. Il considère une longueur réduite pour l'historique d'un mot. Ainsi, pour la prédiction d'un mot m_i, le calcul ne tient compte que des *n*-1 mots qui le précèdent immédiatement (directement), donc l'historique h_i du mot m_i devient :

$$hi = \{m(i-(n-1)) \ldots m\ i-1\}$$

D'où:

$$P(mi\ /m1, m2, \ldots, mi-1) = P(mi\ /mi-n+1, \ldots, mi-1). \qquad \textbf{(3.3)}$$

La valeur de *n* dépasse rarement *3* en pratique **[Yannick Estève, 2002] :**

- Si n =1 alors $P(S) = \prod_{i=1}^{i}(P(m_i)$ **(3.4)**

Ce modèle est dit ***unigramme***, suppose que les mots d'une séquence soient indépendants et ne prend en compte aucun historique.

Le calcul de la probabilité du mot m_i se base sur l'estimation par le maximum de vraisemblance *(MLE)* qui revient à calculer la fréquence relative de *mot* m_i dans le corpus de langage :

$$P(m_i) = P_{MLE}(m_i) = \frac{|m_i|}{N} \qquad \textbf{(3.5)}$$

Avec :

$|m_i|$: La fréquence du mot m_i dans le corpus ;

N : la taille total du corpus.

- Si n=2 alors $P(S) = P_{MLE}(m_1) \prod_{i=2}^{i} P_{MLE}({}^{m_i}/_{m_{i-1}})$ **(3.6)**

Avec : $P_{MLE}({}^{|m_{i-1}m_i|}/_{|m_{i-1}|})$ ***(3.7)***

$|m_{i-1}m_i|$ Est le nombre d'occurrence de la suite de mots $m_{i-1}m_i$ dans le corpus.

Ce modèle est dit ***bigramme***, suppose que chaque mot de la séquence dépend de son prédécesseur direct : L'historique d'un mot m_i prend en compte son précédent m_{i-1}.

- ✓ Si n=3 alors

$P(S) = P_{MLE}(m_1) \times P_{MLE}({}^{m_2}/_{m_1}) \prod_{i=3}^{i} P_{MLE}({}^{m_i}/_{m_{i-2}m_{i-1}})$ **(3.8)**

Avec : $P_{MLE}\left({}^{m_i}/_{m_{i-2}m_{i-2}}\right) = \frac{|m_{i-2}m_{i-1}m_i|}{|m_{i-2}m_{i-1}|}$ **(3.9)**

$|m_{i-2}m_{i-1}|$ Est le nombre d'occurrence de la suite de mots m_{i-2} $m_{i-1}m_i$ dans le corpus.

Ce modèle est dit ***trigramme,*** suppose que chaque mot de la séquence dépend de ses deux prédécesseurs directs : L'historique d'un mot m_i prend en compte les deux mots qui le précédent, m_{i-2} m_{i-1}.

Le modèle n-gramme est caractérisé par plusieurs variantes, parmi elles :

- ***Modèle n-gramme distant*** propose de prédire un mot (n-gramme) en fonction d'un historique qui ne le précède pas immédiatement et cela dans le cas ou son historique immédiat lui porte peu d'informations.

Exemple

Un trigramme distant utilise comme historique d'un mot m_i, les mots m_{i-3} m_{i-2}.

➢ ***Modèle x-gramme*** est basé sur l'idée que certains termes de l'historique d'un mot, ne sont pas très utiles pour sa prédiction. Le principe est donc d'extraire dans l'historique d'un mot seul les termes jugés pertinents pour sa prédiction.

Une qualité fondamentale d'un modèle de langage *n-gramme* est la couverture totale des séquences (unités linguistiques) pouvant être exprimées dans le langage correspondant **[*Yannick Estève*, 2002].** En contrepartie sa limite est que *l'estimation par le maximum de vraisemblance* ne peut pas se formuler pour des unités linguistiques absentes dans le corpus de langage. Ce qu'est appelée, problème de fréquence **[Zéro Lafferty et Zhai].**

Exemple

Soient :

Un corpus C = {les modèles de langage pour la recherche d'information sont fondamentaux}.

Une séquence *S* = «les modèles de langage sont importants».

Pour déterminer la probabilité de cette séquence, on utilise un modèle de langage bigramme *(n-gramme d'ordre 2).*

$$P(S) = P(S/C)$$
$$= P(les) \times P(modèles/les) \times P(de/modèles) \times P(langage/de) \times P(sont/langage) \times P(importants/sont)$$

*C*omme les unités linguistiques « langage sont » et « sont importants» sont absentes dans le corpus alors leurs probabilités seront nulles :

$$P(S) = P(les) \times P(modèles/les) \times P(de/modèles) \times P(langage/de) \times 0 \times 0 = 0$$

Ceci pose un problème fondamental, car seuls les séquences apparus qui ont une probabilité non nulle. Comme solution à ce problème et afin de généraliser les modèles de langage, comme les n-grammes, il faut effectuer une redistribution de probabilités des unités linguistiques, rencontrées dans le corpus vers celles non rencontrées. Cette technique est dite, ***le lissage*** **[Victor Lavrenko, 2000],** elle sera détaillée plus loin dans ce chapitre.

***3.2.1.2 Modèle n-classes* :**

C'est une technique dite aussi *Clustering*, permet le regroupement des mots (n-grammes) dans des classes C_i et la maximisation de la quantité des données de corpus de langage afin de minimiser le problème de fréquence zéro **[*Philippe Langlais*, 2003]**.

Si un mot n'est associé qu'à une seule classe alors on parle de *Clustering dur* (hard clustering) sinon on parle de *clustering mou* (soft clustering).

Soit *Ci* la classe à laquelle le n-gramme m_i est assigné, exemple, pour un trigramme :

$$P(m_3/m_1m_2) = P(m_3/C_3)P(C_3/m_1m_2)$$
$$= P(m_3/C_3)P(C_3/m_1C_2)$$
$$= P(m_3/C_3)P(C_3/C_1C_2)$$
$$= P(m_3/C_1C_2) \cdot$$
$$= P(C_3/C_1C_2)$$

Le calcul de la probabilité d'une classe dans un modèle n-classes est alors donné comme suit : $P(c_i/c_{i-2}c_{i-1}) = \frac{|c_{i-2}c_{i-1}c_i|}{|c_{i-2}c_{i-1}|}$ **(3.9)**

La qualité du modèle résultant dépend des classes C_i et son principal avantage réside dans le fait qu'un mot d'une classe donnée ne se trouve pas forcement dans le corpus de langage, hérite des probabilités des autres mots de sa classe.

Trois types de classes sont utilisés **[*Yannick Estève* .2002]**:

- ***les classes syntaxiques*** regroupent les mots de même catégorie grammaticale afin de réduire le volume des données de corpus de langage. Les modèles basés sur ce type de classes sont utilisés dans le cas de langages sollicitant un grand nombre de données.
- ***les classes morphologiques*** regroupent les mots selon leur racine morphologique (lemme). Les modèles basés sur ces classes complètent ceux basés sur des classes syntaxiques; ils ajoutent aux connaissances syntaxiques des connaissances sémantiques fournies par les lemmes des mots.
- ***les classes obtenues par classification automatique*** regroupent les mots en se basant sur des algorithmes prédéfinis, indépendamment de toute connaissance syntaxique ou morphologique, par exemple, le regroupement se base sur le contexte de similarité entre les mots, c'est-à-dire qu'ils peuvent se substituer dans le corpus de langage.

***3.2.1.3 Modèle Adaptive*:**

Les techniques précédentes traitent le langage comme source homogène. En réalité le langage peut être hétérogène (plusieurs domaines). A cet effet, l'utilisation d'un modèle statique n'est pas efficace **[Ronald Rosenfeld]**.

Pour palier à ce problème, des adaptations inter-domaine et intra-domaine sont effectuées :

- Si les informations de test sont différentes des informations du corpus caractérisé par un modèle de langage (c-à-d, elles n'appartiennent pas au même domaine) et afin d'adapter ces informations; une technique efficace , dite de cache est utilisée : l'historique dynamique des informations de test est employé pour créer un modèle n-gramme dynamique qui sera interpolé avec le modèle statistique:

$$\boldsymbol{P_{Adaptive}(m_i/h_i) = \lambda_i P_{Statistique}(m_i/h_i) + (1-\lambda_i) P_{Cache}(m_i/h_i)} \qquad \textbf{(3.10)}$$

- Si les informations de test et les informations du corpus de langage appartiennent au même domaine (source), mais les informations du corpus sont hétérogènes ; l'adaptation est de découper le corpus en ***N*** sous-domaines différents, ce qui entraîne ***N*** modèles de langage spécialisés, interpolés avec un modèle général.

3.2.1.4 Modèle exponentiel:

Contrairement aux autres modèles, le modèle exponentiel est basé sur une approche qui peut coïncider avec la distribution de maximum d'entropie (ME), qui a une force considérable dans la combinaison de sources d'informations arbitraires, tout en évitant le problème de la fragmentation de données **[Ronald Rosenfeld].**

Ce modèle est présenté comme suit :

$$P(m/h) = \frac{1}{Z(h)} \exp[\textstyle\sum_i \lambda_i f_i(h,m)] \qquad \textbf{(3.11)}$$

Avec :

λ_i Sont des paramètres estimés pour la pondération des fonctions f_i.

$f_i(h,m)$ Appelées les traits (features), permettant de déterminer l'historique *h* pour des paires de mots.

$Z(h)$ est un paramètre de normalisation.

3.2.2 Les techniques de lissage :

Le lissage est une technique qui permet de prélever une quantité de probabilités associées aux n-grammes *(unités linguistiques)* observés dans le corpus de langage et de la redistribuer sur ceux non observés. Plusieurs techniques basées sur des méthodes de *Discounting* et de *Redistribution* ont été proposées afin que les n-grammes absents dans le corpus reçoivent des probabilités non nulles **[*Yannick Estève*]**.

➢ ***Méthode de Discounting (diminution)*** permet d'introduire pour un n-gramme m_i caractérisé par un historique h_i, une fréquence conditionnelle $f^*(m_i/h_i)$

Tel que :$0 \leq f^*(m_i/h_i) \leq f(m_i/h_i)$

Avec

$$f(m_i/h_i) = \begin{cases} \frac{|h_i m_i|}{h_i} & si\ |h_i| > 0 \\ 0 & si\ \ |h_i| = 0 \end{cases} \qquad \textbf{(3.12)}$$

est ce qu'il ne s'agit pas f*

➢ ***Méthode de Redistribution*** permet la redistribution de probabilités sur un historique h_i d'un n-gramme m_i, en faisant appel à la composante *probabilité de fréquence zéro*, calculée à partir de la fréquence conditionnelle f^*.

Pour un historique $h_{i,}$ la probabilité de fréquence zéro, notée $\lambda(h_i)$, est définie comme suit :

. $\lambda(h_i) = 1 - \sum_{m_i} f^*(m_i/h_i)$ **(3.13)**

A partir de cette formule, on peut conclure que :

Si $|h_i|$=0 *(l'historique h_i n'est pas observé dans le corpus)* alors

$f^*(m_i/h_i) = 0\ et\ \boldsymbol{\lambda(h_i) = 1}$

Plusieurs techniques de lissage ont été proposées. Elles peuvent être subdivisées en deux grandes familles **[Richard Beaufort]** :

- ***Les techniques de backoff***, où le modèle *d'ordre n-1* se substitue au modèle *d'ordre n*, s'il ne détient pas suffisamment d'informations pour un contexte donné ou le modèle *d'ordre n* se substitue au modèle *d'ordre n-1* si son maximum de vraisemblance vaut zéro.
- ***Les techniques* d'*interpolation linéaire***, où la probabilité d'un n-gramme est une combinaison linéaire des modèles d'ordre 0 à n.

Chaque technique de lissage peut être envisagée en version *backoff* ou en version *interpolation linéaire*.

***3.2.2.1 Lissage de Laplace (Additif)*:** Afin d'éviter les comptages nuls, cette technique prétend que tout n-gramme m_i est vu une fois de plus dans le corpus de langage *C*, d'où l'appellation, *technique «ajouter-un»* **[Philippe Langlais.2003]**. Sa probabilité, notée $P_{LAP}(m_i/m_{i-1})$ est calculée comme suit :

$$P_{LAP}(m_i/m_{i-1}) = \frac{|m_{i-1}m_i|+1}{\sum_m(|m_{i-1}m_i|+1)} = \frac{|m_{i-1}m_i|+1}{N+\sum_m(|m_{i-1}m_i|)} \quad \textbf{(3.13)}$$

Donc :

$$P_{LAP}(m_i/m_{i-1}) = \frac{|m_i|+1}{\sum_m(|m_i|+1)} = \frac{|m_i|+1}{N+\sum_m(|m_i|)} \quad \textbf{(3.14)}$$

Avec :

N La taille de corpus *C*.

$|m_i|$ La fréquence de m_i dans le corpus C.

Le problème de cette technique est qu'un grand espace de probabilités est distribué sur des n-grammes non vus dans le corpus. D'où la faible participation de ceux vus, dans la création et la perfection de modèle de langage de corpus associé.

Cette technique a été évoluée en un *lissage Additif* ++, basé sur l'ajout d'une valeur δ et la probabilité d'un n-gramme m_i sera notée $P_{ADD}(m_i/m_{i-1})$, calculée comme suit **[*Richard Beaufort*]** :

$$P_{ADD}(m_i/m_{i-1}) = \frac{|m_{i-1}m_i|+\delta}{\sum_m(|m_{i-1}m_i|+\delta)} = \frac{|m_{i-1}m_i|+\delta}{\delta N+\sum_m(|m_{i-1}m_i|)} \quad \textbf{(3.15)}$$

$$P_{ADD}(m_i/C) = \frac{|m_i|+\delta}{\sum_m(|m_i|+\delta)} = \frac{|m_i|+\delta}{\delta N+\sum_m(|m_i|)} \quad \textbf{(3.16)}$$

Avec : $0 < \delta \leq 1$

***3.2.2.2 Lissage de Good-Turing*:** C'est une version *backoff ;* basée sur une méthode de *discounting* et pour adresser le problème des n-grammes absents dans un corpus de langage, cette technique permet l'ajustement de la fréquence *tf* du n-gramme m_i en une fréquence dite corrigée tf^* :

$$tf^* = (tf+1)(\frac{N_{tf+1}}{N_{tf}}) \quad \textbf{(3.17)}$$

Avec :

tf le nombre d'occurrence de n-gramme dans le corpus ;

N_{tf} le nombre de n-grammes de fréquence tf présents dans le corpus.

Ainsi pour tout n-gramme m_i sa probabilité est notée $P_{GT}(m_i/m_{i-1})$, calculée comme suit :

$$P_{Gt}(m_i/C) = \frac{tf^*}{\sum_{m_i}|m_i|} = \frac{tf^*}{N} \qquad \textbf{(3.18)}$$

.

La fréquence d'ordre $\frac{tf^*}{tf}$ pour un n-gramme vu, sera redistribuée sur les n-grammes non vus dans le corpus. En particulier, dans le cas ou la fréquence tf d'un n-gramme m_i est nulle, tf^*sera réduit en : $tf^* = \frac{N_1}{N_0}$ et $P_{Gt}(m_i/C) = \frac{N_1}{N_0 \times N}$

En pratique, pour les n-grammes de très haute fréquence tf, leurs probabilités seront à zéro, ainsi que N_{tf+1} sera toujours zéro. En outre, le lissage de Good-Turing est appliqué sur des n-grammes dont la fréquence est petite **[Song et Croft].**

***3.2.2.3 Lissage de Katz*:** C'est une variante de lissage Good-Turing, proposée afin de pallier au problème de hautes fréquences des n-grammes. Le principe de cette technique consiste à utiliser un modèle de langage spécifique (d'ordre inférieur) lorsqu'un n-gramme n'est pas observé dans le corpus **[Christian Jauvin.2003]** :

$$P_{katz}(m_i/m_{i-1}) = \begin{cases} P_{GT}(m_i/m_{i-1})\ si\ |m_{i-1}m_i| > 0 \\ \alpha(m_{i-1})P_{MLE}(m_i)\ sinon \end{cases} \qquad (\mathbf{3.19})$$

Avec :

$P_{GT}(m_i/m_{i-1})$Est la probabilité d'un n-gramme m_i basée sur le lissage de Good-Turing

$P_{MLE}(m_i)$Est la probabilité d'un n-gramme m_i basée sur l'estimation par le maximum de vraisemblance *(maximum likelihood).*

$\alpha(m_{i-1})$ Est un paramètre de normalisation qui permet de déterminer la fréquence du n-gramme m_{i-1} après redistribution, calculé comme suit **[Philippe Langlais.2003]** :

$$\alpha(m_{i-1}) = \frac{1-\sum_{m_i:|m_{i-1}m_i|>0} P_{GT}(m_i/m_{i-1})}{1-\sum_{m_i:|m_{i-1}m_i|>0} P_{MLE}(m_i)} \qquad \textbf{(3.20)}$$

Avec :

$|m_{i-1}m_i|$ la fréquence d'apparition de la séquence de mots $m_{i-1}m_i$ dans le corpus.

3.2.2.4 Lissage par interpolation : Le lissage par interpolation, consiste à combiner un modèle avec un ou plusieurs modèles d'ordre inférieur (mixture de modèles) **[Christian Jauvin.2003]**. Il se présente généralement comme suit :

$$P_{INT}(m_i/m_{i-1}) = \lambda_{i-1}P_{MLE}(m_i/m_{i-1}) + (1-\lambda_{i-1})P_{INT}(m_i/m_{i-1}) \quad \textbf{(3.21)}$$

Avec :

λ_{i-1}Est un paramètre déterminé de telle manière à maximiser l'espérance des données.

Ce paramètre dépend du mot m_{i-1} et sa valeur est souvent déterminée avec le processus de maximisation de l'espérance EM.

L'interpolation des modèles est récursive et en recherche d'information le lissage par interpolation combine deux modèles, l'un estimé sur un document *D* et l'autre sur un corpus de langage *C* **[Lafferty and Zhai].**

***3.2.2.5 Lissage de Jelinek-Mercer* :** Cette méthode implémente la forme générale de l'interpolation linéaire ; elle combine des modèles d'ordre 1 à n **[Christian Jauvin .2003] :**

Exemple

Pour n=3 on a :

$$P(m_i/m_{i-2}m_{i-1} = \lambda_1 P_{MLE}(m_i/m_{i-2}m_{i-1}) + \lambda_2 P_{MLE}\left(m_i/m_{i-1}\right) + \lambda_3 P_{MLE}(m_i)$$

Avec : $\lambda_1 + \lambda_2 + \lambda_3 = 1$

L'estimation des valeurs des λ_i se fait par un processus de maximisation d'espérance (EM) **[Philippe Langlais.2003].**

***3.2.2.6 Lissage de Dirichlet* :** Le lissage de Dirichlet est basé sur l'interpolation linéaire. Il permet l'augmentation des fréquences des n-grammes m_i dans un document **[*Lafferty and Zhai*]**, présenté comme suit :

$$P_\mu(m_i/D) = \frac{tf(m_i,D)+\mu\ P(m_i/C)}{\sum_{m_i} tf(m_i,D)+\mu} \quad \textbf{(3.22)}$$

Avec :

$tf(m_i, D)$ Le nombre d'occurrence de n-gramme dans le document *D*.

$\sum_{m_i} tf(m_i, D) + \mu = |D|$ La taille de document *D*.

μ Est un paramètre appelé pseudo fréquence.

Le lissage de Laplace est un cas particulier de ce type de lissage **[Lafferty and Zhai].**

***3.2.2.7 Lissage Absolute Discounting* :** L'idée dans cette technique, consiste à minimiser les probabilités des n-grammes vus dans le corpus en retranchant une quantité constante à partir de leurs fréquences d'apparition, puis la redistribuer sur les valeurs des fréquences des n-grammes non vus.

Le lissage *Absolute Discounting* est similaire avec le lissage de *Jelinek-Mercer***,** mais leurs façons de diminuer les probabilités des n-grammes vus sont différentes **[*Lafferty and Zhai*].**

$$P_\delta(m_i/D) = \frac{max\,(tf\,(m_i,D)-\delta,0)+\delta|D|_\mu P(m_i/C)}{\sum_{m_i} tf\,(m_i,D)} \qquad \textbf{(3.23)}$$

Avec :

$\delta \in [0,1]$Est une constante de diminution.

$|D|_\mu$ Est le nombre des n-grammes différents apparaissant dans le document D.

3.2.3 Modélisation de langage en recherche d'information :

L'idée fondamentale de la modélisation de langage en *recherche d'information* est d'estimer un modèle de langage M_d pour chaque document d'une collection et cela dans le but de retrouver les documents pertinents pour une requête spécifiée **[Ponte and Croft. 1998].**

L'estimation des modèles de langage en recherche d'information s'est basée sur l'approche de modélisation statistique, développée en linguistique informatique. Plusieurs variantes de modèles ont été proposées et appliquées, introduites en premier par Ponte et Croft en 1998 puis Hiemstra en 1998, Song et Croft en 1999, Miller et al en 1999 et Berger et Lafferty en 1999.

La caractéristique commune entre ces différentes variantes est l'estimation de la pertinence d'un document face à une requête en mesurant la vraisemblance requête-document. Ces variantes ne tentent pas de modéliser directement la notion de pertinence, mais cela est en rapport avec la probabilité que la requête puisse être générée par le modèle de langage du document.

Le calcul de score de pertinence d'un document *D* pour une requête *Q* de *n* mots diffère d'un modèle à un autre.

3.2.3.1 Les différents modèles de langage en RI :

3.2.3.1.1 Modèle de Ponte et Croft : C'est le premier modèle de langage destiné pour la recherche d'information. Son idée de base est de déterminer la probabilité que la requête *Q* puisse être générée par le modèle de langage de document M_d et cela pour chaque document supposé pertinent, appartenant à la collection *C*.

La classification des documents selon leurs degrés de pertinence est déterminée par une fonctionn donnée par : $Score\ (Q, D) = P(Q/M_d)$

Pour ce calcul, Ponte et Croft supposent que la requête est un ensemble de mots choisis suivant le document le plus pertinent et ils prennent en compte même les mots absents dans la requête *Q.* Cela afin de faire la différence entre un document qui couvre beaucoup de sujets et un autre qui ne couvre que le sujet de la requête, en se basant sur le modèle de *Bernoulli multiple.*

$$P(Q/M_d) = P(m_1 m_2 \ldots . m_n/M_d) \times P(\neg m_{n+1} \neg m_{n+2} \ldots \neg m_t/M_d)$$
$$= \prod_{i=1}^{n} P(m_i/M_d) \times \prod_{i=n+1}^{t} P(\neg m_i/M_d)$$
$$= \prod_{i=1}^{n} P(m_i/M_d) \times \prod_{i=n+1}^{t} P(1 - P(M_i/M_d) \qquad \textbf{(3.21)}$$

Avec :

$m_1 m_2 \ldots \ldots . . m_n$ Les termes présents dans la requête *Q* ;

$m_{n+1} m_{n+2} \ldots \ldots . . m_t$ Les termes absents dans la requête *Q*.

Pour calculer la probabilité $P(m_i/Md)$, Ponte et Croft se basent sur l'estimation par le maximum de vraisemblance des termes de la requête dans le document *D* :

$$P(m_i/M_d) = \frac{tf\,(m_i,D)}{|D|} \qquad \textbf{(3.22)}$$

Avec :

$tf\,(m_i\ ,D)$ La fréquence du terme m_i dans le document *D* ;

$|D|$ Le nombre total des termes composant le document *D.*

Comme déjà vu, le problème de cette estimation est celui des fréquences zéro des termes absents dans le document *D*. Pour le corriger, on fait appel au modèle de langage de la collection *C*:

$$P(m_i/C) = \frac{tf\,(m_i,C)}{|C|} \qquad \textbf{(3.23)}$$

Avec :

$tf(m_i, C)$ la fréquence du terme m_i dans la collection *C* ;

$|C|$ le nombre total des termes de la collection *C*.

Pour que l'estimation par le maximum de vraisemblance soit plus confiante, Ponte et Croft utilisent une probabilité moyenne d'un terme m_i de tous les documents qui le contiennent, dans le calcul de $P(m_i/M_d)$, notée $P_{avg}(m_i)$:

$$P_{avg}(m_i) = \frac{\sum_{d(m_i \in d)} P_{MLE}(m_i/M_d)}{tf(m_i,d)} \qquad \textbf{(3.24)}$$

L'estimation de cette probabilité est basée sur la supposition que tous les documents qui contiennent le terme m_i, proviennent d'un même modèle de langage de document donc il n'y aurait aucune distinction entre ces documents malgré les fréquences différentes du terme m_i. Ceci cause du risque dans l'utilisation de cette probabilité pour le calcul de $P(m_i/M_d)$.

Dans le but de minimiser ce risque, une distribution géométrique dite *fonction de risque* est utilisée, associée à la probabilité moyenne du terme m_i dans le document *D*, calculée comme suit : $R(m_i, D) = \frac{1}{1+\bar{f}_{m_i}} \times (\frac{\bar{f}_{m_i}}{1+\bar{f}_{m_i}})^{tf(m_i,D)}$ **(3.25)**

Avec :

$\bar{f}_{m_i}$ la fréquence moyenne du terme m_i dans les documents qui le contient.

Enfin, d'après Ponte et Croft, l'estimation de la probabilité de produire les mots de la requête Q à partir d'un document *D* se base complètement sur leurs fréquences observées dans la collection et il n'est pas nécessaire d'apprendre aucun paramètre spécifique **[TEBRI Hamid, 2004]**. Ce qui est dit approche non paramétrique, donnée selon la formule:

$$P(m_i/M_d) = \begin{cases} P_{MLE}(m_i/D)^{(1-R(m_i,D))} \times P_{avg}(m_i)^{R(m_i,D)} & si\ tf(m_i,D) > 0 \\ \frac{tf(m_i,C)}{|C|} & sinon \end{cases} \qquad \textbf{(3.26)}$$

La combinaison $P_{avg}(m_i)^{R(m_i,D)}$ est définie comme étant une fonction de la fréquence du terme m_i dans le document *D* et dans la collection *C* **[TEBRI Hamid.2004]**, ajoutée pour contrer le risque de l'estimation de $P_{MLE}(m_i/D)$; ce qui ressemble au lissage par interpolation basée sur le produit.

3.2.3.1.2 Modèle de Song et Croft : Dans ce modèle, Song et Croft supposent l'indépendance des mots m_i *(termes)* de la requête utilisateur *Q*. Ils proposent le lissage de modèle de document en utilisant la technique de Good-Turing, afin d'effectuer des probabilités non nulles pour les mots de la requête absents dans le document correspondant, en ajustant la fréquence d'apparition *tf* de chaque mot m_i de la requête en tf^* [***Song and Croft***]:

$$tf^* = (tf + 1)\frac{E(N_{tf+1})}{E(N_{tf})} \qquad \textbf{(3.27)}$$

Avec :

N_{tf} le nombre de n-grammes de fréquence tf dans le document ;

$E(N_{tf})$ la valeur prévue de N_{tf}.

Une fonction dite de curve-fitting notée $s(N_{tf})$ est utilisée dans ce modèle, permettant le lissage de la valeur prévue de N_{tf}.

Si celle-ci est nulle, alors la probabilité d'un terme m_i de fréquence tf dans un document *D* de taille N_d est donnée selon la formule :

$$P_{Gd}(m_i/D) = \frac{tf^*}{N_d} = (tf + 1)\frac{S(N_{tf+1})}{S(N_{tf})\times N_d} \qquad \textbf{(3.28)}$$

D'après Song et Croft, le modèle de document n'est pas stable vu les termes qu'il ne contient pas *(termes absents).* A cet effet, il sera combiné avec le modèle de langage de toute la collection auquel le document correspondant appartient, c'est-à-dire, interpolation de la probabilité du mot m_i dans le document avec sa probabilité dans la collection.

Deux approches à appliquer pour faire cette interpolation:

1. une approche basée sur la somme, où la combinaison des deux modèles est comme suit :

   $$P_{SUM}(m_i/D) = \lambda_{document}\ (m_i/D) + (1 - \lambda_i)P_{collection}\ (m_i) \qquad \textbf{(3.29)}$$

2. une approche basée sur le produit représentée comme suit :

$$P_{PRODUCT}(m_i/D) = P_{document}\ \ (m_i/D)^{\lambda_i}\ P_{collection}\ (m_i)^{(1-\lambda_i)} \qquad \textbf{(3.30)}$$

Avec $\lambda_i \in [0,1]$ un paramètre de pondération.

La différence entre ces deux approches est que celle basée sur la somme donne des résultats de probabilités normalisés *(somme de probabilités égale à 1)*, contrairement à la deuxième approche.

Donc la probabilité de produire les mots de la requête *Q* à partir d'un document *D* suivant le modèle de Song et Croft est donnée comme suit:

$$P(m_i/D) = \lambda_i P_{GD}\,(m_i/D) + (1-\lambda_i)P(m_i/C) \qquad \textbf{(3.30)}$$

Où la valeur optimale du paramètre λ_i est déterminée empiriquement.

Dans le but de chercher une séquence composée de paire de mots (m_{i-1} m_i) dans un document *D* caractérisé par un modèle unigramme; Song et Croft proposent une extension de ce modèle en un modèle bigramme. Ce qui donne pour le calcul de la probabilité de cette séquence, la formule suivante:

$$P(m_{i-1}, m_i/D) = \lambda_1 P_{MLE}(m_i/D) + \lambda_2 P_{MLE}(m_{i-1}, m_i/D) \qquad \textbf{(3.31)}$$

Où : $\lambda_1 + \lambda_2 = 1$

Le but dans l'utilisation des paramètres λ_i estimés par une procédure automatique, appelée EM *(Expectation Maximization)*, est d'améliorer le calcul de la probabilité de la séquence de mots - $P(m_{i-1},m_i/D)$- ainsi que l'information recherchée ne sera pas perdue.

3.2.3.1.3 Modèle de Hiemstra *:* Comme dans le modèle de Song et Croft, Hiemstra traite la requête comme une séquence de mots indépendants **[Victor Lavrenko and Crof W]**. Dans ce modèle la probabilité de générer une requête Q sachant le modèle de document, est calculé selon la formule:

$$P(Q/M_d) = \prod_{i=1}^{n} P(m_i/M_d)^{tf(m_i,Q)} \qquad \textbf{(3.32)}$$

Où : $\boldsymbol{tf(m_i, Q)}$ est la fréquence du terme m_i dans la requête *Q*.

L'intégration de l'élément $tf(m_i, Q)$- la fréquence du terme m_i dans la requête *Q* - dans ce modèle était pour mieux modéliser la requête, dans laquelle certains mots peuvent être non pertinents.

Aussi pour l'estimation de $P(m_i/M_d)$, Hiemstra emploie l'approche par interpolation qui combine le modèle de langage de document avec le modèle de langage de collection. Il introduit un coefficient λ_{mi} pour chaque terme m_i de la requête afin d'estimer son importance **[TEBRI Hamid.2004]**.

Le calcule de la probabilité interpolée est alors exprimée par :

$$P(m_i/M_d) = \lambda_{mi} P_{MLE}(m_i/D) + (1-\lambda_{mi})P_{MLE}(m_i/C) \qquad \textbf{(3.33)}$$

Contrairement au modèle de Song et Croft, l'interpolation dans ce modèle est appliquée sans effectuer aucun lissage sur le modèle de langage de document, tel que

Hiemstra utilise directement la technique d'estimation par le maximum de vraisemblance pour le calcul de $P_{MLE}\,(m_i/D)$ et $P_{MLE}\,(m_i/C)$.

3.2.3.1.4 Modèle de Miller et al : Pour le calcul de $P(Q/M_d)$, Miller et al utilisent un modèle de Markov caché à deux états **[Boughanem and Kraaij]**, représenté comme suit :

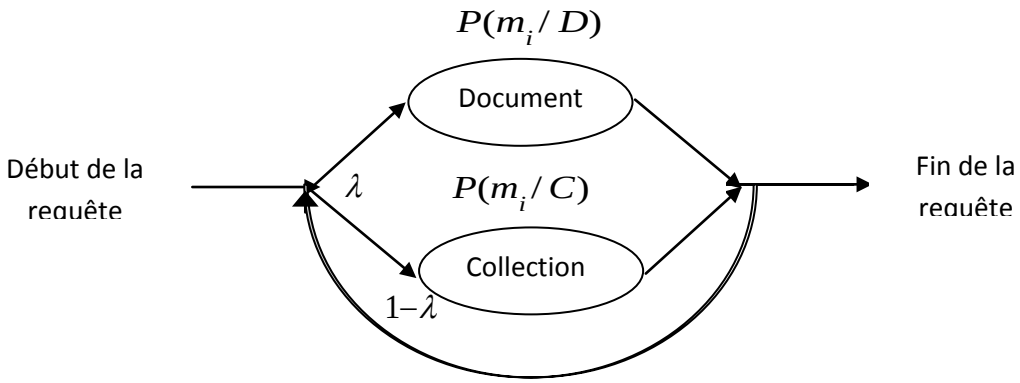

Figure 3.1 : Modèle de Markov caché à deux états.

Ce modèle correspond à la formule suivante :

$$P(Q/M_d) = \prod_{i=1}^{n} P(m_i/M_d) = \prod_{m\in Q}(\lambda_i P_{MLE}(m_i/D) + (1-\lambda_i)P_{MLE}(m_i/C)) \quad \textbf{(3.34)}$$

Ce qui est similaire au modèle de Hiemstra à quelques détails près [**[Boughanem and Kraaij]** La différence entre eux est dans leurs implantations.

3.2.3.1.5 Modèle de Berger et Lafferty : La modélisation de langage est une approche très efficace dans le domaine de la recherche d'information. Toutefois, les modèles précédents ne traitent pas toutes les formes et styles de requêtes, ni les problèmes de synonymies et de polysémies de termes : multiples termes partageant des significations semblables et le même terme ayant des significations multiples **[*Berger and Lafferty, 1999]*.**

Pour aborder ces problèmes, la recherche d'information est traitée comme un processus de traduction statistique, qui traduit la requête utilisateur *Q* en un document *D*. Cette idée peut être exprimée comme suit :

- L'utilisateur à un besoin d'information *B* ;
- À partir de ce besoin, l'utilisateur génère un fragment d'un document D_B supposé idéal ;
- Il choisit un ensemble de mots clés à partir du fragment désigné D_B et génère sa requête.

La tâche du processus de recherche d'information est alors de trouver les documents les plus similaires au fragment D_B et le calcul de leurs probabilités, sachant l'utilisateur et sa requête est donné selon la loi de *Bayes* :

$$P(Q,U) = \frac{P(Q/D,U)P(D/U)}{P(Q/U)} \quad \textbf{(3.35)}$$

La probabilité *P(Q/U)* est une constante fixée pour une requête et un utilisateur donnés, qu'on peut ignorer dans la classification des documents. Donc la probabilité de pertinence d'un document *D* face à une requête *Q* est donnée par :

$$P(D/Q) = P(Q/D)P(D) \quad \textbf{(3.36)}$$

Pour le calcul de la probabilité *P(Q/D),* **Berger et Lafferty** se sont inspirés du modèle combinatoire *(mixture model)* de l'approche traduction statistique; proposé par les travaux d'*IBM* **[*Berger and Lafferty*].** Leur modèle est le résultat d'une combinaison d'un modèle de langage de document $l(d_j/D)$ avec un modèle de traduction $t(m_i/d_j)$ qui permet la traduction des termes d_j (j varie de 1..r) du document *D* en termes m_i (i varie de 1..n); générant ainsi la requête *Q* . Ce qui donne d'une manière simplifiée:

$$P(Q/D) = \emptyset(n) \prod_{i=1}^{n} \sum_{j=1}^{r} t(m_i/d_j) l(d_j/D) \quad \textbf{(3.37)}$$

Avec :

$\emptyset(n)$ La probabilité de générer une requête de longueur *n* ;

r Le nombre de termes dans le document.

Pour l'appliquer à la recherche d'information, le lissage du modèle de Berger et Lafferty est effectué selon la technique d'interpolation linéaire; combinant un modèle unigramme $P(m_i)$ avec un modèle basé sur la traduction statistique, ce qui donne la formule suivante:

$$P(D, m_1 \ldots\ldots, m_n) = P(D) \prod_{i=1}^{n} \lambda P(m_i/D) + (1-\lambda) P(m_i) \quad \textbf{(3.38)}$$

Avec : $P(m_i/D) = \sum_{j=1}^{r} t(m_i/d_j) l(d_j/D)$

$P(D)$ la probabilité du document.

3.3 Classification des documents pertinents :

Quelque soit le modèle de langage utilisé, la tâche principale d'un système de recherche d'information est de présenter d'une manière ordonnée les différents documents satisfaisant un besoin en information d'un utilisateur. Cependant le processus de

classification de ces documents peut être modélisé par un formalisme proposé par Lafferty et Zhai [**Lafferty and Zhai],** basé sur la théorie de la décision bayésienne:

Soit $C = \{d_1, d2,\ldots, d_k\}$ une collection de *K* documents, chacun est caractérisé par une action $a(d_i)$ qui correspond à sa recherche.

Le problème est dans le choix de l'action $a(d_i)$ *(i varie de 1… k)* associée au document d_i *le* plus pertinent, noté d^*, face à une requête *Q* donnée **[*Chengxiang Zhai* .2001]**.

Le risque de choisir l'action $a(d_i)$ est alors donné par [**Lafferty and Zhai]** :

$$R(a(d_i)/Q,D) = \int L(a(d_i),\theta)P(\theta/Q,D)d\theta \qquad \textbf{(3.39)}$$

Avec : $L(a(d_i),\theta)$ une fonction dite de coût *(loss function)*, associe chaque action

Avec le paramètre θ qui représente la bonne décision.

Le document le plus pertinent est celui qui correspond à l'action $a(d_{i)}$dont la valeur de $a(d_i)$ risque est minimale :

$$a(d^*) = arg_{a(d_i)} \min R(a(d_i)/Q,D) \qquad \textbf{(3.40)}$$

Ceci est le critère général de la recherche d'information dans un cadre dit de minimisation de risque. Afin de le raffiner on suppose que pour une action $a(d_{i)}$, la valeur de la fonction du coût dépend des deux représentations de modèles de la requête et de document, notées respectivement $\theta_Q et\ \theta_D$avec une variable de pertinence *R* qui est en rapport avec leur comparaison **[*Chengxiang Zhai* .2003].**

La requête *Q* et le document *D*, sont générés selon le schéma suivant:

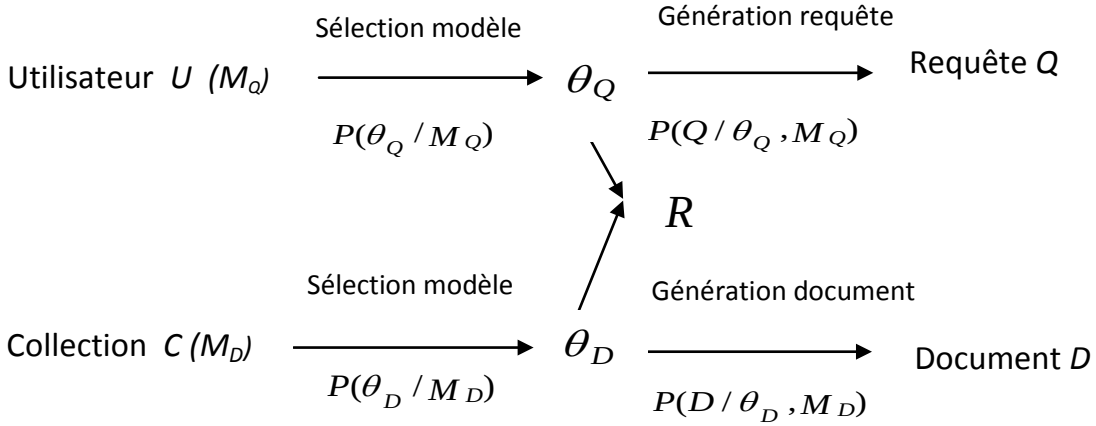

Figure 3.2 : Génération de la requête et le document.

$R \in \{0.1\}$ est une variable de pertinence estimée selon $P(R\backslash\theta_Q,\theta_D)$.

Le risque est alors une fonction de classification des documents, représente le score d'un document *D* pour une requête *Q*, calculé comme suit :

$$R(D,Q) = \sum_{R\in\{0.1\}} \int_{\theta Q} \int_{\theta D} \int_{\theta D} L(\theta_Q, \theta_D, R) P(\theta_D/D, D, M_d) P(R/\theta_Q, \theta_D) d\theta_D d\theta_Q \quad \textbf{(3.41)}$$

Avec : $L(\theta_D, \theta_D, R)$ la fonction de coût associée au fait de dire que le document représenté par θ_D est pertinent pour une requête représentée par θ_Q ;

$$P(\theta_Q/Q, M_Q) = \frac{P(Q/\theta_Q, M_Q) P(\theta_Q/M_Q)}{P(Q/M_Q)} \quad \textbf{(3.42)}$$

$$P(\theta_D/D, M_D) = \frac{P(D/\theta_D, M_D) P(\theta_D/M_D)}{P(Q/M_D)} \quad \textbf{(3.43)}$$

Pour évaluer la valeur de risque, deux simplifications liées à la fonction de coût permettent d'arriver à deux modèles différents :

1- La fonction de coût $L(\theta_Q, \theta_D, R)$ dépend seulement de la variable de pertinence *R*, définie comme suit :

$$L(\theta_Q, \theta_D, R) = \begin{cases} c_0 \; si \; R = 0 \\ c_1 \; si \; R = 1 \end{cases}$$

Avec : c_0 et c_1 des constantes.

La fonction de classification de documents est alors définie suivant la formule :

$$R(D,Q) = c_0 P(R = 0/Q, D) + c_1 P(R = 1/Q, D)$$

$$= c_0 + (c_1 - c_0) P(R = 1/Q, D) \quad \textbf{(3.44)}$$

On remarque que le critère de classification est basé sur la probabilité de pertinence sachant la requête et le document et si $c_1 > c_0$ on retrouve le classement défini par le cadre probabiliste classique $P(R\backslash Q, D)$ **[Chengxiang Zhai.2001].**

2- La fonction de coût dépend des deux paramètres $\theta_Q et\, \theta_D$ et pas de la variable de pertinence *R*, définie comme suit : $L(\theta_q, \theta_D, R) = c\Delta(\theta_q, \theta_D)$

Avec : *C* une constante positive.

$\Delta(\theta_Q, \theta_D$ Mesure la distance *(similarité)* entre les deux paramètres $\theta_Q et\, \theta_D$, calculée selon une fonction dite divergence de Kullback-Leibler *(KL-divergence)* mesure la divergence entre deux distributions de probabilité) **[Chengxiang Zhai.2001]**.

Définie selon la formule suivante :

$$\Delta(\theta_q, \theta_D) = \sum_{m_i \in Q} P(m_i/\theta_Q) \log \frac{P(m_i/\theta_Q)}{P(m_i/\theta_D)} \quad \textbf{(3.45)}$$

Dans ce cas, la fonction de classification mesure l'écart entropique sur le fait que le modèle de document M_D correspond bien au modèle de la requête M_Q, décrite comme suit :

$$R(D,Q) = -KL(Q,D) = -\sum_{m_i \in Q} P(m_i/\theta_Q, M_q) \log \frac{P(m_i/\theta_Q, M_q)}{P(m_i/\theta_D, M_D)} \quad \textbf{(3.46)}$$

Avec : *KL (Q, D)* mesure la divergence entre la requête *Q* et le document *D.*

Ces travaux représentent les critères de vraisemblance dans le cadre de minimisation de risque. Ils ont des apports importants en recherche d'information, comme, d'abord la représentation d'une nouvelle fonction de classification de documents selon leurs degrés de pertinence, basée sur la vraisemblance requête-document et en second lieu l'indication des liens entre la représentation des informations et les techniques de la modélisation de langage.

Ils sont utilisés dans l'approche modélisation de langage où ils permettent l'apport de modèle de requête afin de mieux cerner la notion de pertinence **[*Lafferty and Zhai*][2.7].**

3.4 Mesure de la qualité d'un modèle de langage :

La qualité d'un modèle de langage est exprimée en terme de ***perplexité***, qui représente le choix moyen auquel doit faire face un modèle lors d'une prédiction.

On interprète une perplexité de valeur *K* comme le fait que, pour un historique donné, le modèle doit choisir entre *K* mots équiprobables pour déterminer le prochain mot émis. **[*Yannick Estève.2002][*2.15].** Alors la perplexité, notée *PP***,** d'un modèle de langage est mesurée par : $PP = 2^{LP}$

Avec : ***LP*** une quantité appelée *logprob* qui est une approximation de la valeur moyenne d'émission d'une séquence *S* de *n* mots *(entropie)*, mesurée par :

$$LP = -\frac{1}{n} \log P(S) = -\frac{1}{n} \sum_{i=1}^{n} \log P(m_i/h) \quad \textbf{(3.47)}$$

Plus la valeur de la perplexité est petite, plus le pouvoir de prédiction de modèle de langage est grand.

3.5 Modèle de langue appliqué à la recherche d'information personnalisée

Dans la recherche d'information le contexte de l'usager est important mais peu de modèle opérationnel l'utilise de fait que les entités contextuelles à considérer sont encore difficiles à définir. Toute fois, depuis plusieurs années on voit apparaitre des approches qui intègrent certain aspects du contexte.

Ainsi **[Liu and Croft and B.W, 2001]** considère la similarité dans le corpus ce qui revient à regrouper les documents similaires pour former des agrégats ou cluster représentant les domaines ou thèmes abordés dans la collection. Ces agrégats sont utilisés pour étendre la porté des documents en lissant le modèle du document par le modèle de domaine. La contrainte de cette approche est que sa qualité dépend de la capacité de la méthode d'agrégation à capter les similarités du corpus.

Une autre approche considère le contexte cognitif de l'usager où on exploite l'expertise de l'usager pour désambigüiser la requête inférée par les documents consultés ultérieurement. La requête la plus compatible avec le profil est retenue. Les résultats de cette approche sont assez intéressants mais il demeure difficile à la mettre en œuvre.

Il est également possible de contextualiser une requête en exploitant des évidences implicites inférées du comportement de l'usager. Plusieurs facteurs peuvent refléter l'appréciation de l'usager **[Kelly, D., Teevan, J., 2003],** comme la consultation, la sauvegarde et le mouvement oculaire. Dans **[Saracevic, T, 1999],** on propose de redéfinir la requête d'après les documents consultés durant la session. Cependant, le résultat de cette redéfinition est considéré seulement pour la session courante, et il n'a pas un effet à plus long terme.

Une autre approche est réordonner les documents retrouvés selon le domaine d'intérêt de l'usager. Le domaine d'intérêt de l'usager peut être représenté de plusieurs façons: une hiérarchie de domaine préconisés par l'usager est utilisée **[Chirita, P.A., Paiu, R., Nejdl, W., Kohlschütter, C., 2005],** l'ensemble des documents consultés antérieurement par l'usager seront considérés comme une spécification du domaine **[Kim, H.-R., Chan, P.K.],** on infère un modèle probabiliste de l'ensemble des documents présents dans l'ordinateur de l'usager **[Teevan, J., Dumais, S.T., Horvitz, E.]**. la recherche d'information personnalisée est une partie de la recherche d'information contextuelle.

3.5.1 Modèles théoriques proposés par Hugues Bouchard et Jian-Yun Nie

Ils proposent trois méthodes d'intégration du modèle de domaine dans le processus de recherche :

- pour compléter le modèle de requête;
- pour réordonner les résultats;
- pour étendre la requête selon les relations lexicales extraites du domaine.

3.5.1.1 Compléter le modèle de la requête

Ayant un modèle qui caractérise le domaine de la requête, le modèle de langue de la requête n'est plus isolé, mais accompagné d'un autre modèle pour le domaine. Il apparaît donc légitime de le combiner au modèle du domaine. Une méthode intuitive est de lisser le modèle original de la requête par le modèle du domaine. Il en résulte une requête étendue qui, par l'entremise des termes additionnels du domaine, exprime mieux le besoin d'information de l'usager. En effet, l'ajout de termes spécifiques du domaine peut accroître le pouvoir d'expression de la requête tout en la désambiguïsant.

Intuitivement, l'approche adoptée se résume à une expansion de requête réalisée dans le cadre formel des modèles de langue. Mais cette expansion dépend du domaine. En effet, en interpolant le modèle du domaine au modèle original de la requête, on redistribue une partie de la masse de probabilité du modèle de la requête sur les termes caractéristique du domaine. Ces derniers sont les termes qui auraient pu être utilisés dans la requête par l'usager. Formellement, le nouveau modèle de la requête peut être obtenu comme suit :

$$\theta'_{\mathrm{Q}} = (1-\alpha)\theta_{\mathrm{Q}} + \alpha\theta_{\mathrm{Dom}} \qquad \textbf{(3.48)}$$

Où

- θQ est le modèle original de la requête qui est estimé directement par la fréquence relative des termes dans Q
- θ_{Dom} est le modèle du domaine, et
- θ'_Q est le modèle engendré par les deux précédents.
- α est un paramètre de lissage.

Bien entendu, θ_{Dom} n'est pas le seul modèle pouvant être combiné à θ_Q. Un autre modèle fort utile est celui construit sur une rétroaction de pertinence, c'est-à-dire à partir des documents qui se retrouvent en tête de la liste de résultats. Ainsi, en utilisant les n premiers documents retrouvés par la requête initiale, nous estimons un modèle additionnel θ_R. Ce

modèle peut être combiné aux précédents (via un autre paramètre de lissage β), donnant le modèle suivant pour la requête :

$$\theta'_Q = (1 - \alpha - \beta)\theta_Q + \alpha\theta_{Dom} + \beta\theta_R \qquad \textbf{(3.49)}$$

Tout comme θ_{Dom}, θ_R est estimé en appliquant l'algorithme EM sur les documents considérés. L'intérêt du modèle mixte défini en (**3.49**) est de considérer deux sources d'informations distinctes, soient, qui modélise le discours du domaine, et qui reflète ce que l'usager recherche. L'équation (**3.49**) est donc un modèle plus complet de la requête.

En pratique, un domaine peut être plus ou moins large. Pour un domaine large, les documents exemples ne sont pas concentrés sur certains thèmes, et le vocabulaire peut disperser. Un tel modèle de domaine complet pourrait introduire du bruit dans le modèle de requête lorsqu'il est interpolé avec le modèle de la requête. Pour résoudre ce problème, il est également possible de créer un sous domaine selon la requête, en considérant seulement les n premiers documents du domaine jugés les plus pertinents pour la requête. Cette approche est justifiée par le fait que l'opération d'expansion de requête est une opération critique.

Les termes qui y sont introduits doivent être pertinents pour la requête courante, sans quoi un glissement thématique de la requête se produit.

Pour un domaine aussi vague que « Science et Technologie » par exemple, les documents peuvent porter sur des thèmes très différents.

Ainsi, il est avantageux de limiter le discours du domaine aux documents qui s'apparentent le plus à la requête.

Une fois le nouveau modèle de la requête θ'_Q obtenu, ils ont utilisé une KL-divergence négative pour déterminer le rang de chaque document de la collection. Le pointage d'un document D pour une requête Q du domaine Dom est déterminé comme suit :

$$Score\ (Q, D, Dom) = \sum_{t \in V} P\left(t \backslash \theta'_Q\right) \log \frac{P(t \backslash \theta'_D)}{P(t \backslash \theta'_Q)} \qquad \textbf{(3.50)}$$

$$\propto \sum_{t \in V} P\left(t \backslash \theta'_Q\right) \log P(t \backslash \theta'_D)$$

θ'_D Est le modèle lissé du document tel que défini par le lissage de Jelinek-Mercer.

$$P(t \backslash \theta'_D) = (1 - \boldsymbol{\lambda})\boldsymbol{P}(\boldsymbol{t} \backslash \theta_{\boldsymbol{D}}) + \boldsymbol{\lambda}\boldsymbol{P}(\boldsymbol{t} \backslash \theta_{\boldsymbol{c}})$$

3.5.2.2 Réordonner les documents retrouvés

Une seconde stratégie est d'utiliser le modèle du domaine pour réordonner les documents retrouvés. Les documents sont d'abord ordonnés selon leur relation à la requête. Ensuite, l'ordre des documents est modifié d'après leur correspondance avec le domaine. Cette approche permet de favoriser les documents qui s'apparentent les plus au domaine de la requête.

La fonction d'ordonnancement proposée est une combinaison des KL-divergences négatives, reflétant respectivement la correspondance du document à la requête Q et au domaine Dom. la fonction suivante est utilisé pour calculer un nouveau score :

$$score(Q, D, Dom) = -[(1 - \boldsymbol{\lambda})\underbrace{\boldsymbol{KL}\left(\theta_{\boldsymbol{Q}} \backslash\backslash \theta'_{\boldsymbol{D}}\right)}_{\text{Similarité Q-D}} + \boldsymbol{\lambda}\underbrace{\boldsymbol{KL}(\theta_{\boldsymbol{D}} \backslash\backslash \theta'_{\boldsymbol{Dom}})}_{\text{Similarité D-Dom}} \qquad \textbf{(3.50)}$$

Dans cette fonction, un coefficient $\boldsymbol{\lambda}$ est utilisé pour contrôler l'importance relative accordée au domaine. Lorsque $\boldsymbol{\lambda}$ = 0, le modèle redevient l'approche de base, qui ne considère que la requête et le document.

3.5.2.3. Exploiter les relations lexicales du domaine

Dans les deux approches précédentes, la distribution de termes dans les documents du domaine est exploitée. Mis à part une distribution particulière, les documents du domaine peuvent aussi nous révéler les relations possibles entre les termes. Par exemple, dans le domaine « Finance », le terme « budget » est fortement relié au terme « planification ». Cette relation peut être utilisée pour effectuer une expansion de la requête sur « planification ». Ainsi, dans ce troisième modèle, on exploite les dépendances lexicales du domaine.

Comme nous avons mentionné, il est possible d'exploiter un thésaurus pour obtenir des relations lexicales. Dans cette étude, ces relations sont extraites selon les cooccurrences. Inspiré des travaux **[*Berger and Lafferty, 1999]*** et [**Cao, G., Nie, J.-Y., Bai, J., 2005**], l'idée est d'étendre la requête avec tous les termes fortement corrélés aux mots-clefs qui la composent. Intuitivement, il s'agit de lisser le modèle du document non pas uniformément, mais plutôt suivant les dépendances lexicales entre les termes dans le domaine. Durant le lissage, un terme fortement lié se voit attribuer une probabilité plus grande qu'un autre terme non relié. Ainsi, nous pouvons parler d'un lissage sémantique.

Concrètement, étant donné une requête Q et le domaine Dom, la probabilité d'observer le document D peut être exprimée comme suit :

$$P(D\backslash Q, Dom) = \frac{P(Q\backslash D,Dom)P(D\backslash Dom)}{P(Q\backslash Dom)} \infty P(Q\backslash D, Dom)P(D\backslash Dom) \qquad \textbf{(3.51)}$$

La probabilité P (D\ Dom) peut être estimée comme une probabilité de génération, soit :

$$P(D\backslash Dom) = \prod_{t\in D} P(t\backslash \theta_{Dom})^{c(t:D)} \qquad \textbf{(3.52)}$$

La probabilité P (Q\D,Dom) traduit la probabilité de générer la requête étant donné le document et les dépendances lexicales inférées du domaine. Le modèle combinant D et Dom est dénoté par le modèle de dépendance θ D,Dom. La probabilité de chaque terme t de Q dans ce modèle est estimé en considérant sa dépendance t Dom(t |d) à chaque terme dans D. Ainsi :

$$P\left(t\backslash \theta_{D,Dom}\right) = \sum_{d\in D} t_{Dom}\,(t\backslash d)P(d\backslash \theta'_{D}) \qquad \textbf{(3.53)}$$

où θ'D est un modèle lissé du document et $t_{Dom}\,(t|d)$ est la probabilité de dépendance de t à d estimée dans Dom. Cette probabilité de dépendance est basée sur la cooccurrence des termes dans les documents du domaine.

Dans cette étude, ils ont considéré une fenêtre de 5 mots pour la cooccurrence. Soit c (<t,d> ;D), le nombre de fois que **t** co-occurre avec **d** dans un document D du domaine Dom. Nous définissons la dépendance lexicale comme suit :

$$t_{Dom}\,(t\backslash d) = \frac{\sum_{\boldsymbol{D\in Dom}} \boldsymbol{c}((\boldsymbol{t},\boldsymbol{d}):\boldsymbol{D})}{\sum_{\boldsymbol{t'}\in \boldsymbol{V}} \sum \boldsymbol{C}((\boldsymbol{t'},\boldsymbol{d}):\boldsymbol{D})} \qquad \textbf{(3.54)}$$

Remarquons que les documents du domaine peuvent couvrir seulement une partie de relations lexicales utiles. Une autre partie de relations utiles peut se trouver dans la langue générale. Ainsi, nous devons aussi considérer les dépendances lexicales dans cette dernière, reflétée par toute la collection. Le modèle de dépendance est donc redéfini comme suit :

$$P\left(t\backslash \theta_{D,Dom}\right) = \sum_{d\in D}[(1-\mu)t_{Dom}\,(t\backslash d) + \mu + t_c(t\backslash d)]P(d\backslash \theta'_{\mathrm{D}}) \qquad \textbf{(3.55)}$$

Finalement, en plus de l'expansion utilisant les relations lexicales, le modèle $\delta KL(\theta_D \backslash\backslash \theta_{Dom})$'D,Dom peut aussi être lissé par le modèle uni-gramme du document θ'D. Ainsi, le modèle final utilisé est le suivant :

$$P\left(t\backslash \theta''_{D,Dom}\right) = (1-\lambda)P\left(t\backslash \theta'_{\mathrm{D,Dom}}\right) + \lambda P(t\backslash \theta'_{\mathrm{D}}) \qquad \textbf{(3.56)}$$

Où λ est un paramètre de lissage. Par conséquent, P(Q|D) est estimé comme suit :

$$\boldsymbol{P(t\backslash D, Dom)} = \prod_t \boldsymbol{P(t\backslash \theta''_{D,Dom})^{c(t;Q)}} \qquad \textbf{(3.57)}$$

Substituant les équations **(3.53)** et **(3.57)** dans l'expression logarithmique de **(3.52),** on obtient :

$$\log P(D\backslash Q, Dom) = \sum_{t\in Q} c(t;Q) \log P\left(t\backslash \theta''_{D,Dom}\right) + \sum_{t\in D} c(t;D) \log P(t\backslash \theta'_{Dom})$$

$$\propto \backslash Q \backslash \sum_{t\in Q} P\left(t\backslash \theta_Q\right) \log P\left(t\backslash \theta''_{D,Dom}\right) + \backslash D \backslash \sum_{t\in D} P(t\backslash \theta_D) \log P(t\backslash \theta'_{Dom}) \qquad \textbf{(3.58)}$$

Les deux composantes de l'équation (20) sont en fait $KL\left(\theta_Q \backslash\backslash \theta''_{D,Dom}\right)$ et $\delta KL(\theta_D \backslash\backslash \theta_{Dom})$ Multipliés par deux constantes reliées à la requête et au document. Pour simplifier, on suppose qu'elles sont invariables à travers la collection. On les dénote par (1-δ) et δ. Ainsi :

$$Score\ (Q, D, Dom) = -[(1-\delta) KL\left(\theta_Q \backslash\backslash \theta''_{D,Dom}\right) + \delta KL(\theta_D \backslash\backslash \theta_{Dom})] \qquad \textbf{(3.59)}$$

Finalement, soulignons que lorsque λ= 1, le modèle de dépendance est ignoré. Le modèle défini en **(3.59)** se réduit alors au modèle de ré-ordonnancement défini en **(3.51)**.

Conclusion :

Dans ce chapitre, nous avons présenté les modèles de langage, nous avons commencé par une présentation de l'approche modélisation de langage ainsi que ses différentes variantes dans le domaine de la linguistique informatique. Nous avons poursuivis par une présentation des différentes techniques de lissage permettant l'amélioration des modèles proposés et par la représentation des variantes de cette approche en recherche d'information. Nous avons introduit la notion de la classification des documents pertinents ainsi que la mesure permettant d'évaluer les performances des modèles de langage. Enfin, nous avons présenté une approche d'application des modèles de langue dans la recherche d'information personnalisée.

Modèle de langue appliqué à la recherche d'information personnalisée

Introduction

Dans la partie théorique nous avons présenté les concepts relatifs à la recherche d'information personnalisée ainsi que les modèles de langue.

Nous avons consacré cette partie à présenter notre travail qui consiste à proposer un système de recherche d'information personnalisée en se basant sur un modèle de langue.

Notre objectif est alors, de proposer une approche pour le calcul de la pertinence d'un document face à une requête utilisateur avec une prise en compte du profil de ce dernier.

La notion de pertinence d'un document face à une requête dépend uniquement des attentes de l'utilisateur. C'est pour cela que l'intégration du profil utilisateur dans le processus de la recherche est une nécessité afin d'améliorer la pertinence. Les modèles de langage qui sont des modèles probabilistes, apparus en 1998, offrent des performances consenties dans ce domaine. Afin de mesurer la pertinence d'un document face à une requête ils proposent deux modèles: un modèle de document et un modèle de requête. Aucun de ces modèles n'a pris en compte le profil de l'utilisateur. C'est dans ce contexte que s'inscrit notre contribution. Dans notre cas, pour la personnalisation du SRI, le profil utilisateur sera intégré dans la phase d'appariement document requête.

4.1 Modélisation du profil utilisateur:

Le profil utilisateur correspond à son centre d'intérêt qui est représenté sous forme d'un vecteur de termes **[Salton, 71]**.

4.2 Construction du profil (vecteur du centre d'intérêt):

Pour construire le centre d'intérêt d'un utilisateur donné nous considérons une requête $Q = m_1 m_2 \dots m_n$. Après le choix des documents pertinents face à cette requête, un centre d'intérêt C_k est défini comme un vecteur de termes pondérés en appliquant la formule BM25 **[Robertson et al. 1997]** définit comme suit :

$$W_{ki} = Log\left(\frac{\frac{(r+0.5)}{(R-r+0.5)}}{\frac{(n-r+0.5)}{(N-n-(R-n)+0.5)}}\right) \quad \textbf{(4.1)}$$

Avec :

N : Le nombre total de documents de la collection.

n: Le nombre de documents de la collection contenant le mot m_i.

r: Le nombre de documents pertinents contenant le mot m_i.

R : Le nombre de documents pertinents face à la requête utilisateur.

4.3 Description du modèle proposé:

Dans notre cas nous considérons une requête Q, l'objectif du SRIP est d'identifier les documents D qui sont appropriés au centre d'intérêt de l'utilisateur C_k. D'un point de vue probabiliste, cet objectif peut être formulé ainsi :

Etant donné une requête Q, l'objectif du SRIP est de retrouver les documents d_j pour lesquels la probabilité de pertinence, considérant la requête Q et le centre d'intérêt de l'utilisateur C_k, noté $P(d_j|q,c_k)$, est la plus élevée.

Cette probabilité est formulée en appliquant la loi de Bayes comme suit :

$$P(d_j/q, c_k) = \frac{P(q/d_j,c_k)P(d_j,c_k)}{P(q,c_k)} \quad \textbf{(4.2)}$$

Etant donné que le dénominateur $P(q,c_k)$ est constant pour une requête Q et un centre d'intérêt utilisateur c_k donné, nous pouvons tenir compte uniquement du numérateur afin d'ordonner les documents selon leur pertinence.

Ainsi la formule peut être définie comme suit :

$$P(d_j/q, c_k) = P(q/d_j, c_k)P(d_j, c_k) = P(q/d_j)P(q/c_k)P(d_j/c_k) \quad \textbf{(4.3)}$$

Pour quantifier chacun des paramètres de la formule **(4.3)**, on se base sur le modèle de langue.

4.3.1 Calcul de P (Q/d_j):

Le modèle de langue va ordonner chaque document D de la collection C suivant leur capacité à générer la requête Q. Il s'agit alors d'estimer la probabilité de génération $P(Q/D)$. Ainsi, pour une requête Q=$m_1 m_2 \dots m_n$, cette probabilité de génération est estimée comme suit:

$$P(Q\backslash D) = P(m_1 m_2 \dots m_n \backslash D) = \prod_{i=1}^{n} P(m_i \backslash D) \qquad \textbf{(4.4)}$$

La probabilité de générer la suite de mots composant la requête *Q* à partir du modèle de document, plusieurs mesures de probabilité ont été proposées. Nous avons opté pour celle de Ponte & Croft qui est l'estimation par le maximum de vraisemblance *(MLE)* :

$$P_{MLE}(m_i \backslash M_d) = \frac{tf\ (m_i, D)}{|D|} \qquad \textbf{(4.5)}$$

Avec

$tf\ (m_i, D)$: La fréquence du terme m_i dans le document

$|D|$: La taille du document.

- si un mot m_i de la requête est absent d'un document, alors sa probabilité $P\ (m_i/D) = 0$. Or, un document dans lequel un terme de la requête est absent peut aussi être pertinent. Pour éviter ce problème, nous optons pour le lissage du modèle du document par le modèle de la collection C selon l'approche d'interpolation linéaire de Jelinek-Mercer,
- soit :

$$\boldsymbol{P(mi \backslash D)} = \lambda\, P(m_i \backslash D) + \ (1 - \lambda) P(m_i \backslash C) \qquad \textbf{(4.6)}$$

- Le modèle de langage général de collection *C* sera basé sur l'estimation par le maximum de vraisemblance, donné comme suit:

$$P_{MLE}(m_i/C) = \frac{tf(m_i,C)}{|C|} \quad \textbf{(4.7)}$$

Avec : $tf(m_i, C)$ La fréquence du mot m_i dans la collection *C* ;

$|C|$ Le nombre total des mots de la collection *C*.

4.3.2 Calcul de $P(Q\backslash C_k)$:

De manière similaire que $P(Q/D)$, $P(Q/C_k)$ est calculé selon la formule suivante pour une requête Q=$m_1 m_2 \dots m_n$:

$$P(Q\backslash C_k) = P(m_1 m_2 \dots m_n \backslash C_k) = \prod_{i=1}^{n} P(m_i/C_k) \quad \textbf{(4.8)}$$

La probabilité de $P(m_i/C_k)$ est estimé par le poids du mot m_i dans le vecteur C_k comme suit:

$$P(m_i\ /\ C_k) = \begin{cases} W_{ki} & si\ m_i \in C_k \\ 0 & sinon \end{cases} \quad \textbf{(4.9)}$$

4.3.3 Calcul de $P(D\backslash C_k)$:

De manière similaire, si D=$m_1 m_2 \dots m_l$, $P\ (D/C_k)$ se fait comme suit:

$$P(D\backslash C_k) = P(m_1 m_2 \dots m_l \backslash C_k) = \prod_{i=1}^{l} P(m_i/C_k) \quad \textbf{(4.10)}$$

Et l'estimation de la probabilité de $P(m_i/C_k\ )$ est donné par la formule **(4.9)**.

4.4 Exemple illustratif:

Pour mieux illustrer nos propos, nous considérons une collection *C* de huit (8) documents D_i, où la distribution des termes (mots simples indépendants) est comme suit :

D_1= {2informatique, 10language, 4web, 3java, 8programmation}

D_2= {5java, 9île, 7tourisme, 4hôtel, 5vacance}

D_3= {2java, 1île, 4tourisme, 7hôtel, 6vacance, 1voyage}

D_4= {2informatique, 7langage, 6java, 8programmation}

D_5= {5informatique, 3web, 16java, 4programmation}

D_6= {3java, 8île, 6tourisme, 3hôtel, 6voyage}

D_7= {1informatique, 1langage, 1programmation, 2ile, 3tourisme, 1hôtel, 1vacance, 1voyage}

D_8= {6informatique, 8langage, 3web, 8programmation}

Dans cette collection, nous constatons d'emblé qu'il ya deux centre d'intérêt qui se dessine le premier qui relève de l'informatique et le deuxième qui s'intéresse au tourisme.

Pour chaque terme de chaque document nous avons calculé son poids par la mesure *(MLE)* donné en **(4.5)** et **(4.7)** nous avons obtenu les résultats sont présentés dans le tableau **(4.1)** *:*

	informatique	Langage	web	Java	programmation	île	tourisme	Hôtel	Vacance	Voyage
D1	**0,07**	**0,37**	**0,15**	**0,11**	**0,30**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
D2	**0,00**	**0,00**	**0,00**	**0,17**	**0,00**	**0,30**	**0,23**	**0,13**	**0,17**	**0,00**
D3	**0,00**	**0,00**	**0,00**	**0,10**	**0,00**	**0,05**	**0,19**	**0,33**	**0,29**	**0,05**
D4	**0,09**	**0,30**	**0,00**	**0,26**	**0,35**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
D5	**0,18**	**0,00**	**0,11**	**0,57**	**0,14**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
D6	**0,00**	**0,00**	**0,00**	**0,12**	**0,00**	**0,31**	**0,23**	**0,12**	**0,00**	**0,23**
D7	**0,09**	**0,09**	**0,00**	**0,00**	**0,09**	**0,18**	**0,27**	**0,09**	**0,09**	**0,09**
D8	**0,24**	**0,32**	**0,12**	**0,00**	**0,32**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**
Poids (m_i/C)	**0,08**	**0,14**	**0,05**	**0,18**	**0,15**	**0,10**	**0,10**	**0,08**	**0,06**	**0,04**

Tableau 4.1 : Poids des mots dans les documents et dans la collection

4.4.1 Définition des centres d'intérêt:

Pour la définition de nos centres d'intérêt on propose deux requêtes d'apprentissage Q1 et Q2 comme suit:

Q1= informatique java

Q2= île java

Les valeurs de pertinence retournées pour ces requêtes sont dans le tableau **(4.2):**

	P (Q1/D)	P (Q2/D)
D1	0,03	0
D2	0	0,05
D3	0	0
D4	0,09	0
D5	0,08	0
D6	0	0,04
D7	0	0
D8	0	0

Tableau 4.2 : probabilité d'une requête face à un document P (Q1/D) et P (Q2/D)

*4.4.1.1 Choix des termes pertinents***:**

Dans notre cas, un document est jugé pertinent si sa valeur de pertinence est supérieure à un seuil prédéfini que nous avons fixé à **0.01**.

D'après les valeurs présenté dans le tableau **(4.2)**, pour Q1, D1 D4 D5 sont les documents les plus pertinents. Pour Q2 c'est D2 D6 qui sont les plus pertinents.

*4.4.1.2 Construction du vecteur centre d'intérêt***:**

Comme il a été défini dans la section 4.1, le centre d'intérêt se présente sous forme d'un vecteur de termes pondérés issus des documents jugés pertinents face à la requête.

Les centres d'intérêt C_1 et C_2 ainsi obtenus sont représentés dans les tableaux **(4.3)** et **(4.4)** suivants:

	Informatique	Langage	Web	java	Programmation
w1i	0,99	0,37	0,70	0,70	0,99

Tableau 4.3 : Poids des termes de C1 selon BM25 avec R=3 et N=8

	Java	Ile	Tourisme	hôtel	Vacance	Voyage
W2i	0,44	0,95	0,95	0,95	0,26	0,26

Tableau 4.4 : Poids des termes de C_2 selon BM25 avec R=2 et N=8

Les documents qui contiennent des mots ne figurant pas dans le vecteur du centre d'intérêt considéré se verront attribué une probabilité nulle. Nous avons calculé la probabilité de générer chaque document sachant les centres d'intérêts C_1 et C_2. Les résultats sont présentés dans le tableau **(4.5)**.

	P (D/ C_1)	P (D/ C_2)
D1	0,18	0
D2	0	0,10
D3	0	0,03
D4	0,25	0
D5	0,48	0
D6	0	0,10
D7	0	0
D8	0,25	0

Tableau 4.5 : Pertinence des documents de la collection selon C_1 et C_2

4.4.2 Tests et interprétation des résultats:

Maintenant que les trois paramètres de notre formule sont calculés nous pouvons procéder à la mesure du score de pertinence avec le modèle proposé. Pour illustrer nos propos, nous proposons trois requêtes de test Q1, Q2, et Q3 comme suit:

Q3=java,

Q4=informatique web,

Q5= voyage ile java.

Nous calculons probabilités des termes des requêtes dans les documents avec lissage selon la formule **(4.6)** où nous avons fixé λ**=0.5** et les résultats sont présentés dans le tableau **(4. 6):**

	Q3=java	***Q4= informatique web***	***Q5=voyage ile java***
D1	0,1472	0,0077	0,0004
D2	0,1833	0,0022	0,0004
D3	0,1440	0,0022	0,0005
D4	0,2351	0,0022	0,0004
D5	0,4059	0,0118	0,0004
D6	0,1551	0,0022	0,0045
D7	0,0916	0,0022	0,0004
D8	0,0916	0,0166	0,0004

Tableau 4.6 : Probabilités des termes des requêtes dans les documents après lissage.

Nous présentons dans le tableau **(4.7)** les probabilités des termes de chaque requête sachant les centres d'intérêts C_1 et C_2 calculés selon la formule **(4.8).**

Q3=*java*	$P(Q3\backslash C_1)$	$P(Q3\backslash C_2)$
D1	0,70	0,44
D2	0,70	0,44
D3	0,70	0,44
D4	0,70	0,44
D5	0,70	0,44
D6	0,70	0,44
D7	0	0
D8	0,70	0
Q4=informatique web	$P(Q4\backslash C_1)$	$P(Q4\backslash C_2)$
D1	0,69	0
D2	0	0
D3	0	0
D4	0,69	0
D5	0,69	0
D6	0	0
D7	0,69	0
D8	0,69	0
Q5=voyage île java	$P(Q5\backslash C_1)$	$P(Q5\backslash C_2)$
D1	0	0
D2	0	0,11
D3	0	0,11
D4	0	0
D5	0	0
D6	0	0,11
D7	0	0
D8	0	0

Tableau 4.7 : Probabilités des termes des requêtes sachant les centres d'intérêts C_1 et C_2.

Et selon la formule **(4.5)** nous allons obtenir les résultats suivant:

Q_3=java						
document	*P (D/Q3, C1)*	*Rang*	*P (D/Q3, C2)*	*Rang*	*P (Q3, D)*	*Rang*
D1	0,018	3	0		0,111	**5**
D2	0		0,008	**1**	0,167	**3**
D3	0		0,002	**3**	0,095	**6**
D4	0,042	2	0		0,261	**2**
D5	0,136	1	0		0,571	**1**
D6	0		0,007	**2**	0,115	**4**
D7	0		0		0	
D8	0,016	4	0		0	
Q_4=informatique web						
document	*P (D/Q4, C1)*	*Rang*	*P (D/Q4, C2)*	*Rang*	*P (Q4,D)*	*Rang*
D1	0,0009	**3**	0		0,0110	**3**
D2	0,0000		0		0	
D3	0,0000		0		0	
D4	0,0004	**4**	0		0	
D5	0,0039	**1**	0		0,0191	**2**
D6	0,0000		0		0	
D7	0,0000		0		0	
D8	0,0029	**2**	0		0,0288	**1**
Q_5=voyage île java						
document	*P (D/Q5, C1)*	*Rang*	*P (D/Q5, C2)*	*Rang*	*P (Q5, D)*	*Rang*
D1	0		0		0	
D2	0		0,000004	**2**	0	
D3	0		0,000001	**3**	0,000216	**2**
D4	0		0		0	
D5	0		0		0	
D6	0		0,000048	**1**	0,008193	**1**
D7	0		0		0	
D8	0		0		0	

Tableau 4.8 : Probabilité des termes des requêtes selon le modèle de langue de base et le modèle proposé.

Compte tenu des résultats du tableau **(4.8)**:

1- La requête **Q3=Java**. Le terme java figure dans C_1 et C_2.

- Avec le centre d'intérêt C_1, les documents retournés sont D5 D4 D1 D8. Sachant que C_1 est un profil informatique, on remarque bien que les documents retournés sont ceux jugés pertinents pour ce profil. Bien que le document D8 ne contienne pas le mot java, il est quand même retourné car il peut être intéressant pour le profil C_1. Ceci est rendu possible grâce au lissage du modèle proposé.

  On remarque aussi que les documents D2, D3, D6 D7 ne sont pas retournés du fait qu'ils contiennent des termes absents dans le vecteur C_1.
- Avec le centre d'intérêt C_2, les documents retournés sont: D6 D3 D2. Nous remarquons ici aussi que ces documents sont ceux du profil tourisme et que ceux qui traitent de l'informatique sont exclus pour la même raison citée précédemment. L'ordre dans lequel les documents sont présentés est toujours proportionnel au produit de leur pertinence dans le centre d'intérêt par celle de la requête dans le document.

Ces résultats sont intéressants lorsque nous les comparons à ceux retournés par le modèle de base (D5 D4 D2 D6 D1).

Nous constatons une amélioration de la pertinence pour la requête Q3=Java qui est un terme commun au deux profil C_1 et C_2. Avec C_1, elle retourne les documents informatiques et avec C_2 elle retourne ceux du tourisme.

2- La requête Q4=***informatique web,*** correspond au profil utilisateur C_1.

- Le modèle proposé pris avec C_1 retourne les documents D5 D8 D1 D4 qui sont les plus pertinents pour le profil informatique(C_1).

  Si on compare ce résultat à celui retourné par le modèle de base (D8 D5 D1) on remarque que l'ordre de pertinence différent en effet avec le modèle de base c'est D8 qui est le plus pertinents mais il reste un document moins important que D5 dans le profil C_1 selon les résultats du tableau **(4.5)** donc le modèle proposé reclasse les

documents selon leurs importances aussi bien pour la requête que pour le centre d'intérêt.

On remarque aussi que le D4 n'a pas été retourné par le modèle de base et a été retourné par le modèle proposé cela est dû au lissage appliqué à ce dernier.

- Le modèle proposé pris avec C_2, la requête Q4 ne retourne aucun document du fait qu'aucun terme de la requête ne figure dans le vecteur du centre d'intérêt C_2.

Dans ce cas aussi nous pouvons affirmer que les hypothèses de notre approche sont vérifiées. Nous constatons que les documents renvoyés avec l'intégration du centre d'intérêt sont plus pertinents au profil de l'utilisateur que ceux renvoyé par le modèle de base avec un ordre plus intéressant.

3- La requête **Q5=voyage île java.** Les termes de cette requête figurent dans le vecteur centre d'intérêt C_2:

- le modèle proposé pris avec C_1 ne retourne aucun document du fait que les termes voyage et île ne figure pas dans le vecteur centre d'intérêt C_1.
- le modèle proposé pris avec C_2 retourne les documents D6 D2 D3 dans cet ordre. Effectivement, D6 est le document le plus pertinent pour le profil C_2 et la requête Q5. lorsque on compare ces résultats à ceux retournés par le modèle de base (D6 D3), on remarque que le document D2 n'est pas retourné par le modèle de base mais il est retourné par le modèle proposé, ceci grâce au lissage appliqué à ce dernier. l'ordre de D2 est plus important par le modèle proposé parce que ce document est plus important que D3 dans le profil C_2 comme illustré dans le tableau **(4.7).**

  Pour cette requête aussi, nous avons obtenu des résultats satisfaisants puisque le modèle proposé retourne les documents jugés pertinents aussi bien pour la requête que le profil C_2.

  Les résultats obtenus démontrent que l'intégration du centre d'intérêt de l'utilisateur améliore la pertinence des documents retournés pour chaque profil. Car en effet, il ne retourne que les documents jugés pertinent pour l'utilisateur dans l'ordre de leur importance aussi bien pour la requête que le centre d'intérêt de l'utilisateur.

Conclusion

Dans ce chapitre, nous avons présenté notre contribution qui a consisté à proposer une approche pour la recherche d'information personnalisée basé sur le modèle de langue.

Nous avons introduit un modèle de langue pour le calcul de la pertinence d'un document face à une requête utilisateur avec une prise en compte du profil de ce dernier.

Le modèle proposé retourne les documents jugés pertinent pour le profil de l'utilisateur considéré, dans leur ordre d'importance aussi bien dans la requête que dans le centre d'intérêt. Ce résultat démontrent l'importance de l'intégration du profil utilisateur dans la RI.

Conclusion et perspectives

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la recherche d'information personnalisée. Il consiste à intégrer le profil utilisateur dans le processus de recherche d'information. Nous avons proposé une approche basée sur les modèles de langage.

Les modèles de langage sont basés sur des collections de documents. Le but est de modéliser la pertinence d'un document vis-à-vis d'une requête comme la probabilité que la requête puisse être générée à partir du modèle de langage de ce document. Leurs utilisation s'est avérée intéressante pour mesurer la pertinence d'un document face à une requête en proposant deux modèles: un modèle de document et un modèle de requête. Ces modèles sont souvent estimés en utilisant la fréquence des termes (mots simples) dans les documents ou dans la collection.

Nous avons proposé un modèle de langue en intégrant le profil utilisateur dans la phase d'appariement document requête. L'objectif visé est l'amélioration des résultats retournés en fonction du profil de l'utilisateur.

Nous avons testé notre modèle sur un exemple de collection composé de huit documents. Nous avons construit nos centres d'intérêts par des requêtes d'apprentissage. Ensuite, nous avons utilisé des requêtes de test pour évaluer le modèle proposé. Les résultats obtenus ont montré une amélioration de la pertinence par rapport au modèle de langue de base.

Pour pouvoir conclure d'une façon concrète sur l'efficacité de notre modèle, nous envisageons de l'implémenter sur une plateforme de test.

Ces résultats ouvrent des perspectives intéressantes comme:

1. La prise en compte de l'évolution profil utilisateur dans notre modèle.
2. Approfondir l'approche de la construction du profil de l'utilisateur.

BIBLIOGRAPHIE

Bibliographie

[Abdou et al, 06] : Abdouroihamane Anli, Mourad Abed PerSyst : Un Système de Personnalisation de l'information transport multimodale. Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines UVHC, Le Mont Houy, F-59300 Valenciennes Cedex 9, France, 2006.

[Amato, 99]: G. Amato, U. Straccia, User Profile Modeling and Applications to Digital Libraries, Proc. 3rd European Conf. Research and Advanced Technology for Digital Libraries, ECDL, 1999.

[AMI 08] : F. Amirouche, « Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets », Thèse de doctorat, soutenue à l'IRIT, l'université de Toulouse III, 2008.

[BAZ 05]: M. Baziz , « Indexation conceptuelle guidée par ontologie pour la recherche d'information », Thèse de doctorat , soutenue à l' IRIT, 2005.

[Beklin, 92]: N. Belkin and W. Croft. Information filtering and information retrieval: Two sides of the same coin? Communication of the ACM, 1992.

[*Berger and Lafferty, 1999] :* *Berger and Lafferty*. Information retrieval as statistical translation, 1999.

[Billsus et Pazzani, 1999] : Billsus D., and Pazzani M., «A hybrid user model for news stories classification », In Proceedings of the seventh International Conference on User Modelins, Banff, Canada, pp. 99-108, , 1999.

[BK, 05]: M. Bouzeghoub and D. Kostadinov. Personnalisation de l'information : Aperçu de l'état de l'art et définition d'un modèle flexible de définition de profils. In Actes de la seconde édition de la Conférence en Recherche d'Information et Applications(CORIA), pages 201.218, Grenoble, France, 2005.

[Bottraud, 04]J.C. Bottraud , G. Bisson , M.F. Bruandet, apprentissage de profils pour un agent de recherche d'information. Coria'04, IRIT Toulouse France, 2004.

[Boughanem and Kraaij]: *Boughanem and Kraaij.* Modèles de langue pour la recherche d'information.

[Bouzeghoub] : M.Bouzeghoub, «Action spécifique sur la personnalisation de l'information », CNRS-AS98/RTP9, laboratoire PRISM, Université de Versailles.

[Cao, G., Nie, J.-Y., Bai, J., 2005]: Cao, G., Nie, J.-Y., Bai, J., 2005, Integrating word relationships into language models, SIGIR'05 : Proceedings of 28th ACM International conference on research and development in information retrieval, pages 298-305, New-York : ACM, 2005.

[Chen et al, 2002] : Chen C., Chen M., and Sun Y., «A self-adaptive personal view agent», Journal of Intelligent Information Systems, pp.173±194, Mars 2002.

[Chengxiang Zhai.2001] : *Chengxiang Zhai.* Risk minimization and language modelling in text retrieval, 2001.

[Chirita, P.A., Paiu, R., Nejdl, W., Kohlschütter, C.]: Chirita, P.A., Paiu, R., Nejdl, W., Kohlschütter, C., Using ODP metadata to personalize search, SIGIR'05 : Proceedings of 28th ACM International conference on research and development in information retrieval ,

pages 178-185, New-York : ACM, 2005.
[Christian Jauvin.2003] : *Christian Jauvin* .Quelques modèles de langage statistiques et graphiques lissés avec Word Net, 2003.
[Dahak, 06]: F. DAHAK, « Indexation des documents semi-structurés: Proposition d'une approche basée sur le fichier inversé et le Tree. » Mémoire de Magister, 2006
[Denos, 97] : Denos N., Modélisation de la pertinence en recherche d'information - modèle conceptuel, formalisation et application, thèse d'université, Grenoble, octobre 1997.
[Gauch, 03]: S. Gauch, J. Chaffe, A. Pretschner, Ontology-Based User Profiles for Search and Browsing, To appear in J. User Modeling and User-Adapted Interaction, the Journal of Personalization Research , Special Issue on User Modeling for Web and Hypermedia Information Retrieval, 2003.
[Hlaoua, 07] : L.Hlaoua, « reformulation de requêtes par réinjection de pertinence dans les documents semi-structurés », Toulouse 2007.
[Huhn ,99]= [Luhn 99] : M. N. Huhn, L.M. Stephens, Personal Ontologies. Internet Computing, Vol. 3, No. 5, pp. October 1999.
[**Ingwersen, 92] :** Ingwersen P, *Information retrieval interaction*, Taylor-Graham Publishing, 1992 Herman, N, *The creative brain*. Brain Books, 1988.
[Jam, 95]: Jameson, A., Numerical uncertainty management in user and student modeling: an overview of systems and issues, user modeling and user adapted interaction 5(3-4), 193:251, 1995.
[Janowski et al, 01] : W. Janowski, A. Sarner. Five Opportunities for Personalization. Gartner Group, pp. 1, 05/2001.
[Jones, 1972] : Jones, K. S. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation, 28(1), 11-21. 1972
[Kelly, D., Teevan, J., 2003]: Kelly, D., Teevan, J., Implicit fee dback for inferring user preference: A bibliography, SIGIR Forum, vol 37, pages 18-28, New-York : ACM, 2003.
[Kim, H.-R., Chan, P.K.], Personalized ranking of search results with learned user interest hierarchies from bookmarks, WEBKDD'05 Workshop at the 11th ACM International conference on Knowledge discovery and data mining, pages 32-43, New-York : ACM, 2005.
[Kob.01]: A. Kobsa, Genericuser modeling systems, User modeling anduser adapted intraction, 11 49 :63, Kluwer Academic Publishers. 2001.
[Kostadinov, 03] : D. Kostadinov. Personnalisation de l'information et gestion des profils utilisateurs, Rapport de DEA, Université de Versailles, France, 2003.
[Kurki, 99]: T. Kurki, S. Jokela, R. Sulonen, M. Tirpeinen, Agents in delivering personalized content based on semantic metadata, In Proc, AAAI Sping Symposium, 1999.
[Lafferty et Zhai] *Lafferty and Zhai.* A study of smoothing methods for language models applied to ad Hoc information retrieval.
[Liu and Croft and B.W, 2001]: Liu, X., Croft, B.W., Cluster-based retrieval using language models, SIGIR'01, Proceedings of 24th ACM International conference on research

and development in information retrieval, pages 186-193, New-York : ACM, 2001.
[Moukas, 1997] : Moukas A., « Information discovery and filtering using a multi-agent evolving ecosystem », Applied Artificial Intelligence, pp. 437±457, 1997.
[Per, 88] : Peral J., Probabilistic reasoning in intelligent systems, San Mateo, California: Morgan Kaufmann Publishers.
[PG, 1999]: A. Pretshner, S. Gauch, Personalization on the web, Technical report ITTC-FY2000-TR-13591601? Information and telecommunication technology center, Department of electrical engineering and computer science, University of Kansas, December 1999.
[*Philippe Langlais*, 2003] : *Philippe Langlais*, Modèle de langue, 2003.
[Ponte et Croft, 98]*: Ponte and Croft.* A language modeling approach to information retrieval, 1998.
[Ponte, 1998] : Ponte, J.M., Croft, W. B.language modeling approach to information retrieval. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 275-281, 1998.
[Pretschner, 99]: Alexander Pretschner, Susan Gauch. Ontology Based Personalized Search. In Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), November 1999.
[Richard Beaufort] : *Richard Beaufort, Thierry Dutoit, Vincent Pagel.* Analyse syntaxique du français.
[Riw, 94]: Resnick, P., Iacovou, N. And Ward, B., An open architecture for collaborative filtering of netnews, In CSCW'94 Proceedings of the conference on computer Supported Collaborative Work, 175:186, 1994.
[Robertson et al. 1997] Walker M., Robertson S., Sparck Jones, and al., « Okapi at trec-3. In Second Text Retrieval Conf (TREC-3) » , 1995.
[Ronald Rosenfeld.] : *Ronald Rosenfeld.* Two decades of statistical language modelling.
[Salton et Al, 1983]: G.Salton, E.Fox, and H.Wu. « Extended boolean information retrieval ». Communications of the ACM, 26(11): 1022-1036, 1983.
[Salton, 1971]: Salton .G. A comparison between manual and automatic indexing methods. Journal of American Documentation, 20(1): pages 61–71, 1971.
[Saracevic, 70] : Tefko Saracevic, "Introduction to Information Science", 111-151. New York: R.R. Bowker, 1970. Chap. 3: The concept of "relevance" in information science: A historical review.
[Saracevic, T, 1999]: Saracevic, T, 1999, Information science, Journal of the American society for information science, vol 50, pages 1051-1063, Mississauga: Wiley, 1999.
[SKW, 2000]: Scime, A., Kershberg, L., WebSifter: An Ontology-based personalizable search agent for the Web, In Proceedings of International Conference on Digital Librairies: Research and Practice (2000).
[Song et Croft] : *Song and Croft.* A general language model for information retrieval.
[Tam ,Bough , 05] : L.Tamine, M.Boughanem « accès personnalisé a l'information,

Approches et techniques», rapport interne, Institut de recherche en informatique de Toulouse, janvier 2005.
[Tamine, 98] : L.Tamine « les systèmes de recherche d'information : reformulation de requête et apprentissage bases sur les algorithmes génétiques », thèse de magister, université Mouloud MAMMERI, Tizi_ouzou ,1998.
[Tebri Hamid, 2004] : *TEBRI Hamid*. Formalisation et spécification d'un système de filtrage incremental d'information, December, 2004.
[Teevan, J., Dumais, S.T., Horvitz, E.]: Teevan, J., Dumais, S.T., Horvitz, E Personalizing search via automated analysis of interests and activities, SIGIR'05: Proceedings of 28th ACM.
[Victor Lavrenko and Crof W]: *Victor Lavrenko and Crof W. Bruce t*. Relevance Based Language Models.
[Victor Lavrenko, 2000]: *Victor Lavrenko*. Localized smoothing for multinomial language models, 2000.
[Yannick Estève, 2002] : *Yannick Estève*. Thèse sur l'intégration de sources de connaissances pour la modélisation stochastique du langage, appliquée à la parole continue dans un contexte de dialogue oral Homme-Machine, 2002.
[Zemirli, 03/04] : Zemirli W.N,Vers le développement d'un système de recherche d'information personnalisé intégrant le profil utilisateur, Université Paul Sabatier Toulouse III, Equipe SIG/RI ,2003/2004.
[Zemirli, 08] : Zemirli W.N, « Modèle d'accès personnalisé a l'information basé sur les diagrammes d'influence intégrant un profil multidimensionnel », Thèse de doctorat, Université Paul Sabatier, Toulouse, France, juin 2008.
[Zipf, 1949] : George.K. Zipf. Human Behavior and the Principle of Least Effort Addison-Wesley, Reading MA (USA). 1949.
[ZTB, 05] W. N. Zemirli, L. Tamine, and M. Boughanem. Accès personnalisé à l'information : vers la définition d'un profil utilisateur multidimensionnel. In International Symposium On Programming Systems, pages 20.28. USTHB, 2005.
[ZUC, 01]: I. Zuckerman, D.W. Albercht, Predictive statistical models for user modeling and user adapted interaction, 11 5:18, 2001.