**République Algérienne Démocratique et Populaire**
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**
**Université Mouloud Mammeri de Tizi-Ouzou**
**Faculté de Génie Electrique et de l'Informatique**

# Thèse de Doctorat en Informatique

## Thème

# *« Recherche d'Information : un modèle de langue combinant mots simples et mots composés »*

Présenté par **M. Hammache Arezki**

**Devant le jury composé de :**

**M. M. SI-MOHAMMED** : Professeur à l'UMM de Tizi-Ouzou (Président)
**M. M. BOUGHANEM** : Professeur à l'Université de Toulouse (Directeur de thèse)
**M. R. AHMED-OUAMER**: Maître de Conférences à l'UMM de Tizi-Ouzou (Co-directeur de thèse)
**Mme. H. DRIAS** : Professeur à L'USTHB d'Alger (Examinatrice)
**M. M. MEZGHICHE** : Professeur à l'Université de Boumerdès (Examinateur)
**M. I. RASSOUL** : Maître de Conférences à l'UMM de Tizi-Ouzou (Examinateur)

*À la mémoire de ma mère*

# Remerciements

Je tiens à exprimer ma profonde gratitude à Monsieur Mohand Boughanem, Professeur à l'Université Paul Sabatier de Toulouse, pour avoir encadré et dirigé mes recherches, malgré l'éloignement. Je le remercie pour la confiance qu'il a bien voulu m'accorder, ses conseils et remarques constructives qui m'ont permis d'améliorer la qualité de mes travaux. Qu'il soit ici assuré de mon très grand respect et de ma profonde reconnaissance.

Je tiens à remercier Monsieur Rachid Ahmed-Ouamer, Maître de Conférences à l'Université Mouloud Mammeri de Tizi-Ouzou, pour avoir co-encadré ce travail.

J'adresse mes plus sincères remerciements À Monsieur Malik Si-Mohammed, Professeur à l'Université Mouloud Mammeri de Tizi-Ouzou, Madame Drias Habiba, Professeur à L'Université des Sciences et Technologies Houari Boumediene d'Alger , Monsieur Mohamed Mezghiche, Professeur à l'Université M'Hamed Bouguera de Boumerdès et Monsieur Rassoul Idir, Maître de Conférences à l'Université Mouloud Mammeri de Tizi-Ouzou, pour l'intérêt qu'ils ont porté à mes travaux en examinant ce mémoire et pour l'honneur qu'ils me font en participant à ce jury

Mes remerciements vont également à tous mes collègues et amis du département d'informatique de l'université de Mouloud Mammeri de Tizi-Ouzou.

Je souhaiterais adresser des remerciements plus particuliers à toute ma famille : tout d'abord je souhaite dédier cette thèse à la mémoire de ma mère, qui a tant fait pour moi et sans qui ce travail de thèse n'aurais peut être pas vu le jour. Je dédie aussi ce travail de thèse à mon père qui à toujours cru en moi et pour son soutien permanent tout au long de mes années d'études. Je remercie tout mes frères et sœurs qui de par leur soutien au quotidien ont contribués à la réalisation de ce travail.

# Table des Matières

# Liste des tableaux

# Table des figures

# INTRODUCTION GENERALE

## Contexte et Problématiques

Aujourd'hui, l'information joue un rôle primordial dans le quotidien des individus et dans l'essor des entreprises. Cependant, le développement de l'Internet et la généralisation de l'informatique dans tous les domaines ont conduit à la production d'un volume d'information sans précédent. En effet, la quantité d'information disponible, particulièrement à travers le web, se mesure en milliards de pages. Il est par conséquent, de plus en plus difficile de localiser précisément ce que l'on recherche dans cette masse d'information. La recherche d'information (RI) est le domaine par excellence qui s'intéresse à répondre à ce type d'attente. En effet, l'objectif principal de la RI est de fournir des modèles, des techniques et des outils pour stocker et organiser des masses d'informations et localiser celles qui seraient pertinentes relativement à un besoin en information d'un utilisateur, souvent, exprimé à travers une requête. Ces outils sont appelés des Systèmes de Recherche d'Information (SRI). De manière générale, le fonctionnement d'un SRI consiste à construire une représentation des documents et de la requête et d'établir une comparaison entre ces deux représentations (requête, documents) pour retourner les documents pertinents. Cette comparaison est réalisée au moyen d'un modèle de recherche. Afin d'obtenir un SRI performant, il est nécessaire de construire une bonne représentation du document et de la requête et de développer un modèle de RI qui supporte ces représentations.

La plupart des SRI existants représentent les documents comme un ensemble de mots clés, ce que l'on appelle communément une représentation par sac de mots. Ces mots clés sont généralement pondérés en utilisant des schémas de pondération tels que $tf \times idf$ **[183]**, *BM25* **[162]** ou le modèle de langue uni-gramme **[151]** qui prennent en compte les statistiques suivantes : la fréquence du terme dans le document (*tf*), sa fréquence dans la collection (*idf*) et la taille du document. Tous ces modèles supposent que les mots clés sont indépendants. Cette hypothèse d'indépendance entre termes facilite grandement les calculs. De ce fait, l'ordre des termes dans une phrase est donc ignoré. Ceci peut de toute évidence conduire à l'ambigüité entre termes, qui pourraient engendrer des résultats bruités (contenant beaucoup de documents non pertinents). A titre d'exemple, prenons la requête « recherche d'information », avec le processus de sac de mots, un document comportant dans une partie le mot « recherche » et dans une autre partie le rôle de « l'information dans le développement d'une entreprise », serait présenté à l'utilisateur comme pertinent. Or, on voit bien que ce document n'est pas pertinent car il ne traite pas de la « recherche d'information ».

Il est par conséquent nécessaire de développer des modèles de recherche d'information allant au-delà de la représentation avec une liste de mots clés. Parmi les pistes les plus investies on trouve celles prenant en compte la proximité entre termes et l'utilisation d'unités de représentation plus complexes (les expressions, les mots composés). Par exemple, les documents contenant le mot composé « recherche d'information» devraient être avantagés lors du classement des documents dans la requête précédente.
Il est évident que l'utilisation de telles techniques conduit à une représentation plus précise et plus réaliste du contenu des documents. C'est dans ce cadre que se situe l'une de nos contributions.

Dans le contexte du web, d'autres sources d'information autre que le contenu textuel du document peuvent être exploitées, en particulier la structure des liens, pour améliorer les performances de la recherche d'information. La plupart des méthodes proposées dans la littérature intègrent cette dimension en se basant sur le postulat suivant: «une page web référencée par beaucoup d'autre pages est a priori pertinente ». Cependant, ces méthodes ne prennent pas en compte le fait que le web est structuré sous forme de sites web. Notre seconde contribution consiste à prendre en compte cette réalité.

L'autre challenge des SRI est la difficulté de la représentation du besoin en information de l'utilisateur dû à la nature des requêtes exprimées par celui-ci, particulièrement sur le web. Le plus souvent, les utilisateurs expriment leur besoin en information avec des requêtes courtes, qui contiennent peu de mots, en moyenne 2.35 mots. Avec ce type de requêtes, il est difficile de décrire de manière complète et précise le besoin en information. De plus, les termes de ces requêtes peuvent avoir plusieurs sens, ce qui rend ces requêtes ambiguës. Le terme « virus » dans une requête peut avoir au moins deux sens. Enfin, les termes utilisés dans ces requêtes ne correspondent pas toujours aux termes utilisés par les auteurs des documents pour décrire le même concept. Ce problème est nommé problème de disparité de termes. Par exemple, un utilisateur intéressé par l'achat d'une voiture ; formule la requête « achat voiture». Cependant, des documents pertinents peuvent utiliser « vente automobile» pour exprimer le même concept. L'expansion de requêtes est l'une des techniques utilisée pour résoudre ce problème. Elle consiste à étendre la requête originale avec des termes liés aux termes de celle-ci. Par exemple, dans la requête précédente le terme « automobile » est ajouté à la requête initiale car il est lié au terme « voiture » et le terme « vente » est ajouté car il est lié au terme « achat ». Dans le cadre de ce mémoire nous présenterons notre troisième contribution qui est une nouvelle technique d'expansion de la requête.

Comme nous l'avions signalé précédemment, l'établissement de la correspondance entre la représentation de la requête et la représentation des documents nécessite l'usage d'un modèle de RI. Plusieurs modèles de RI ont été développés, on y trouve le modèle booléen, le modèle vectoriel, le modèle probabiliste de base et le modèle de langue. Ce dernier a acquis une grande popularité, en raison de sa simplicité, efficacité, performance et son fondement théorique solide, basé sur la théorie des probabilités. De plus, il offre la possibilité de combiner différentes informations (évidences) sur un document pouvant être liées au contenu textuel du document ou non liées telles que la structure du document, la structure des liens, etc. Toutes ces raisons motivent l'utilisation du modèle de langue dans notre travail.

## Contributions

Les travaux présentés dans ce mémoire se situent dans le contexte de la recherche d'information. Nous nous intéressons plus particulièrement à :

(1) la définition d'un modèle de langue permettant de mieux prendre en compte les problèmes d'ambigüité et de disparité de termes posés par la représentation de type: sac de mots clés ;

(2) la prise en compte de la structure du web et de la structure des hyperliens afin d'estimer la pertinence a priori d'un document web.

Nous proposons pour notre part :

**(1) Un modèle de langue mixte combinant les mots simples et mots composés**

La plupart des modèles de langue développés sont des modèles uni-grammes (sac de mots clés), qui sont caractérisés par la non prise en compte des relations entre les termes. Ce qui conduit aux problèmes d'ambigüité et de disparité de termes. Pour pallier ces problèmes nous proposons un modèle de langue combinant les mots simples et mots composés.

**(a) Réduction de l'ambiguïté des termes**

Afin de réduire l'ambiguïté de termes dans la requête et les documents, nous proposons l'utilisation des mots composés comme unités de représentation. Ceci est dicté par le fait que les mots composés sont moins ambigus et plus précis que les mots simples. Par exemple, le terme « *virus* » est ambigu, par contre les mots composés «virus *informatique* » et « *virus de paludisme»* sont non ambigus, et le mot composé *« pollution de l'air »* et plus spécifique que les termes *« pollution »* et *« air »* pris séparément. Cependant, l'apport des mots composés comme unités de représentation des requêtes et des documents est tributaire de plusieurs paramètres, entre autres, la procédure de sélection de mots

composés et le schéma de pondération utilisé. À cet effet nous proposons une approche caractérisée par les points suivants :

- Une méthode de sélection de mots composés basée sur un filtrage des bi-grammes.
- Un nouveau schéma de pondération des mots composés basée sur la notion de dominance entre termes.
- La définition d'un modèle de langue permettant une meilleure prise en compte des mots composés et des mots simples.

**(b) Résolution du problème de disparité de termes avec l'expansion de la requête**

Il existe deux techniques pour pallier le problème de disparité de termes : l'expansion de requêtes et l'expansion de document. Cependant, l'expansion du document réalisée à l'étape recherche a un handicap majeur qui est le temps de réponse qui pourrait être trop long pour être accepté par les utilisateurs. Par contre, l'expansion de la requête est moins gourmande en termes de temps de traitement car la requête est généralement très courte. Une autre contrainte relative à l'expansion de requêtes, plus particulièrement à la technique de réinjection de pertinence (relevance feedback), concerne la sollicitation de l'utilisateur pour juger quelques documents que le système lui propose. Les utilisateurs n'acceptent pas toujours d'être sollicités pour effectuer ce type de tâche. De ce fait, la pseudo-réinjection de pertinence est l'une des méthodes permettant de pallier cette contrainte. Elle consiste à supposer que les premiers documents renvoyés par le système sont pertinents, sans les juger. Ces documents sont ensuite utilisés pour extraire les termes d'expansion. Cependant, la plupart des méthodes de pseudo réinjection de pertinence considèrent la requête comme un ensemble de termes indépendants. Ce qui amplifie le problème d'ambigüité de la requête. Pour y remédier, nous avons proposé une extension du modèle précédent pour prendre en compte l'expansion de la requête par réinjection de pertinence. L'approche de réinjection de pertinence que nous proposons est caractérisée par les points suivants :

- Une requête est représentée comme un ensemble de mots simples et de mots composés.
- Le choix des termes d'expansion est basé sur la relation de cooccurrence entre les termes de la requête initiale et les termes des premiers documents retournés en réponse à la requête initiale. Les termes qui ont une plus grande probabilité de cooccurrence avec l'ensemble des termes de la requête initiale sont alors choisis pour étendre la requête initiale.

- Les termes d'expansion choisis peuvent être des mots simples ou des mots composés, ce qui n'est pas le cas avec les méthodes antérieures de réinjection de pertinence.

**(2) La prise en compte de la structure du web pour l'estimation de la pertinence d'un document web**

Comme nous l'avons signalé précédemment, la plupart des techniques proposées pour intégrer la structure des liens ou autres caractéristiques sur la page web (longueur, facteur temps) dans l'estimation de la pertinence d'une page web, considèrent la page web indépendante de son site web. L'idée que nous explorons dans cette contribution consiste à utiliser les caractéristiques du site contenant la page concernée pour conditionner la probabilité de pertinence de la page. Une fois cette probabilité calculée nous la combinons avec le score obtenu par le contenu de la page web. Cette combinaison des deux évidences est réalisée dans le cadre du modèle de langue.

## Organisation du mémoire

Nous structurons ce présent mémoire en deux parties ; la première traite de l'état de l'art et elle est constituée de trois chapitres, et la seconde partie, comporte trois chapitres, et concerne les contributions apportées.

La première partie dénommée « Etat de l'art » comprend les chapitres 1,2 et 3.
Dans le chapitre 1 nous décrivons trois points essentiels. Tout d'abord, nous donnons les concepts de base de la recherche d'information. On y trouve les notions de besoin en information, de requête, de document et de pertinence et le processus d'indexation. Nous décrivons aussi les différents modèles de la recherche d'information en particulier le modèle booléen, le modèle vectoriel et le modèle probabiliste. Ensuite, nous introduisons les différentes approches proposées pour remédier aux problèmes d'ambigüité et de disparité des termes dus à l'utilisation des mots clés indépendants comme unités de représentation. Le troisième point traité dans ce chapitre concerne l'évaluation des systèmes de recherche d'information.

Le chapitre 2 est consacré à la recherche d'information sur le web. Nous traitons trois points, à savoir, les éléments distinctifs entre la RI classique et la RI sur le web, les sources d'information spécifiques aux documents web et les différentes méthodes ou modèles développés pour combiner ces différentes sources d'information dans le but d'améliorer la pertinence de la recherche d'information.

Le chapitre 3 est dédié au modèle de langue. Ce modèle est caractérisé par son solide fondement mathématique (probabilités statistiques) et sa capacité à combiner différentes sources d'information sur un document. Ces deux caractéristiques conviennent bien à la recherche d'information sur le web. Nous présentons précisément les points suivants dans ce chapitre. Nous définissons tout d'abord l'idée de base des modèles de langue et leur utilisation en RI. Nous décrivons ensuite les différents modèles de langue proposés pour intégrer les relations entre termes afin de solutionner les problèmes d'ambigüité et de disparité entre termes. Enfin, nous présentons les différents travaux intégrant les informations a priori de pertinence d'un document dans le cadre du modèle de langue.

La deuxième partie intitulée « Contributions » est composée des chapitres 4, 5 et 6.
Dans le chapitre 4, nous présentons notre première contribution qui consiste à établir une bonne représentation (modèle) du document (requête) par l'utilisation d'unités plus précises et plus expressives. Plus explicitement, nous présentons un modèle de langue mixte combinant les mots simples et les mots composés. L'évaluation de ce modèle et sa comparaison par rapport aux modèles de l'état de l'art, sur trois collections de TREC est également présentée.

Dans le chapitre 5, nous présentons notre seconde contribution, qui consiste en une extension du modèle précédent pour prendre en compte l'expansion de la requête. Cette expansion est réalisée par réinjection de pertinence. Pour déterminer les termes d'expansion on additionne les poids des relations d'un terme candidat avec chacun des termes de la requête (simple, composé). Un terme candidat est choisi s'il est fortement en relation avec la plupart des termes de la requête. Le modèle ainsi décrit est expérimenté et évalué sur deux collections TREC. Les résultats de ces expérimentations sont présentés et discutés dans ce chapitre.

Dans le chapitre 6 nous présentons notre troisième contribution consistant à intégrer les caractéristiques de site web dans l'estimation de la probabilité a priori de pertinence d'une page web. Après une description du modèle, nous présentons les résultats des expérimentations réalisées sur une collection TREC (*.GOV*). Une comparaison de ce modèle avec les modèles de l'état de l'art est également présentée.

Nous concluons notre mémoire par une conclusion générale, où nous présentons les perspectives de nos propositions.

# Partie I

# Etat de l’art

# Chapitre 1

# Recherche d'information classique

## 1.1 Introduction

La Recherche d'Information (RI) n'est pas un domaine récent, il date des années 40. Une des premières définitions de la RI a été donnée par Salton : « la recherche d'information est un domaine qui a pour objectif, la représentation, l'analyse, l'organisation, le stockage et l'accès à l'information » **[168]**.

Plusieurs tâches se regroupent sous le vocable de la RI, la plus ancienne est la recherche documentaire, on y trouve également d'autres tâches plus au moins récentes comme : le filtrage d'information, l'extraction d'information, la recherche d'information multilingue, les questions réponses, la recherche d'information sur le web, etc.

Ce chapitre a pour but de présenter le domaine de la RI. Dans la première partie, nous présentons les concepts de base de la RI. En particulier, nous décrivons les notions de document, de requête et de pertinence ; les processus d'indexation, de recherche et de reformulation de requêtes ; ainsi que, les modèles de RI. Dans la seconde partie, nous passerons en revue les techniques de recherche d'information sémantique. Dans la dernière partie de ce chapitre est discutée l'évaluation des systèmes de recherche d'information.

## 1.2 Concepts de base de la recherche d'information

Le rôle d'un Système de Recherche d'Information (SRI) est de mettre en œuvre des techniques et des moyens permettant de retourner les documents pertinents d'une collection en réponse à un besoin en information d'un utilisateur, exprimée par un langage de requêtes qui peut être le langage naturel, une liste de mots clés ou un langage booléen **[10]**.

Afin d'atteindre cet objectif, un processus d'indexation des documents de la collection est effectué. Il permet de construire une représentation synthétique des documents, appelée index. Lorsque l'utilisateur formule sa requête un processus similaire est effectué sur la requête. Il consiste à analyser la requête et établir une représentation interne. Puis, le système établit une correspondance entre la représentation de la requête et la représentation des documents (index) pour sélectionner et présenter les documents qui répondent le mieux au besoin de l'utilisateur (les documents pertinents).

Le SRI s'appuie sur des modèles de RI pour établir cette correspondance entre les documents et la requête.

L'architecture générale d'un SRI illustrée par la figure 1.1 fait ressortir des éléments constitutifs tels que : le document, le besoin en information, la requête et la pertinence, ainsi que trois principales fonctionnalités : l'indexation, la recherche et la reformulation de la requête. Dans ce qui suit, nous détaillons ces éléments et ces processus.

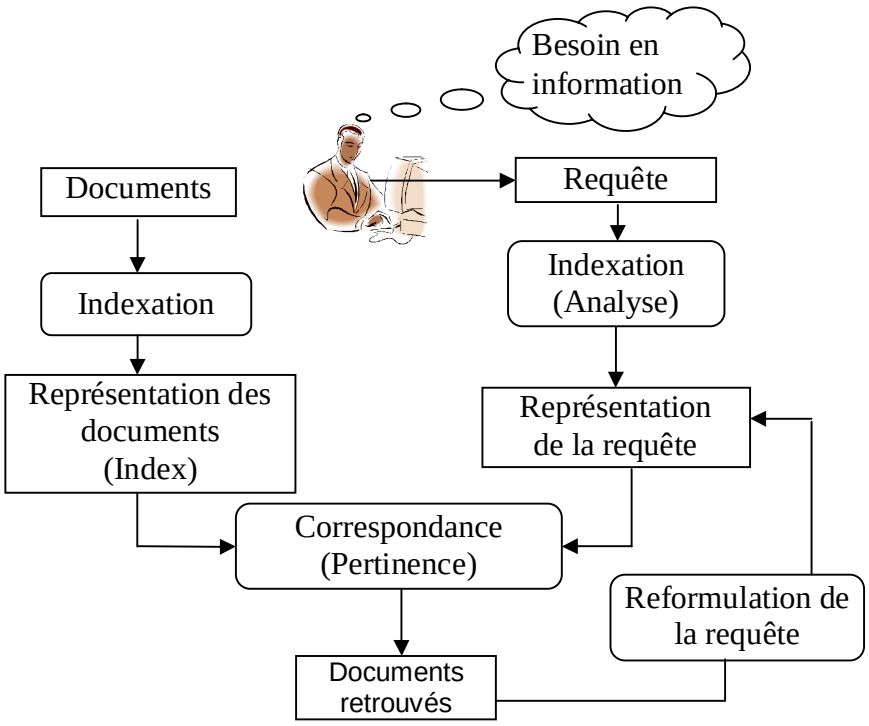



**Figure 1.1 Architecture générale d'un Système de Recherche d'Information**

**Document et collection de documents**

Un document est un élément essentiel dans un SRI. Dans son acceptation courante, l'une des définitions possible du terme document est de le considérer comme un support physique de l'information, qui peut être du texte, une page web, une image, une séquence vidéo, etc.

Dans le cas d'un document texte on peut le représenter selon trois vues **[72] [167]** :

***La vue sémantique (ou contenu) :*** elle se concentre sur l'information véhiculée dans le document.

***La vue logique :*** elle définit la structure logique du document (structuration en chapitres, sections)

***La vue présentation :*** elle consiste en la présentation sur un médium à deux dimensions (alignement de paragraphes, indentation, en-têtes et pieds de pages, etc.).

L'ensemble des documents manipulés par un SRI se nomme collection de documents (ou base documentaire ou encore corpus).

**Besoin en information et requête**

La requête est une expression approximative du besoin en information de l'utilisateur. Ce dernier est une expression mentale des informations que l'utilisateur recherche.

Les requêtes soumises au SRI par les utilisateurs peuvent ne pas refléter leurs besoins en information. Cela est dû, d'une part, au fait que l'utilisateur ignore le fonctionnement interne du SRI, et il n'a qu'une vision restreinte des documents disponibles dans la collection. D'autre part, le SRI n'a souvent aucune connaissance a priori de ses utilisateurs (centres d'intérêts, niveaux, parcours, etc.). Ce biais entre la requête et le besoin en information est une des difficultés majeures de tout système de recherche d'information. Afin de remédier partiellement à ce problème un mécanisme de reformulation de requêtes peut être intégré dans les SRI.

**Pertinence**

La pertinence est une notion fondamentale et cruciale dans le domaine de la RI. Cependant, la définition de cette notion complexe n'est pas simple, car elle fait intervenir plusieurs notions **[21] [142].** Basiquement, elle peut être définie comme la correspondance entre un document et une requête ou encore comme une mesure d'informativité du document à la requête. Essentiellement, deux types de pertinence sont définis : la pertinence système et la pertinence utilisateur.

***La pertinence Système*** **[42]** est souvent présentée par un score attribué par le SRI afin dévaluer l'adéquation du contenu des documents vis-à-vis de celui de la requête. Ce type de pertinence est objectif et déterministe.

***Pertinence utilisateur*** **[89] [142] [172]** quant à elle, se traduit par les jugements de pertinence de l'utilisateur sur les documents fournis par le SRI en réponse à une requête. La pertinence utilisateur est subjective, car pour un même document retourné en réponse à une même requête, il peut être jugé différemment par deux utilisateurs distincts (qui ont des centres d'intérêt différents). De plus, cette pertinence est évolutive, un document jugé non pertinent à l'instant *t* pour une requête peut être jugé pertinent à l'instant *t+1*, car la connaissance de l'utilisateur sur le sujet a évolué.

### 1.2.1 Le processus d'indexation

Pour que la recherche d'information se réalise avec des coûts acceptables, il convient d'effectuer une opération fondamentale sur les documents de la collection. Cette opération est nommée indexation **[10] [132]**. Elle consiste à associer à chaque document une liste de mots-clés appelée aussi descripteur, susceptible de représenter au mieux le contenu sémantique des documents.

La finalité de l'indexation est donc de produire une représentation synthétique des documents, formé de termes, ces termes peuvent être extraits de trois manières :

***Manuelle*** : chaque document de la collection est analysé par un spécialiste du domaine ou un documentaliste. L'indexation manuelle assure une meilleure précision dans les documents restitués par le SRI en réponse aux requêtes des utilisateurs **[155]**. Néanmoins, cette indexation présente un certain nombre d'inconvénients liés notamment à l'effort et le prix qu'elle exige (en temps et en nombres de personnes). De plus, cette indexation est subjective, qui est liée au facteur humain, différents spécialistes peuvent indexer un document avec des termes différents. Il se peut même arriver qu'un spécialiste indexe différemment un document, à différents moments.

***Semi-automatique :*** la tâche d'indexation est réalisée ici conjointement par un programme informatique et un spécialiste du domaine **[98]**. Le choix final des descripteurs revient à l'indexeur humain. Dans ce type d'indexation un langage d'indexation contrôlé est généralement utilisé.

***Automatique :*** dans ce cas, l'indexation est entièrement automatisée. Elle est réalisée par un programme informatique et elle passe par un ensemble d'étapes pour créer d'une façon automatique l'index. Ces étapes sont : l'analyse lexicale, l'élimination des mots vides, la normalisation (lemmatisation ou radicalisation), la sélection des descripteurs, le calcul de statistiques sur les descripteurs et les documents (fréquence d'apparition d'un descripteur dans un document et dans la collection, la taille de chaque document, etc.) et enfin la création de l'index et éventuellement sa compression. Nous détaillons ces différentes étapes ci-dessous.

#### 1.2.1.1 L'analyse lexicale

Elle permet de convertir un texte de document en une liste de termes. Un terme est un groupe de caractères constituant un mot significatif **[69]**. L'analyse lexicale permet de reconnaître les espaces de séparation des mots, les chiffres, les ponctuations, etc.

#### 1.2.1.2 L'élimination des mots vides

Les mots vides (article, proposition, conjonction, etc.) sont des mots non significatifs dans un document, car ils ne traitent pas le sujet du document.

On distingue deux techniques pour éliminer les mots vides :

- L'utilisation d'une liste préétablie de mots vides (aussi appelée *anti-dictionnaire* ou *stop-list*),
- L'élimination des mots ayant une fréquence qui dépasse un certain seuil dans la collection.

L'élimination des mots vides réduit la taille de l'index, ce qui améliore le temps de réponse du système. Cependant, elle peut réduire le taux de rappel (définit dans la section 1.4.2), en réponse à des requêtes bien spécifiques (par exemple, la requête *be or not to be*).

#### 1.2.1.3 La normalisation

La normalisation consiste à représenter les différentes variantes d'un terme par un format unique appelé lemme ou racine. Ce qui a pour effet de réduire la taille de l'index. Plusieurs stratégies de normalisation sont utilisées : la table de correspondance, l'élimination des affixes (l'algorithme de Porter **[152]**), la troncature, l'utilisation des N-grammes **[1]**.

L'inconvénient majeur de cette opération est qu'elle supprime dans certains cas la sémantique des termes originaux, c'est le cas par exemple des termes derivate/derive, activate/active, normalisés par l'algorithme de Porter.

#### 1.2.1.4 Le choix des descripteurs

Elle consiste à déterminer le type d'unités élémentaires pour représenter les documents. On parle aussi de descripteur. L'objectif est d'avoir une représentation des documents permettant une moindre perte d'information sémantique possible. On distingue plusieurs types de descripteurs [**15**].

- **Les mots simples** : les mots simples du texte de document en éliminant les mots vides,
- **Les lemmes** ou les racines des mots extraits.
- **Les N-grammes** : qui sont une représentation originale d'un texte en séquence de N caractères consécutifs. On trouve des utilisations de bi-grammes et trigrammes dans la recherche d'information.
- **Les mots composés :** groupes de mots ou expression (phrase en anglais) sont souvent plus riches sémantiquement que les mots qui les composent pris séparément. Par exemple, le mot composé "imprimante laser" est plus précis que "imprimante" et "laser" pris isolément. Cet argument a conduit à leur large utilisation en RI. L'utilisation des mots composés en RI est discutée en détails dans la section 1.3.1.
- **Les concepts :** qui sont des expressions pris généralement d'une structure conceptuelle, tels que les thésaurus ou les ontologies.

#### 1.2.1.5 La création de l'index

Au terme du processus d'indexation, un ensemble de structure de données sont crées. Ces dernières permettent un accès efficace à la représentation des documents. Le fichier inverse est la structure de données la plus utilisée **[10] [132]**, il enregistre pour chaque descripteur les identificateurs des documents qui le contiennent et sa fréquence dans chacun de ces documents.

Généralement, les structures de données sont compressées avant d'être enregistrées sur le disque, ce qui permet de réduire la taille de l'index. Parmi les méthodes de compression utilisées on peut citer la méthode Elias Gamma **[203]** qui opère au niveau bit requérant ainsi beaucoup d'opérations pour la compression et la décompression. D'autres méthodes plus efficaces, opérant au niveau octet ont été proposées dans **[202]**.

D'autres caractéristiques sur un document, permettant de calculer la pertinence a priori d'un document indépendamment de toute requête, peuvent être calculées et stockées a ce stade **[58]**. Ces caractéristiques seront discutées dans le chapitre trois, section 3.5.

### 1.2.2 Appariement document-requête

La fonction d'appariement document-requête permet de mesurer la valeur de pertinence d'un document vis-à-vis d'une requête. Afin de réaliser cela, le SRI représente le document et la requête avec un même formalisme, puis le SRI compare les deux représentations. Le résultat de cette comparaison se traduit par un score qui détermine la probabilité de pertinence (degré de similarité ou degré de ressemblance) du document vis-à-vis de la requête. Cette fonction d'appariement est notée $RSV(d,q)$ (*R*etrieval *S*tatut *V*alue), où $d$ représente un document de la collection et $q$ la requête. Cette valeur permet ensuite au SRI d'ordonner les documents renvoyés à l'utilisateur.

### 1.2.3 Les modèles de recherche d'information

Un modèle de RI fournit une interprétation théorique de la notion de pertinence. Plusieurs modèles de RI on été proposés dans la littérature, ils s'appuient sur des cadres théoriques différents, théorie des ensembles, algèbre, probabilités, etc. Globalement, on distingue trois principales catégories de modèles: modèles booléens, modèles vectoriels et modèles probabilistes **[63]**.

#### 1.2.3.1 Le modèle booléen

Les premiers SRI développés sont basés sur le modèle booléen, même aujourd'hui beaucoup de systèmes commerciaux (moteurs de recherche) utilisent le modèle booléen. Cela est dû à la simplicité et à la rapidité de sa mise en œuvre.

Le modèle booléen est basé sur la théorie des ensembles et l'algèbre de Boole. Dans ce modèle, un document $d$ est représenté par un ensemble de mots-clés (termes) ou encore un vecteur booléen. La requête $q$ de l'utilisateur est représentée par une expression logique, composée de termes reliés par des opérateurs logiques : ET ($\wedge$), OU ($\vee$) et SAUF ($\neg$). L'appariement ($RSV$) entre une requête et un document est un appariement exact, autrement dit si un document implique au sens logique la requête alors le document est pertinent. Sinon, il est considéré non pertinent. La correspondance entre document et requête est déterminée comme suit :

$$RSV(d,q) = \begin{cases} 1 & si\ d\ appartient\ \text{à}\ l'ensemble\ décrit\ par\ q \\ 0 & sinon \end{cases} \quad (1.1)$$

Malgré la large utilisation de ce modèle, il présente un certain nombre de faiblesses :

- Les documents retournés à l'utilisateur ne sont pas ordonnés selon leur pertinence.
- La représentation binaire d'un terme dans un document est peu informative, car elle ne renseigne ni sur la fréquence du terme dans le document ni sur la longueur de document, qui peuvent constituer des informations importantes pour la RI.
- Il est difficile pour les utilisateurs de formuler de bonnes requêtes. Par conséquent, l'ensemble des documents trouvés est souvent trop grand, pour les requêtes courtes, ou complètement vide dans le cas de requêtes longues.
- Ce modèle ne supporte pas la réinjection de pertinence.
- Les tests effectués sur des collections d'évaluation standards de RI ont montré que les systèmes booléens sont d'une efficacité de recherche inférieure.

Afin de remédier à certains problèmes de ce modèle, des extensions ont été proposées, parmi elles on trouve : le modèle booléen basé sur la théorie des ensembles flous [**107**] [**154**], le modèle booléen étendu [**167**].

#### 1.2.3.2 Les modèle vectoriel

Le modèle vectoriel de base a été introduit par Salton **[166]**, concrétisé dans le cadre du système SMART. Ce modèle se base sur une formalisation géométrique. En effet, les documents et les requêtes sont représentés dans un même espace, défini par un ensemble de dimensions, chaque dimension représente un terme d'indexation. Les requêtes et les documents sont alors représentés par des vecteurs, dont les composantes représentent le poids du terme d'indexation considéré dans le document (la requête). Formellement, si on a un espace $T$ de termes d'indexation de dimension $n$, $T = \{t_1, t_2, \dots, t_j, \dots t_n\}$. Un document $d_i$ est représenté par un vecteur $d_i(w_{i1}, w_{i2}, \dots, w_{ij}, \dots w_{in})$. Une requête $q$ par un vecteur $q(w_{q1}, w_{q2}, \dots, w_{qj}, \dots w_{qn})$.

Où $w_{ij}$ (resp. $w_{qj}$ ) représente le poids du terme $t_j$ dans le document $d_i$ (respectivement dans la requête $q$).

Le modèle vectoriel offre des moyens pour la prise en compte du poids de terme dans le document. Dans la littérature, plusieurs schémas de pondération ont été proposés. La majorité de ces schémas prennent en compte la pondération locale et la pondération globale [**127**].

La pondération locale permet de mesurer l'importance du terme dans le document. Elle prend en compte les informations locales du terme qui ne dépendent que du document. Elle correspond en général à une fonction de la fréquence d'occurrence du terme dans le document (noté *tf* pour *t*erm *f*requency), exprimée ainsi :

$$tf_{ij} = 1 + log\left(f(t_i, d_j)\right) \quad (1.2)$$

Où $f(t_i, d_j)$ est la fréquence du terme $t_i$ dans le document $d_j$.

Quant à la pondération globale, elle prend en compte les informations concernant le terme dans la collection. Un poids plus important doit être assigné aux termes qui apparaissent moins fréquemment dans la collection. Car les termes qui apparaissent dans de nombreux documents de la collection ne permettent pas de distinguer les documents pertinents des documents non pertinents (i.e. peu utile pour la discrimination). Un facteur de pondération globale est alors introduit. Ce facteur nommé $idf$ (*i*nverted *d*ocument *f*requency), dépend d'une manière inverse de la fréquence en document du terme et exprimé comme suit :

$$idf = log\left(\frac{N}{n_i}\right) \quad (1.3)$$

Où $n_i$ est la fréquence en document du terme considéré, et $N$ est le nombre total de documents dans la collection.

Les fonctions de pondération combinant la pondération locale et globale sont référencées sous le nom de la mesure $tf \times idf$. Cette mesure donne une bonne approximation de l'importance du terme dans les collections de documents de taille homogène. Cependant, un facteur important est ignoré, la taille du document. En effet, la mesure $(tf \times idf)$ ainsi définie favorise les documents longs, car ils ont tendance à répéter le même terme, ce qui accroit leur fréquence, par conséquent augmentent la similarité de ces documents vis-à-vis de la requête. Pour remédier à ce problème, des travaux ont proposé d'intégrer la taille du document dans les formules de pondération, comme facteur de normalisation **[158] [180].**

L'appariement document-requête dans le modèle vectoriel, consiste à trouver les vecteurs documents qui s'approchent le plus de vecteur de la requête. Cet appariement est obtenu par l'évaluation de la distance entre les deux vecteurs. Plusieurs mesures de similarité ont été définies **[194],** dont les plus courantes sont décrites dans le tableau 1.1 ci-dessous.

| Mesures | Formules |
|---|---|
| Le produit scalaire | $RSV(q,d_i) = \sum_{j=1}^{\|T\|} w_{qj}.w_{ij}$ |
| La mesure de cosinus | $RSV(q,d_i) = \frac{q.d_i}{\|q\|.\|d_i\|} = \frac{\sum_{j=1}^{\|T\|} w_{qj}.w_{ij}}{\sqrt{\sum_{j=1}^{\|T\|} {w_{qj}}^2 \sum_{j=1}^{\|T\|} {w_{ij}}^2}}$ |
| La mesure de Dice | $RSV(q,d_i) = \frac{2 \times \sum_{j=1}^{\|T\|} w_{qj}.w_{ij}}{\sum_{j=1}^{\|T\|} {w_{qj}}^2 + \sum_{j=1}^{\|T\|} {w_{ij}}^2}$ |
| La mesure de Jaccard | $RSV(q,d_i) = \frac{\sum_{j=1}^{\|T\|} w_{qj}.w_{ij}}{\sum_{j=1}^{\|T\|} {w_{qj}}^2 + \sum_{j=1}^{\|T\|} {w_{ij}}^2 - \sum_{j=1}^{\|T\|} w_{qj}.w_{ij}}$ |

**Tableau 1.1 Les mesures de similarité utilisées dans le modèle vectoriel**

Le modèle vectoriel caractérisé par sa prise en compte du poids des termes dans les documents, permet de retrouver des documents qui répondent partiellement à une requête.

De plus, ce modèle offre un moyen facile pour classer les résultats d'une recherche, qui est basée sur la similarité potentielle entre documents et requête. L'inconvénient majeur de modèle vectoriel est qu'il repose sur l'hypothèse de l'indépendance des termes d'indexation, or ces termes dans les documents sont souvent sémantiquement liés.

Plusieurs variantes du modèle vectoriel ont été proposées, pour remédier à cette limitation, c'est-à-dire prendre en compte la dépendance entre termes d'indexation. Parmi elles, on trouve, le modèle vectoriel généralisé **[204]**, le modèle LSI (Latent Semantic Indexing) **[19] [64] [68] [73] [176]** et le modèle connexionniste **[24] [110]**.

#### 1.2.3.3 Les modèles probabilistes

##### 1.2.3.3.1 Le modèle probabiliste de base

Le modèle probabiliste est fondé sur la théorie des probabilités **[160].** Il trie les documents selon leur probabilité de pertinence vis-à-vis d'une requête. La fonction de classement (tri) de ce modèle est exprimée ainsi :

$$RSV(q,d) = \frac{P(Per|q,d_i)}{P(NPer|q,d_i)} \quad (1.4)$$

L'idée de base de cette fonction est de sélectionner les documents ayant à la fois une forte probabilité d'être pertinents et une faible probabilité d'être non pertinents à la requête.

Où $P(Per|q,d_i)$ et $P(NPer|q,d_i)$ : la probabilité qu'un document $d_i$ soit pertinent ($Per$) vis-à-vis de la requête $q$ (respectivement non pertinent ($NPer$)).

En appliquant la formule de bayes pour les deux probabilités on obtient :

$$P(Per|q,d_i) = \frac{P(Per|q).P(d_i|Per,q)}{P(d_i)} \quad (1.5)$$

$$P(NPer|q,d_i) = \frac{P(NPer|q).P(d_i|NPer,q)}{P(d_i)} \quad (1.6)$$

Où :

$P(d_i)$ est la probabilité de choisir le document $d_i$ , on considère qu'elle est constante ;

$P(d_i|Per,q)$ indique la probabilité que $d_i$ fait partie des documents pertinents pour la requête $q$ ;

$P(d_i|NPer,q)$ indique la probabilité que $d_i$ fait partie des documents non pertinents pour la requête $q$ ;

$P(Per|q)$ et $P(NPer|q)$ indiquent respectivement la probabilité de pertinence et de non-pertinence d'un document quelconque (avec $(Per|q) + P(NPer|q) = 1$ ) qui sont fixes.

Après remplacement dans la fonction de tri, on aura la formule suivante :

$$RSV(q,d) = \frac{P(d_i|Per,q)}{P(d_i|NPer,q)} \quad (1.7)$$

Si on suppose que les termes d'indexation sont indépendants, alors on peut estimer les deux probabilités ainsi :

$$P(d_i|Per,q) = \prod_{t_j \in d_i} P(t_j|Per,q) \times \prod_{t_j \notin d_i} 1 - P(t_j|Per,q) \quad (1.8)$$

$$P(d_i|NPer,q) = \prod_{t_j \in d_i} P(t_j|NPer,q) \times \prod_{t_j \notin d_i} 1 - P(t_j|NPer,q) \quad (1.9)$$

Où $P(t_j|Per,q)$ indique la probabilité d'apparition du terme $t_j$ sachant que le document appartient à l'ensemble des documents pertinents et $P(t_j|NPer,q)$ indique la probabilité d'apparition du terme $t_j$ sachant que le document appartient à l'ensemble des documents non pertinents.

En posant $p_i = P(t_j|Per,q)$ , $q_i = P(t_j|NPer,q)$ et $p_i = q_i$ pour les termes qui n'apparaissent pas dans la requête, et après simplification, le calcul du score de correspondance entre un document et une requête peut être exprimé ainsi :

$$RSV(d_i, q) = \sum_{t_i \in q} log \left[\frac{p_i(1-q_i)}{q_i(1-p_i)}\right] \quad (1.10)$$

Afin de classer les documents avec cette formule, il faut estimer les valeurs des deux probabilités $p_i$ et $q_i$. En l'absence de collection (documents) d'apprentissage ; on peut attribuer la valeur fixe à $p_i$ comme par exemple 0.5 **[52]**; comme elles peuvent être estimées à l'aide de l'avis de l'utilisateur sur les résultats d'une première recherche (réinjection de pertinence).

#### 1.2.3.3.2 Le modèle de langue

Les modèles statistiques de langue sont exploités avec beaucoup de succès dans divers domaines : la reconnaissance de la parole [**100**], la traduction automatique **[29]** [**133]**, la recherche d'information **[94] [111] [116] [151]**, etc.

L'utilisation des modèles de langue en RI remonte à 1998 **[151]**. Le principe de ce modèle consiste à construire un modèle de langue pour chaque document, soit $M_d$ , puis de calculer la probabilité qu'une requête $q$ puisse être générée par le modèle de langue du document, soit $P(q|M_d)$ **[26] [151]**. Le modèle de langue utilisé est souvent le modèle uni-gramme, la probabilité $P(q|M_d)$ est alors exprimée ainsi :

$$P(q|M_d) = \prod_{t \in Q} P(t|M_d) \quad (1.11)$$

$P(t|M_d)$ peut être estimée en se basant sur l'estimation maximale de vraisemblance (maximum likelihood estimation). Elle est donnée par :

$$P(t|M_d) = \frac{tf(t,d)}{|d|} \quad (1.12)$$

Où $tf(t, d)$ est la fréquence du terme $t_i$ dans le document $d$.

Pour remédier au problème posé par les mots de la requête absents dans le document, qui ont pour effet d'avoir la probabilité $P(q|M_d)$ nulle ; des techniques de lissage (smoothing) sont utilisées, dont le lissage de Laplace (ajouter-un), le lissage de Good-Turing, le lissage Backoff, le lissage par interpolation, etc **[209]**. Leur principe consiste à assigner des probabilités non nulles aux termes, qui n'apparaissent pas dans les documents **[26]**.

Une description détaillée de ces modèles est présentée dans le chapitre trois (3).

### 1.2.4 La reformulation de la requête

La reformulation de la requête est un processus permettant la construction d'une nouvelle requête, plus à même de représenter les besoins en information de l'utilisateur. Elle est souvent opérée par ajout et/ou réévaluation des poids des termes de la requête initiale.

Les techniques de reformulation de la requête peuvent être classées en tenant en compte de plusieurs paramètres **[36]**:

- Les sources de données utilisées pour l'expansion de requête ;
- La méthode de sélection des termes d'expansion : la relation de cooccurrence, les mesures d'information, les techniques de classification, etc.
- La sélection des termes d'expansion en considérant chaque terme de la requête individuellement, ou la requête dans son ensemble ;
- La représentation de la requête (document) comme un ensemble de mots simples (sac de mots) ou une représentation prenant en compte les relations de proximité entre termes ;
- Le type de terme d'expansion (mot simple ou mot composé).
- L'intervention de l'utilisateur dans le processus d'expansion de la requête (automatique, manuelle, semi-automatique).

Différentes sources de données sont utilisées pour reformuler la requête initiale. Elles peuvent être **[132]**: (1) des ressources externes, telles que les ontologies, les thésaurus et la relation de cooccurrence entre termes dans la collection. Les méthodes basées sur ces sources sont dites méthodes globales. (2) Les documents résultants de la première recherche; les méthodes basées sur ces sources sont dites méthodes locales. Ces méthodes sont également connues sous le nom de réinjection de pertinence. La réinjection de pertinence a montré son efficacité avec différents modèles de la RI **[116] [128] [162] [163] [169]** et a affiché de meilleurs résultats que les méthodes globales.

D'autres sources de données ont été utilisées, on peut citer les textes d'ancre **[57]**, les logs des moteurs de recherche **[198]**, les liens dans les documents Wikipedia **[7]** et les FAQs **[156]**. D'autres études utilisent des informations complémentaires fournies par l'utilisateur pour l'extension de la requête telles que, les balises **[41]**, les images **[43]** et les catégories **[199].**

La reformulation de la requête peut être réalisée par l'utilisateur, dans ce cas elle est dite manuelle, ou par le système (dite automatique) comme elle peut être réalisée conjointement par l'utilisateur et le système, dans ce cas elle est dite semi-automatique.

#### 1.2.4.1 Reformulation par réinjection de la pertinence

Ces méthodes impliquent que l'utilisateur doit sélectionner les documents qu'il considère pertinents à partir des résultats issus de sa requête initiale. Ce jugement de pertinence de l'utilisateur est ensuite exploité pour reformuler la requête initiale en modifiant le poids des termes qu'elle contient et/ou en ajoutant de nouveaux termes considérés utiles pour retrouver des documents pertinents.

La technique de réinjection de pertinence a été mise en place à l'origine dans le modèle vectoriel. Rocchio **[163]** a proposé le modèle de reformulation de requête suivant :

$$Q_N = \alpha . Q_O + \beta . \frac{1}{|R|} \sum_{r \in R} r - \frac{1}{|R'|} \sum_{r' \in R'} r' \qquad (1.13)$$

Où :

$Q_N$ est le vecteur de la nouvelle requête (reformulée) ;

$Q_O$ est le vecteur de la requête originale ;

$R$ est l'ensemble des vecteurs $r$ des documents jugés pertinents par l'utilisateur ;

$R'$ est l'ensemble des vecteurs $r'$ des documents jugés non pertinents par l'utilisateur ;

$\alpha$ ,$\beta$ ,$\delta$ sont les paramètres de la reformulation.

On peut remarquer que cette formule permet d'obtenir une nouvelle requête dont le vecteur se rapproche des vecteurs des documents jugés pertinents et s'éloigne des vecteurs des documents jugés non pertinents.

Dans le modèle probabiliste, la réinjection de pertinence est mise en place directement dans le modèle de mesure de pertinence. Elle consiste à revoir les poids des termes de la requête **[161]**, comme suit :

$$w_{q_j} = log \left[ \frac{r' + 0.5/(r - r' + 0.5)}{(df_j - r' + 0.5)/(n - df_j - r + r' + 0.5)} \right] \qquad (1.14)$$

Où :

$r$ représente le nombre de documents pertinents ;

$r$' est le nombre de documents pertinents contenant le terme $q_j$ ;

$df_j$ est le nombre de documents contenant le terme $q_j$ ;
$n$ est le nombre total de documents dans la collection.

#### 1.2.4.2 La réinjection par pseudo feedback (réinjection aveugle)

Ces méthodes de reformulation nommées aussi, pseudo-réinjection de pertinence (ou blind) sont effectuées de manière automatique. Elles se basent sur l'hypothèse que les documents les mieux classés (les premiers) sont considérés comme pertinents. Le système utilise alors les premiers documents pour reformuler la requête.
La variante de la formule de Rocchio (formule (1.13)) pour la réinjection automatique de la requête est exprimée par la formule suivante :

$$Q_N = \alpha . Q_O + \beta . \frac{1}{|R|} \sum_{r \in R} r \qquad (1.15)$$

On voit dans cette formule que l'expansion de la requête est uniquement positive, car on ne peut faire aucune hypothèse sur les documents non pertinents, mais rien ne nous empêche de prendre les derniers documents de la liste comme non pertinents.

Plusieurs travaux **[25] [88]** ont tenté d'évaluer l'impact de la pseudo-réinjection, en variant le nombre de nombre de termes à rajouter à la requête. Ils montrent que la performance du système est obtenue lorsque la requête est construite entre 20 et 40 termes.
Les méthodes d'expansion automatique de la requête proposées dans le contexte du modèle de langue seront étudiées en détails dans le chapitre 3.

## 1.3 Au-delà des mots simples

La majorité des approches (modèles) développées en RI se basent sur l'utilisation des mots simples comme unités de représentation des documents et des requêtes, souvent appelé représentation en sac de mots. Ces approches posent deux problèmes, l'ambiguïté des mots et leur disparité **[22]**.

> ***L'ambiguïté des mots,*** se rapporte à des mots lexicalement identiques et portant des sens différents. Ce problème conduit à avoir des documents non pertinents en réponse à une requête contenant des mots ambigus. Par exemple, des documents sur la « programmation java » peuvent être renvoyés en réponse à la requête « aéroport de java », car le terme java contient plus d'un sens (ile, programmation, etc.).

***La disparité des mots (en anglais word mismatch),*** se réfère à des mots lexicalement différents mais portant un même sens. Ce problème implique que des documents pertinents ne sont pas retrouvés en réponse à une requête, car ils utilisent des mots différents que ceux de la requête pour exprimer le même concept. Par exemple, des documents contenant le terme «tablette tactile » peuvent ne pas être retrouvés en réponse à une requête « I-pad».

En plus de l'expansion de la requête, étudiée dans la section 1.2.4, diverses approches ont été proposées pour remédier à ces problèmes. Ces approches permettent d'incorporer ou d'utiliser des informations conceptuelles ou sémantiques dans les méthodologies de recherche. Nous présentons ci-dessous les trois types d'approches les plus utilisées, l'indexation sémantique, l'indexation conceptuelle et l'indexation par des mots composés.

### 1.3.1 L'indexation par des mots composés

L'indexation par des mots composés est une technique qui permet l'utilisation des mots composés comme unités d'indexation. Ceci a pour objectif une représentation plus précise du contenu sémantique des documents et des requêtes.

L'idée d'utiliser les mots composés comme unités d'indexation est que ces derniers sont moins ambigus et plus précis que les mots simples. Par exemples : le terme « *java* » est ambigu, par contre les mots composés « *ile de java* » et « *langage java* » sont non ambigus ; le terme « *voiture électrique* » est plus spécifique que l'un des deux termes « *voiture* » et « *électrique* ».

L'intuition est claire, les mots composés aident à construire des unités d'indexation non ambiguës et plus précises et peuvent par conséquent améliorer la précision de la RI.

Cinq paramètres sont généralement à considérer dans l'exploitation des mots composés comme unités d'indexation.

1. **La directionnalité** : c'est-à-dire l'ordre des termes. Dans certains cas la préservation de l'ordre est important pour préserver le sens de l'unité d'indexation. Par exemple, « Recherche d'information », dans d'autre cas l'ordre n'est pas important, « Recherche et développement ». Peu de travaux existent en RI où sont utilisés les mots composés directionnels (ex : **[141]**), la plupart des travaux exploitant les mots composés sont basés sur la non directionnalité de ces derniers **[67] [140]**. Cependant,

Fagan *et al* **[67]** ont rapporté des problèmes dans l'utilisation des mots composés non directionnels.

2. **La distance** : la distance entre les termes formant le mot composé (l'adjacence ou la non-adjacence des termes) ; l'intensité de liens entre termes opérationnalisée à travers la distance reflète la proximité sémantique entre termes. La capture de cette proximité est importante pour la recherche d'information.

   Les études effectuées en RI sur l'extraction des mots composés supposent que la cooccurrence des mots dans les éléments fortement structurés (c.-à-d., une phrase) est plus significative que dans les éléments moins structurés (c.-à-d., des paragraphes ou des sections). Ainsi, la recherche sur l'extraction des mots composés a été dominée par l'analyse de phrase. L'analyse empirique justifie de limiter l'extraction des mots composés aux combinaisons des termes apparaissant dans la même phrase. Martin *et al* **[134]** ont constaté que 98% de combinaisons syntaxiques associent les termes qui sont dans la même phrase et sont séparés par cinq mots au plus. Fagan **[66]** a constaté que la restriction de l'extraction des mots composés à une fenêtre de distance de cinq termes est presque aussi efficace que des mots composés extraits dans une phrase sans une telle restriction, soutenant ainsi les résultats de Martin *et al* **[134]**. D'autres travaux **[34] [130]** ont adopté cette hypothèse et ils ont utilisé une fenêtre de cinq mots pour l'extraction des mots composés.

3. **La taille des mots composés** : en principe la taille d'un mot composé peut être de n'importe quelle longueur (supérieure ou égale à 2). Dans la pratique les mots composés longs conduisent à des index très spécifiques qui sont généralement moins utiles pour la RI.

4. **La pondération des mots composés** : les différents schémas de pondération proposés pour l'attribution d'un poids à un mot simple dans un document, prennent généralement en considération trois facteurs : le facteur de pondération local ($tf$), qui mesure l'importance du terme dans le document ; un facteur de pondération globale, mesurant la représentativité globale du terme dans la collection ($idf$) et un facteur de normalisation qui prend en compte la longueur du document.

   Cependant, pour les mots composés, il n'y pas de schéma de pondération bien accepté. En général, trois approches sont proposées pour la pondération des mots composés :

- L'utilisation de la fréquence ($tf$) du mot composé dans le document **[104]**; en se basant sur le fait que la fréquence d'un terme est corrélée avec son importance **[126]**.
- L'adaptation de schéma de pondération ($tf \times idf$) appliqué pour les mots simples. Comme c'est le cas dans **[17]**.
- L'utilisation des mesures d'association, telle que l'information mutuelle **[14]**.

**5. Repérage des mots composés** : trois approches principales existent dans la littérature pour le repérage et l'extraction des mots composés.

**a. Approches linguistiques** : ces approches **[177]**, se basent sur une analyse syntaxique partielle ou l'utilisation de patrons (templates) syntaxiques pour détecter les mots composés **[9] [27] [40] [59] [98] [102] [164]**. Le plus souvent, un ensemble de patrons syntaxiques comme (NOM NOM) ou (NOM PREP NOM) est utilisé pour l'identification. Malgré les nombreuses études consacrées à ce problème, il n'existe pas encore, à notre connaissance, une méthode effective qui permette de distinguer les termes des non termes d'un point de vue syntaxique. Des exemples d'outils issus de ces approches sont TreeTagger **[175],** AZ NOUN PHRASER de l'université de l'Arizona. Cependant, ces approches souffrent d'un inconvénient majeur puisque elles sont basées sur des règles, et ces règles sont dépendantes de la langue.

**b. Approches statistiques** : ces approches se basent sur la cooccurrence des termes dans le corpus pour extraire les mots composés **[6] [49] [66] [115] [160] [165] [193] [194],** et cela en partant de l'hypothèse que des termes (souvent réduits à deux ou trois mots) qui apparaissent ensemble dans le texte sont susceptibles de représenter un concept. Les mots composés sont extraits ici soit en se basant sur leurs fréquences observées dans le corpus soit par l'utilisation des mesures d'association qui déterminent le degré d'association entre les mots composants.

- **Les mesures d'association** : les mesures d'association permettent de calculer « un score d'association » pour chaque paire de termes candidat dans le corpus ; ce score indique le potentiel de ce candidat d'être reconnu comme un mot composé. Plusieurs mesures d'association ont été proposées dans la littérature, telles que l'information mutuelle et le coefficient de Dice **[133]**. Toutes ces métriques adoptent le postulat suivant : « les mots composés sont ceux dont les composants apparaissent ensembles plus souvent que par hasard », cela est obtenu en comparant la fréquence observée dans le corpus

et la fréquence attendue (qui se base sur l'hypothèse d'indépendance des termes).

Pour la reconnaissance de mots composés de taille supérieure à deux (taille>2), des algorithmes ont été proposés **[190]**, d'autres mesures ont été définies **[61]** et des extensions des métriques précédentes ont été proposées **[135]**.

Les approches statistiques ont un avantage considérable puisqu'elles ne nécessitent aucune autre information ou ressource pour l'extraction des mots composés. Elles exploitent seulement les informations apparaissant dans le corpus, d'où leurs flexibilité et portabilité (i.e. : elles ne dépendent ni de la langue du corpus ni du domaine traité par le corpus).

c. **Approches mixtes** : ces approches se basent sur les régularités statistiques et les patrons syntaxiques pour l'extraction des mots composés **[56] [122] [140] [181] [187]**. Fagan **[66] [67]** a comparé l'apport pour la RI des mots composés extraits statistiquement et des mots composés extraits linguistiquement, en utilisant l'analyse syntaxique, la troncature et la normalisation. L'évaluation a montré que les mots composés extraits linguistiquement ont donné des résultats semblables ou plus faibles que les résultats obtenus avec les mots composés extraits statistiquement. Les gains de performance constatés en utilisant les mots composés extraits statistiquement dans son expérience étaient de l'ordre de 17% à 39%.

L'exploitation des mots composés dans le contexte du modèle de langue est présentée en détails dans le chapitre trois (3), section 3.4.1.1.

### 1.3.2 L'indexation sémantique

L'indexation sémantique consiste à représenter les documents et les requêtes par les sens des termes qu'ils contiennent plutôt que par les termes eux mêmes. Ce type d'indexation se base sur les techniques de désambiguïsation sémantique (Word Sense Disambiguation WSD).

La désambiguïsation sémantique est une tâche qui a pour but la sélection du sens approprié pour un terme dans un contexte donné **[145]**. Elle implique donc, l'association pour un terme donné dans un texte avec un sens ou une définition qui est distinguable des autres sens ou définitions attribués à ce terme.

L'étude de Krovetz *et al* **[108]** a été la première et la plus élaborée à s'être penchée sur la pertinence de la relation de correspondance du sens des termes dans la requête et les documents. Ils ont découvert que lorsqu'une requête est bien formulée et décrit bien le besoin en information, alors l'ambiguïté est moindre. Considérant les deux requêtes suivantes : « big bank » et « bank economic financial monetary fiscal » ; la deuxième requête est moins ambiguë que la première. Ceci est dû à ce qu'ils ont appelé l'effet de collocation. Un ensemble de termes isolement ambigus contribuent ensemble à désambiguïseur implicitement le sens de terme (bank dans l'exemple).
Ils ont aussi étudié la distribution des sens d'un terme dans les documents. Ils ont trouvé que certains sens du terme occurrent plus fréquemment que les autres sens.

Voorhees **[195]** a proposé une méthode de désambiguïsation basée sur Wordnet, qui est une structure conceptuelle organisée autour de la notion de synset. Un synset regroupe des termes (simples ou composés) ayant un même sens dans un contexte donné. Les synsets sont liés par différentes relations telles que l'Hyperonymie (is-a) et son inverse, l'Hyponymie (instance-de) **[15]**.
Pour déterminer le sens d'un mot ambigu, les synsets (sens) de ce mot sont classés en utilisant la valeur de cooccurrence calculée entre le contexte de ce mot et un voisinage (Voorhees l'a appelé hood) contenant les mots du synset dans la hiérarchie de WordNet. Le synset le mieux classé est alors choisi comme le sens approprié du mot ambigu analysé.
Voorhees a expérimenté cette approche sur une collection de test désambiguïsée (les requêtes de la collection de test sont aussi désambiguïsées manuellement) par rapport aux performances du même processus sur la même collection dans son état d'origine (ambigu). Les tests ont été effectués sur les collections CACM **[168]**, CISI, Cranfield 1400, MEDLINE, et Time **[183]**. Les résultats obtenus sur chacune de ces collections montrent une nette dégradation des performances du SRI. Ceci est expliqué probablement par le taux faible de désambiguïsation.

Sanderson **[171]** a étudié l'effet de la désambiguïsation sémantique en RI, particulièrement l'effet d'une désambiguïsation incorrecte sur les performances de la RI. Il a mené un ensemble d'expérimentations en utilisant la collection Reuters. Il a constaté que l'introduction de l'ambiguïté artificielle dans la collection Reuters ne dégrade pas les performances du système. Il a aussi constaté que les requêtes courtes, comprenant un à deux termes sont

fortement affectées par l'introduction de l'ambiguïté, alors que les requêtes longues le sont beaucoup moins (ce qui confirme l'effet de la collocation noté par Krovetz).
Par contre le travail inverse : désambiguïsation de la collection ambiguë de Reuters, affecte les performances du système, et cela selon le taux de performance de la désambiguïsation effectué (qui est contrôlable). Avec un taux de performance de désambiguïsation égal à 75%, il a constaté que les performances du système se dégradent. Il a conclu que la désambiguïsation est utile lorsque le taux de performance de la désambiguïsation est supérieur à 90%.

Comme Sanderson, Gonzalo *et al* **[76]** ont mesuré l'effet de la désambiguïsation erronée sur les performances de la RI. Ils ont utilisé trois schémas d'indexation : les termes, le sens des termes (pas de prise en compte de la synonymie) et les synsets de WordNet. Des expérimentations ont été effectuées sur la collection SEMCOR**,** une partie du corpus Brown. Cette collection (SEMCOR) est désambiguïsée avec les synsets de WordNet.
Leur expérimentation est effectuée en deux étapes, la première consiste à évaluer l'effet de la désambiguïsation (l'effet de l'indexation avec le sens de terme et le synset). La seconde étape consiste à évaluer l'effet de la désambiguïsation incorrecte. Comme l'a fait Sanderson.
Les résultats obtenus dans la première partie montrent qu'un gain de 11% de performance est obtenu avec l'utilisation des sens des termes, et un gain de 14 % avec l'utilisation des synsets de WordNet.
Quant aux résultats obtenus dans la seconde étape; ils montrent qu'avec un taux d'erreur de désambiguïsation inférieur à 90% les performances se dégradent, ce qui rejoint le constat fait par Sanderson. En revanche, avec une indexation par synset, les performances du système restent meilleures que celles de la configuration basique et cela avec un taux d'erreur de désambiguïsation de 30%. Pour un taux d'erreur de désambiguïsation variant entre 30% et 60% les résultats ne montrent pas de différence significative avec l'indexation par mots-clés (configuration basique). Ce qui est en désaccord avec le constat de Sanderson.
En résumé, l'application des techniques de désambigüisation sémantique en RI présente des limitations en terme de calcul et d'efficacité. Des résultats mixtes ont été reportés dans de récentes séries d'expérimentations, réalisées à CLEF 2008 et à CLEF 2009, dans la tâche Robust–WSD **[3]**.

### 1.3.3 L'indexation conceptuelle

L'indexation conceptuelle consiste à représenter un document par un ensemble de concepts. Ces concepts sont tirés de structures conceptuelles, qui peuvent être génériques (cas de WordNet pour la langue anglaise) ou spécifique à un domaine (cas de MESH pour le domaine médical). Les structures conceptuelles peuvent être construites manuellement, automatiquement ou semi-automatiquement. Ces structures incluent les taxonomies de concepts, les ontologies, les réseaux sémantiques, les dictionnaires, les thésaurus, etc., et qui diffèrent dans la forme de représentation utilisée et dans les relations entre concepts considérées.

Plusieurs travaux en RI ont utilisés ce type d'indexation dans des domaines spécifiques comme le domaine du sport **[114]**, le domaine légal **[186]**, ou dans un domaine générique **[17] [205]**.

Woods **[205]** a présenté une méthode d'indexation conceptuelle, qui consiste à extraire les descripteurs conceptuels (concepts) automatiquement à partir des documents sans faire référence à aucune ressource, puis à organiser ces concepts de manière dynamique sous forme d'une taxonomie de concepts.

L'index conceptuel obtenu peut servir alors à la recherche ou à la navigation dans la collection des documents. L'organisation des concepts dans la taxonomie se base sur la relation de subsumption (is-a), dans laquelle chaque concept identifié est relié à ses concepts parents. Pendant la phase de la recherche, l'algorithme de recherche permet de déterminer l'emplacement de la requête dans la taxonomie.

La méthode ainsi décrite, a été évaluée sur la collection de pages du manuel d'UNIX et sur d'autres collections de Sun Microsystems. Les résultats obtenus ont montré que cette méthode d'indexation améliore le taux de succès de 13% par rapport à la meilleure stratégie d'indexation classique (*TWIDF* : Term Weighted Inverse Document Frequency). Le taux de succès est une mesure qui est définie comme le pourcentage de requêtes pour lesquelles une réponse est obtenue dans les dix premiers documents retournés.

Baziz *et al* **[17]** ont défini une méthode d'indexation conceptuelle basée sur l'utilisation de l'ontologie linguistique WordNet. Chaque document est représenté sous forme d'un réseau sémantique particulier (appelé noyau sémantique), dans lequel les nœuds représentent les concepts et les arcs (bidirectionnels) représentent la distance sémantique entre concepts liés.

Après expérimentations, les résultats obtenus ont montré que cette méthode d'indexation conceptuelle n'améliore pas les résultats obtenus avec une indexation classique (mots-clés). Par contre, les résultats obtenus avec une combinaison des deux types d'indexation (classique et conceptuelle) ont montré une nette amélioration de la précision.

## 1.4 Evaluation des SRI

Dès l'apparition des premiers SRI, la pratique d'évaluation desdits systèmes est apparue; les premières évaluations datent de 1953**[112]**.

L'évaluation des SRI est abordée selon deux angles différents. L'un est dit « paradigme système », qui vise à évaluer les performances du système essentiellement en termes de qualité des documents retournés par le système, c'est-à-dire leur pertinence vis-à-vis des besoins en information des utilisateurs. L'autre est dit « paradigme usager », qui est centré sur la satisfaction de l'utilisateur, et non sur les performances intrinsèques du système, en modélisant le comportement des utilisateurs en situation de recherche.

Nous présentons ci-dessous seulement l'approche basée « système », la plus utilisée dans le domaine de la RI. Elle se base sur deux éléments essentiels à savoir : des collections de test et des mesures d'évaluation.

### 1.4.1 Les collections de test

Une collection (ou corpus) de test constitue le moyen d'évaluation des SRI. Elle est généralement composée d'un ensemble de documents, d'un ensemble de requêtes et des jugements de pertinence associés à ces requêtes. L'évaluation d'un SRI consiste à comparer les résultats retournés par ce dernier par rapport aux jugements de pertinence. Des mesures d'évaluation, décrites dans la section suivante, sont utilisées pour effectuer cette comparaison.

Les collections de test sont le résultat de projets d'évaluation qui se sont multipliés depuis les années 1970, on peut citer la collection CACM[1], la collection CISI[2], la campagne CLEF[3] et la campagne TREC[4].

La campagne TREC constitue à ce jour la campagne de référence dans le cadre de l'évaluation des systèmes de recherche d'information et cela depuis son lancement en 1992 **[196] [197]**. L'objectif de cette campagne est de proposer une plate-forme qui réunit des

[1] http://www.search-engines-book.com/collections/
[2] ftp://ftp.cs.cornell.edu/pub/smart/cisi
[3] www.clef-campaign.org
[4] http://trec.nist.gov/

collections de test, des tâches spécifiques et des protocoles d'évaluation pour chaque tâche afin de mesurer les différentes stratégies de recherche**[23]**.

Les tâches proposées se sont diversifiées d'une campagne à une autre (à raison d'une par an) ; parmi les tâches proposées dans TREC 2012 on peut citer : recherche d'information sur le web, recherche d'information médical, recherche d'information dans les micros blogs, recherche d'information contextuelle, etc.

La taille des collections augmente au fil des années, passant de 2 Go dans TREC 1 à 25 To dans TREC 2011. Chaque collection est composée d'un certain nombre de documents, allant de quelques milliers à plusieurs millions. Les documents sont codés à l'aide de SGML dans un format spécifique TREC. La figure 1.2 illustre un exemple d'un document TREC.

```
<DOC>
<DOCNO> WSJ920324-0113 </DOCNO>
<DOCID> 920324-0113. </DOCID>
<HL>    Venture of Kimbaco </HL>
<DATE> 03/24/92 </DATE>
<SO> WALL STREET JOURNAL (J), PAGE C9 </SO>
<CO>    H.TSI </CO>
<MS> FINANCIAL (FIN) </MS>
<IN> ALL BANKS, BANKING NEWS AND ISSUES (BNK)
SECURITIES (SCR) </IN>
<NS> JOINT VENTURES (JVN) </NS>
<RE> FAR EAST (FE)
HONG KONG (HK)
PACIFIC RIM (PRM)
SOUTH KOREA (SK)
</RE>
<LP>
NEW YORK -- South Korean merchant banking firm Kimbaco said it joined
 with Hong Kong brokerage house Peregrine Securities to form a new
 investment firm Kimbaco Peregrine Capital Ltd.
The firm will seek out cross-border transactions and direct investment
 opportunities in Asia, with special emphasis on U.S.-Korean ventures.
</LP>
<TEXT>
</TEXT>
</DOC>
```

**Figure 1.2 Exemple d'un document TREC.**

Nous utilisons les collections de documents suivantes dans ce mémoire :

- ***WSJ90-92*** : est une partie de la collection Wall Street Journal qui a été distribué sur TREC TIPSTER disque 2, contenant des articles de presse des années 1990 à 1992, elle comporte 74520 documents.
- ***AP88*** : elle contient des articles de Associated Press newswire de l'année 1988, elle comporte 79919 documents.
- ***WT10g*** : est une collection web de 10 giga-octets et comporte 1692096 documents.

- ***.GOV*** : est une collection web de taille 20 giga-octets et comporte 1,25 millions de documents.

Chaque collection TREC a généralement 50 à 100 requêtes correspondantes. Une requête TREC est structurée comme suit: un identifiant de requête unique TREC, un titre, une description plus détaillée du besoin en information et une rubrique qui explique dans quelles circonstances un document doit être jugé pertinent ou non pertinent pour une requête. Un exemple d'une requête TREC est montré dans la figure 1.3. Dans la plupart de nos expérimentations, nous n'utilisons que le champ titre des requêtes, car ceux-ci sont similaires aux besoins en information des utilisateurs sur le web **[99].**

```
<top>
<num> Number: 562
<title>  world population growth
<desc> Description:
What is the outlook for world population growth?
<narr> Narrative:
Relevant documents include projections of and
discussion of world population growth. Growth of
individual nations' populations is relevant, but
data on states within the U.S. is not relevant.
</top>
```

**Figure 1.3 Exemple d'une requête TREC**

### 1.4.2 Mesures d'évaluation de SRI

Le principal objectif d'un système de recherche d'information est de restituer à l'utilisateur tous les documents pertinents et de rejeter tous les documents non pertinents. Cet objectif est évalué à l'aide de différentes mesures d'évaluation **[170]**. On présente ci-dessous les plus utilisées.

- **La précision** : est le rapport du nombre de documents pertinents restitués par le système ($SP$) sur le nombre total de documents restitués ($R$), exprimée ainsi :

$$précision = \frac{SP}{R} \tag{1.16}$$

- **Le rappel** : est le rapport du nombre de documents pertinents restitués ($SP$) sur le nombre total de documents pertinents ($P$), exprimé ainsi :

$$rappel = \frac{SP}{P} \tag{1.17}$$

Des mesures complémentaires au rappel et précision ont été définies, il s'agit de bruit et de silence.

- **Le bruit** : la mesure d'évaluation bruit est une notion complémentaire à la précision, elle est définie par $B = 1 - P$ où $P$ est la précision du SRI.
- **Le silence** : la mesure d'évaluation silence est une notion complémentaire au rappel, elle est définie par $S = 1 - R$ où $R$ est le rappel du SRI.
- **Courbe de Rappel-Précision** : un système idéal devrait retourner tous les documents pertinents et que les documents pertinents ; c'est à dire un taux de précision et de rappel égal à 100%. Cette situation ne se produit pas dans un système réel car le taux de précision et de rappel sont antagonistes. En effet, Lorsque la précision augmente, le rappel diminue et inversement. Ainsi, pour mesurer les performances d'un système il faut utiliser les deux mesures conjointement. Cela est réalisé en calculant la paire des mesures (taux de rappel, taux de précision) à chaque document restitué.

  Nous considérons par exemple une requête pour laquelle il existe six (6) documents pertinents dans le corpus. Le tableau 1.2 illustre le calcul de la précision et de rappel pour les dix (10) premiers documents retournés par un SRI. La lettre (P) précise que le document est pertinent.

| Rang du document renvoyé | Pertinence | Rappel | Précision |
|---|---|---|---|
| Doc1 | P | 0,17 | 1 |
| Doc2 | | 0,17 | 0,5 |
| Doc3 | P | 0,33 | 0,66 |
| Doc4 | | 0,33 | 0,5 |
| Doc5 | P | 0,5 | 0,6 |
| Doc6 | | 0,5 | 0,5 |
| Doc7 | P | 0,67 | 0,57 |
| Doc8 | | 0,67 | 0,5 |
| Doc9 | | 0,67 | 0,44 |
| Doc10 | P | 0,83 | 0,5 |

**Tableau 1.2 Exemple de calcul de rappel et de précision pour une requête**

La figure 1.4 illustre la courbe de rappel et précision correspondante aux résultats du tableau 1.2. Pour rendre la courbe lisible on ne garde que la précision calculée à chaque point de rappel (c'est à dire à chaque document pertinent restitué).

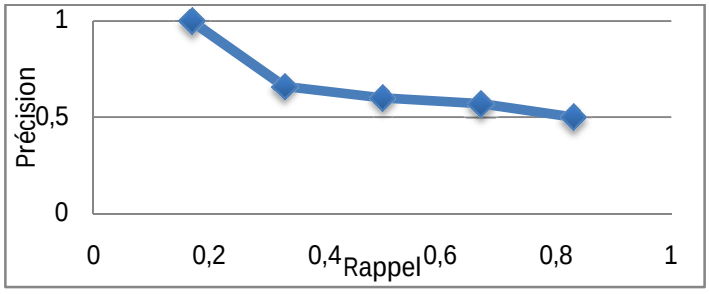



**Figure 1.4 Courbe de rappel et précision**

La courbe ci-dessus permet d'évaluer les performances du système pour la requête considérée. Afin d'évaluer le système pour un ensemble de requêtes, on calcul la moyenne des précisions à chaque niveau de rappel. Comme les niveaux de rappel ne sont pas unifiés pour l'ensemble des requêtes, on retient généralement 11 points de rappel standards de 0 à 1 avec un pas de 0.1.

Les valeurs de précision sont calculées par une interpolation linéaire. Pour deux points de rappel, $i$ et $j$, $i < j$, si la précision au point i est inférieure à celle au point j, on dit que la précision interpolée à i égale la précision à $j$.

Cette interpolation est encore discutable, mais présente un intérêt dans l'évaluation de systèmes de recherche d'information. Elle permet entre autre de construire des courbes décroissantes plus simple à comparer **[10].**

- **Les mesures alternatives**
  - **La précision exacte :** notée aussi R-précision, elle est calculée sur les « R » premiers documents retournés par le système, sachant que la requête admet « R » documents pertinents.
  - **La précision moyenne non interpolée (MAP):** la précision moyenne non interpolée (Average Mean Precision) est calculée en deux étapes. D'abord on calcule la précision moyenne pour une requête donnée ($AP_q$), ainsi pour chaque document pertinent retrouvé on calcule sa précision ($pr(d_i)$) qui est égale au nombre de documents pertinents retrouvés sur le rang de ce document ; pour les documents retrouvés non pertinents leur précision est égale à zéro.

    La précision moyenne pour une requête donnée est alors obtenue en calculant la moyenne des précisions des documents pertinents, exprimée ainsi :

$$AP_q = \frac{1}{N}\sum_{i=1}^{N} pr(d_i) \tag{1.18}$$

Avec

$$pr(d_i) = \begin{cases} \frac{r_{n_i}}{n_i} & si\ d_{ij}\ est\ retrouvé \\ 0 & sinon \end{cases} \tag{1.19}$$

Où $n_i$ dénote le rang du document $d_i$ qui a été retrouvé et qui est pertinent pour la requête, $r_{n_i}$ est le nombre de documents pertinents retrouvé au rang $n_i$ et $N$ est le nombre total de documents pertinents pour la requête $q$.

Dans la seconde étape, on calcule la précision moyenne pour un ensemble de requêtes, en effectuant la moyenne des précisions moyennes de chaque requête, elle est exprimée ainsi :

$$MAP = \frac{1}{M}\sum_{j=1}^{M} AP_{q_j} \quad (1.20)$$

Où $AP_{q_j}$ dénote la précision moyenne pour la requête « $j$ » et $M$ représente le nombre de requêtes considérées.

## 1.5 Conclusion

Dans ce chapitre nous avons passé en revue les principaux concepts de la RI. Nous avons, particulièrement, introduit des notions de base, telles que le besoin en information, la requête, le document et la pertinence. Nous avons aussi décrit les processus de base de la RI, à savoir l'indexation, l'appariement requête-document et la reformulation de la requête. Ensuite, nous avons étudié les différents modèles de la RI et nous avons abordé les différentes directions investies pour introduire la sémantique en RI. Enfin, l'évaluation des systèmes de recherche d'information est traitée.

La plupart des modèles de recherche d'information que nous avons évoqués sont développés afin d'exploiter uniquement le contenu textuel des documents. Cependant, avec l'avènement du web, le document présente de nouvelles dimensions (sources d'information) autre que le contenu textuel. En effet, les documents web sont généralement des documents structurés via les balises HTML et interconnectés entre eux par des liens hypertextes.

Ces nouvelles sources d'information sur le document doivent être intégrées dans les modèles de RI, afin d'améliorer les performances de la recherche d'information. Dans le chapitre suivant, nous abordons la recherche d'information sur le web.

# Chapitre 2

# Recherche d'information sur le web

## 2.1 Introduction

Les problématiques posées en recherche d'information sur le web sont identiques à celles posées par la recherche d'information classique (indexation, appariement, etc.).

Dès, l'apparition du web, différentes catégories d'outils se sont développées pour répondre à ces problématiques et pour faire face aux nouveaux chalenges posés par le web. La spécificité du web réside dans le type de documents manipulés (pages HTML), qui sont des documents structurés, la nature hypertexte du web et le nombre d'utilisateur et le type de requête qu'ils utilisent pour exprimer leur besoin en information. Généralement, ces outils de RI sur le web examinent la combinaison de diverses sources d'information telles que : le contenu textuel du document et la structure du web, pour classer les documents web en réponse à une requête utilisateur.

Ce chapitre présente un aperçu sur la RI sur le web. Dans la première section, nous présentons les différences entre la RI classique et la RI sur le web. Dans la seconde section, nous examinons les différentes sources d'information spécifiques au document web : la structure interne des documents web et la structure des liens. Nous décrivons dans la troisième section les différentes approches utilisées pour combiner les sources d'information sur un document web, dans le but d'améliorer la pertinence de la recherche d'information.

## 2.2 Différences entre la RI classique et la RI sur le web

Les SRI classiques sont souvent développés et utilisés dans des environnements bien contrôlés tel que les bibliothèques, où les collections de documents sont généralement de petites tailles et les utilisateurs ont des besoins en informations bien spécifiques.

Le web diffère sur plusieurs points avec les autres ressources documentaires rencontrées habituellement en recherche d'information. Parmi les facteurs distinctifs, on peut citer, le volume du web; la dispersion, l'hétérogénéité et la nature dynamique de l'information dans cet espace ; enfin, les utilisateurs du web proviennent de divers horizons avec des niveaux de connaissances différents et expriment leur besoins en information avec peu de mots.

Nous analysons brièvement ci-dessous ces facteurs distinctifs.

### 2.2.1 Le volume du web

Le volume d'informations accessible sur le web ne se mesure plus en giga-octets mais en tira-octets voir en péta-octets et exa-octets. Déjà, à la fin de l'année 1995, le moteur Altavista avait reporté qu'il a indexé approximativement 30 millions de pages web statiques. En juin 2000, Netcraft[5] recense plus de 12 millions de sites web, ce qui représente environ 800 millions de pages web. En Mai 2013, une étude a rapporté que le moteur de recherche Google comporte dans son index plus de 46 milliards de pages web[6].

D'un autre coté, chaque mois, plus de 100 milliards de recherches sont effectuées, au travers des moteurs de recherche commerciaux sur le web **[55]**.

Cette augmentation de la taille du web et le nombre de requêtes soumises sont à l'origine de la dégradation des performances des processus de recherche tant en terme d'efficience que d'efficacité. Plus précisément **[117]:**

- L'allongement des délais de réponse;
- L'augmentation des temps d'indexation;
- La diminution de la précision de la recherche.

### 2.2.2 L'hétérogénéité de l'information

L'hétérogénéité des ressources d'information sur le web inclut plusieurs points : les ressources du web sont écrites dans plusieurs langues (une centaine). Une variante de formats sont utilisées, rien que pour le texte on peut citer : HTML, PDF, XML, RTF, DOC, etc., et elles utilisent différents encodages dans la plupart du temps ils sont incompatibles.

[5] http://news.netcraft.com
[6] http://www.worldwidewebsize.com

L'hétérogénéité peut être aussi sémantique, tous les thèmes sont traités sur le web, et ces thèmes sont abordés par diverses sources, qui peuvent être des sources scientifiques, de vulgarisations ou de commercialisations. Cette hétérogénéité créée de nouveaux défis significatifs pour la recherche d'information qui sont : l'interopérabilité entre sources d'information et l'amplification des phénomènes de polysomie et d'homographie, qui ont pour effet d'augmenter le bruit lors d'une recherche.

### 2.2.3 La disparité de l'information

La disparité est une caractéristique qui traduit l'occurrence disséminée de l'information dans de larges collections de documents. Et compte tenu du volume important d'information disponible sur le web, la récupération de toute l'information répondant à une requête de l'utilisateur est une tâche ardue. La disparité de l'information a pour effet d'augmenter le silence en recherche d'information sur le web.

### 2.2.4 La nature dynamique du web

En raison de la nature dynamique du web, les informations peuvent être ajoutées ou supprimées facilement. Il est estimé que 40% des pages web sont modifiées tous les mois.
En outre, différents pages évoluent à des rythmes différents. Par exemple, les pages liées aux nouvelles (les médias), sports et les pages personnelles ont tendance à changer plus fréquemment que celles hébergées dans des domaines éducatifs ou gouvernementaux **[2]**.
Cette nature dynamique du web rend la mise à jour et la maintenance des index des moteurs de recherche extrêmement difficile.

### 2.2.5 La fiabilité de l'information

L'information sur le web est produite par diverses sources ; cette diversité pose le problème, non moins crucial, de la qualité et de la fiabilité de l'information récupérée. En effet, il ne suffit pas de récupérer de l'information sur un sujet, encore faut-il savoir quelle valeur lui attribuer. L'information récupérée peut être une bonne information, une information non complète ou une information fausse ce qui est plus nuisible.

### 2.2.6 Les utilisateurs du web et leurs requêtes

Le plus souvent, les utilisateurs expriment leur besoin en information avec de petites requêtes, qui contiennent peu de mots, en moyenne 2.35 mots. Les courtes requêtes expriment d'une manière inexacte et ambiguë le besoin en information de l'utilisateur. En plus, la plupart des

utilisateurs consultent seulement les premières pages retournées ; et s'engagent rarement dans le processus de la reformulation de la requête.

Ces caractéristiques du web rendent difficile pour les outils de recherche d'information actuels la tâche de sélection d'information désirée parmi le grand nombre de ressources qui répondent aux besoins des utilisateurs. Les limites des outils actuels de recherche d'information sur le web ont incité les chercheurs à développer de nouvelles approches pour aider à améliorer l'exactitude de la tâche de sélection de ressource.

### 2.2.7 La structure du Web

Le web peut être considéré comme un graphe orienté, dont les nœuds sont des pages web identifiées par des adresses URL et les arcs sont des hyperliens entre ces pages web. Trois approches ont été investies pour étudier la structure du web. La première est dite macroscopique, elle s'intéresse à la structure grande échelle du web, son but est d'avoir une vue de loin de graphe du web. La seconde approche est dite microscopique, sont objectif est de rechercher dans le graphe du web de petites structures (ensemble de pages : communautés) qui se répètent souvent. Quant à la dernière approche, elle s'intéresse au calcul de certaines propriétés statistiques du web.

## 2.3 Les sources d'informations sur un document web

L'objectif majeur de la RI est le développement de stratégies qui permettent d'identifier tous les documents pertinents pour une requête de l'utilisateur. Dans la RI traditionnelle, seul le contenu textuel du document est considéré comme source d'information appropriée pour mesurer la pertinence d'un document vis-à-vis d'une requête.

Dans le contexte d'un document web, d'autres sources d'information indépendantes du contenu textuel de document, peuvent être exploitées afin d'améliorer les performances de la RI. Ces sources d'information peuvent être : la structure du document, la structure des hyperliens. La figure 2.1 illustre les différentes sources d'information sur un document web.

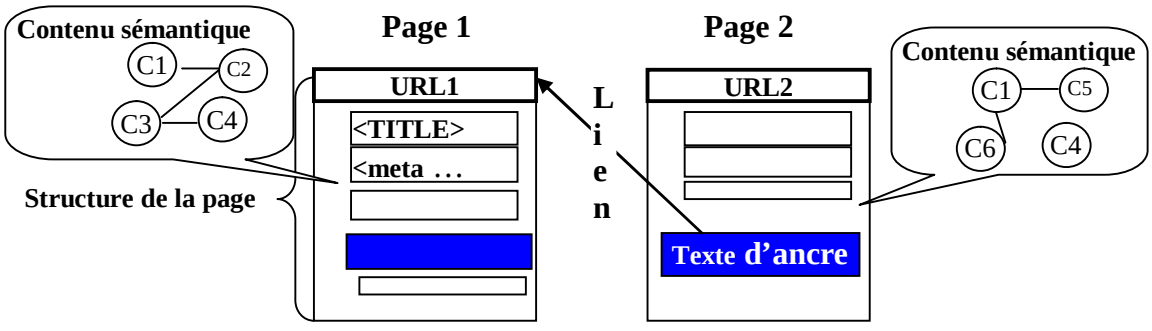



**Figure 2.1 Les sources d'information sur un document web**

### 2.3.1 Exploitation de la structure du document (page web)

Les SRI traditionnels négligent typiquement les informations sur la structure d'un document. La raison principale est que généralement cette information est non disponible ou difficile à acquérir **[54].**

Il est reconnu que la pertinence de la recherche peut être améliorée en tenant compte de la structure d'un document. Plusieurs moteurs de recherche utilisent les balises de HTML pour améliorer la fonction d'appariement des documents. Les moteurs de recherche actuels (Google, Bing, Yahoo) donnent un bon score à un document qui contient les termes de la requête dans le titre de la page web.

Cutler *et al* **[54]** ont mené une étude sur l'apport de la structure des pages HTML pour améliorer la pertinence de la RI. La méthode d'indexation proposée consiste à associer à chaque terme un vecteur de fréquence noté $tfv$, qui contient la fréquence d'apparition du terme dans une des classes de balises. Six classes de balises ont été utilisées : Anchor, <H1>-<H2>, <H3>-<H6>, STRONG, TITLE et le texte plein (toutes les autres balises).

Lors de la recherche un autre vecteur à six éléments noté $CIV$ (Class Importance Vector) est utilisé, chaque élément de ce vecteur représente un facteur d'importance associé pour chaque classe de balises. L'importance (poids) d'un terme dans un document est alors calculée comme suit :

$$w = (tf \times CIV) \times idf \tag{2.1}$$

Ainsi, la traditionnelle formule de pondération $tf$ est étendue comme suit : $tf \times CIV$, qui tient compte de la fréquence du terme dans une classe et de l'importance accordée à cette classe.

Les expérimentations menées ont montré que, l'usage de la structure des pages HTML améliore sensiblement la pertinence de la RI.

Olgivie *et al* **[44] [146] [147]**, ont montré que la surpondération des termes apparaissant dans le texte des balises TITLE, ALT et FONT dans le cadre de la recherche de page d'entrée et de page nommée, n'apporte qu'un petit gain d'efficacité.

D'autre travaux ont utilisé les métadonnées (des données pour décrire les données sur lesquelles elles portent) en RI. Agosti *et al* **[5]** ont utilisé 15166 pages de la bibliothèque du

Congrès[7], et ont observé que la recherche utilisant seulement le contenu a donné des résultats légèrement meilleurs que la recherche utilisant le contenu et les métadonnées. Cette étude est peu concluante car seulement une petite fraction des pages collectées contient des métadonnées.

Zhang *et al* **[211]** ont construit un ensemble de pages web (artificiellement, en ajoutant des métadonnées), et les ont soumis à un ensemble de moteurs, dans le but de mesurer l'impact des métadonnées sur la visibilité de ces pages dans les moteurs de recherche. Ils ont constaté qu'aucune des pages web construites qui contenant les termes des requêtes dans le champ de métadonnées ne figure dans les résultats de recherche. Ceci est dû vraisemblablement à l'utilisation des techniques anti-spam par ces moteurs de recherche.

En plus de la structure interne d'une page web, son adresse URL peut être une source d'information lors du calcul de la pertinence de la page, soit en considérant l'URL comme un texte, dans ce cas on peut appliquer les méthodes de recherche plein texte pour l'appariement entre les termes de la requête et le texte de l'URL, ou en considérant d'autres caractéristiques de l'URL comme : sa forme ou la présence de certains caractères, dans le but d'estimer la pertinence a priori d'un document (indépendamment de la requête).
Kraaij *et al* **[106]** ont utilisé la forme (type) de l'URL pour estimer la probabilité qu'une page soit une page d'entrée. Quant à Kamps *et al* **[103]**, ils proposent trois autres mesures, afin d'estimer la probabilité a priori de pertinence d'une page, en fonction du nombre de slash ('/') dans l'URL, du nombre de caractères de l'URL et enfin de la somme de nombre de caractères '. ' dans la partie domaine de l'URL et du nombre de '/' dans la partie chemin de l'URL.
Les expérimentations menées ont montré que ces trois mesures sont de bons indicateurs afin d'estimer la probabilité a priori de pertinence d'une page.

### 2.3.2 Exploitation de la structure des hyperliens

L'analyse des liens a été étudiée avant l'apparition même du web, le domaine des réseaux sociaux s'y est intéressé pour diverses applications, comme la communication (détection d'espionnage, optimisation des transmissions). D'autres domaines se sont également penchés sur l'analyse des liens, c'est le cas de la bibliométrie, où l'analyse des références bibliographiques entre articles scientifiques est utilisée afin d'estimer le facteur d'impact des

[7] http://www.loc.org

articles ou des journaux. D'autres mesures ont été également identifiées, la co-citation et le couplage bibliographique.

Avec l'émergence du web, l'analyse de la structure des liens est au cœur de nombreux travaux en RI. Les deux principales utilisations des liens en recherche d'information concernent la collecte des pages web (Crawl) **[39]** et le classement des documents (Ranking). Les liens sont également utilisés pour d'autres fins comme : la classification et la catégorisation des pages web **[37]**, la recherche des ressources dupliquées sur le web **[20]**, la recherche de pages similaires **[60]**, etc.

Nous nous intéressons particulièrement dans la suite de cette section à l'usage des liens dans la fonction de calcul de pertinence (classement des pages web).

Les méthodes de calcul de pertinence exploitant la structure des liens peuvent être réparties en trois classes distinctes, chacune d'entre elles est basée sur l'une de ces hypothèses suivantes **[47]**.

#### 2.3.2.1 L'hypothèse de recommandation

Elle stipule que si une page est pointée par un lien alors cette page est recommandée par la page qui l'a référencé. Ainsi, une page avec beaucoup de liens entrants est une page fortement recommandée (populaire, autorité) et donc susceptible d'être mieux classée. Plusieurs algorithmes et méthodes de « Ranking » se sont basés sur cette hypothèse, pour le calcul de l'importance de la page. Nous décrivons ci-dessous quelques algorithmes représentatifs de cette classe.

- **Le nombre de liens**

  Deux types de liens sont considérés pour une page web : les liens entrants et les liens sortants. Plusieurs études se sont penchées sur l'utilité du nombre de ces liens pour la recherche d'information.

  Dans **[214],** il est noté que le nombre de liens entrants peut fournir une indication sur l'importance, la popularité et la qualité de la page.

  Kamps *et al* **[103]** ont utilisé le nombre de liens entrants et sortants pour prédire l'importance de la page (probabilité a priori de pertinence) dans le cadre de la recherche de pages d'entrées et de pages nommées « Named-page ». Ils ont constaté que le nombre de liens entrants est un bon indicateur pour prédire la pertinence de la page.

- ***PageRank***

  L'algorithme de *PageRank (PR)* développé par Brin et Page en 1998 **[28]** est à l'origine du moteur de recherche Google. Cet algorithme assigne une valeur de popularité (un score, un *PR*) à toutes les pages du web indépendamment de toute requête.

  La mesure *PageRank* est une distribution de probabilité sur les pages. Elle mesure en effet la probabilité *PR*, pour un utilisateur navigant au hasard, d'atteindre une page donnée. Elle repose sur un concept très simple : un lien émis par une page A vers une page B est assimilé à un vote de A pour B. Plus une page reçoit de votes, plus cette page est considérée comme importante. Le *PageRank* se calcule de la façon suivante :

$$PR(p) = (1-d) \times \frac{1}{T} + d \times \sum_{i=1}^{k} \frac{PR(p_i)}{C(p_i)} \quad (2.2)$$

  Où :

  $PR(p)$ est le *PageRank* de la page $p$ ;

  $T$ est le nombre total de pages web indexées;

  $d$ est un paramètre fixé à 0.85;

  $C(p_i)$ est le nombre de liens sortant de la page $p_i$, et $k$ est le nombre de pages qui pointent la page $p$.

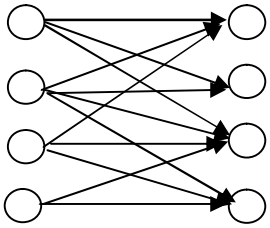
  Le score *PageRank* est utilisé afin de réordonner la liste des pages retournées. L'algorithme de *PageRank* n'améliore pas significativement les résultats des stratégies qui ignorent les hyperliens, selon les évaluations faites lors des campagnes de TREC [**92**] [**93**].

  Plusieurs extensions de *PageRank* ont été proposées, tels que le *PageRank* personnalisé [**91**], qui consiste à incorporer les centres d'intérêts des utilisateurs dans le calcul de *PageRank*.

- ***HITS***

  Kleinberg a proposé un algorithme dit *HITS* (Hyperlink Induced Text Search) **[105]** pour identifier les documents autorités au moment de la recherche. L'algorithme est basé sur l'analyse de la matrice d'adjacence des pages retournées pour une requête donnée. Les pages web concernant le sujet de la requête peuvent être soit des Hubs ou des Autorités. Les pages Autorités contiennent de l'information sur le sujet. Par contre, les pages Hubs pointent les pages Autorités. Ces deux types de pages ont une relation de renforcement mutuelle entre elles, ainsi : un bon Hub est une page qui

pointe beaucoup de bonnes Autorités, et une bonne Autorité est une page qui est pointée par beaucoup de bons Hubs. La figure 2.2 ci-dessous illustre la relation entre ces deux types de pages.



**Hubs** **Autorités**

**Figure 2.2 Les pages Hubs et Autorités**

L'algorithme HITS calcule deux scores pour une page : le score d'Autorité et le score de Hub. L'évaluation de ces deux scores se fait au moment de l'interrogation, selon les deux formules ci-dessous, traduisant la relation de renforcement mutuelle entre les deux types de pages :

$$A_p = \sum_{q \to p} H_q \quad (2.3)$$

$$H_p = \sum_{p \to q} A_q \quad (2.4)$$

Où $A_p$ et $H_p$ représentent respectivement le score d'Autorité et de Hub de la page « $p$ », $q \to p$ indique l'ensemble des pages qui pointent la page « $p$ » et $p \to q$ indique l'ensemble des pages pointées par la page « $p$ ».

Le calcul des scores $A_p$ et $H_p$ se fait d'une manière itérative. Il se termine lorsque la convergence des scores s'opère, ces scores sont alors utilisés pour classer les documents.

Plusieurs autres variantes ont été proposées élargissant les méthodes présentées précédemment, comme l'algorithme SALSA **[120]**.

#### 2.3.2.2 L'hypothèse de proximité sémantique

Elle stipule que si deux pages sont liées par un hyperlien, alors il est fort probable que les deux pages portent sur un même sujet, c'est-à-dire elles sont sémantiquement liées. Des travaux proposent alors de propager une fraction du score de pertinence d'une page vers ses pages voisines **[48] [174]**. De manière générale ces approches se basent sur deux étapes : la première consiste à classer les pages web selon la similarité de leur contenu avec la requête.

Dans la seconde étape, les premières pages seulement sélectionnées précédemment transmettent une fraction de leur score vers leurs voisines. L'orientation des liens peut être respectée ou ignorée. Ainsi, un nouveau score est calculé pour chaque page selon la formule suivante :

$$RSV(d_i, q) = Sim(d_i, q) + \lambda \sum_{j=1}^{m} Sim(d_j, q) \quad (2.5)$$

Où $RSV(d_i, q)$ représente le nouveau score du document $d_i$ dépendant de son score initial noté $Sim(d_i, q)$ et du score de ses $m$ voisins et $\lambda$ est un facteur de pondération.

Savoy *et al* **[174]** ont noté que les différentes évaluations effectuées sur un corpus d'articles et sur une collection de pages extraites du web n'ont pas amélioré d'une manière significative les résultats, en appliquant différentes valeurs pour le facteur λ. Cependant, ils ont constaté une amélioration en ne considérant que les pages possédant un meilleur score initial.

#### 2.3.2.3 L'hypothèse de la description du texte d'ancre

Un texte d'ancre donne une petite description de la page référencée. Ceci le rend utile pour l'indexation de la page pointée plutôt que la page source. Une manière simple d'exploiter cette source est de prendre en compte tous les textes d'ancre pointant un document, au même titre que le contenu de ce document lors du calcul de son score de pertinence.

Dès les années 90, des moteurs de recherche tels que le moteur Altavista ont recouru à l'utilisation du texte d'ancre ; et il a été montré qu'il est un élément utile pour la RI sur le web **[148]**.

Des études académiques ont rejoint cette constatation, et cela dans le cadre de la recherche de pages d'entrée et de la recherche du site **[47] [106].**

Plusieurs facteurs (la plupart statistiques) laissent à penser que le texte d'ancre est utile pour l'amélioration de la qualité de la RI **[65]** : les requêtes des utilisateurs sont dans la plupart des cas de petites tailles, cette caractéristique est généralement partagée par les textes d'ancre; la requête comme le texte d'ancre tendent à ne pas être des phrases complètes, et leurs vocabulaires ainsi que leurs formes grammaticales sont semblables; les termes de la requête et les termes du texte d'ancre appartiennent généralement au même espace concept (décrivent le même concept).

En plus de ces facteurs liés à la requête, d'autres facteurs liés au document plaident pour l'usage des textes d'ancre : il est fréquemment observé que le texte d'ancre contient des termes qui n'apparaissent pas dans le contenu de la page pointée, donc le texte d'ancre est une

source d'information supplémentaire pour le contenu de la page ; des points de vue différents sur un même contenu sont donnés par les auteurs des pages qui pointent une page donnée.
Ce qui est reproché à l'usage des textes d'ancre ; est que certaines pratiques de la part des auteurs des pages peuvent avoir un impact négatif sur le résultat de la recherche. Par exemple, pour la requête « more evil than Satan himself », le moteur Google qui utilise le texte d'ancre proposait, en octobre 1999, le site web de Microsoft comme réponse la plus pertinente. Cette réponse est la conséquence de l'existence des termes de la requête dans les textes d'ancre (afin de dénoncer les pratiques commerciales de Microsoft) qui pointent le site web Microsoft.
Les textes d'ancre peuvent aussi contenir des termes peu informatifs, exemple (cliquez ici, Bas, Haut, etc.). Pour contourner ce problème et prendre en compte le contexte du texte d'ancre plusieurs travaux ont proposé d'inclure le texte encadrant le texte d'ancre. Ainsi, Chakrabarti *et al* **[38]** ont examiné la distribution du terme "Yahoo" autour d'ancre de http://www.yahoo.com dans 5000 pages. Ils ont trouvé que la plupart des occurrences de ce terme font partie d'un cadre de 50 termes, et ont montré que l'usage du texte de proximité améliore le rappel en dépit de la précision.
Glover *et al* **[75]** quant à eux ont montré que l'usage d'un cadre de 25 termes autour du texte d'ancre améliore les performances de classification des pages web.

D'autres sources d'information sur un document web ont étés utilisées, comme le facteur temps **[62] [121]** et le rapport information/bruit **[214]**.

### 2.4 La combinaison des sources d'information

Il est reconnu que la combinaison de différentes sources d'information sur un document peut améliorer l'efficacité d'un SRI **[51]**. Cette combinaison est réalisée suivant deux approches:

1. ***Combinaison de résultats :*** elle consiste à traiter chaque source d'information (ou plusieurs) comme un SRI à part, chacun de ces systèmes peut utiliser différentes stratégies de recherche; puis à combiner les résultats de recherche obtenus par ces derniers dans une seule liste ordonnée. Cette approche se réfère à la fusion de données dans les environnements de RI traditionnels et à la méta-recherche dans le contexte du web.
2. ***Combinaison de facteurs :*** elle consiste à développer des modèles (ou étendre les modèles existants) qui supporteront la combinaison de plusieurs sources d'information sur un document sous un même cadre. Souvent, ce sont des modèles basés sur l'apprentissage automatique.

### 2.4.1 Combinaison de résultats

Plusieurs stratégies de combinaison de résultats de recherche de différents SRI ont été proposées. Certaines se basent sur les scores des documents retournés par chaque système pour effectuer la fusion et d'autres se basent sur le rang des documents dans les listes retournées par chaque SRI. Le tableau 2.1 recense les différentes méthodes de combinaison de résultats.

| | |
|---|---|
| **Méthodes basées sur le rang** | Borda-fuse et Borda-fuse pondérée **[8]** ;<br>Réordonnancement **[118]**. |
| **Méthodes basées sur le score** | CombMNZ, CombSUM, CombMAX, CombMIN et CombMED **[71]** ;<br>Combinaison linéaire **[14]** |

**Tableau 2.1 Stratégies de combinaison de résultats**

### 2.4.2 Combinaison de facteurs

Cette approche de combinaison consiste à développer des modèles de RI qui supporteront explicitement la combinaison de plusieurs sources d'informations sur un document, sous un cadre unique.

Nous citons ci-dessous quelques travaux ayant été réalisés dans ce sens.

En 1988, Fox *et al* **[70]** ont mené un certain nombre d'expérimentations sur ce type de combinaison dans le cadre du modèle vectoriel. Ils ont proposé de décrire chaque représentation d'un document par un sous-vecteur. Par exemple : un sous-vecteur pour les termes, un autre pour les auteurs et un autre pour les citations. La fonction de similarité document-requête est alors une combinaison linéaire des similarités des différents sous-vecteurs avec la requête.

Tsikrika *et al* **[192]** ont utilisé le modèle de réseau bayésien pour la combinaison des différentes représentations d'un document web. Le modèle proposé est composé de quatre couches : couche documents, couche représentations des documents, couche requêtes et couche besoins des utilisateurs. Les deux premières couches sont construites au moment de l'indexation et les deux dernières au moment de l'interrogation. Deux sources d'information sont explorées, le contenu du document, à partir de laquelle une représentation du document est construite (représentation du contenu) et la structure d'hyperliens, pour laquelle deux types de représentation sont construites, le texte (étendu) d'ancre des liens entrants et le texte (étendu) des liens sortants de la page. Sur la base du typage sémantique des liens (trois types de liens : composition, séquence et référence), huit représentations au total sont obtenues. En plus des textes d'ancre, l'utilisation de l'algorithme *HITS* a été aussi investie.

Les expérimentations menées en utilisant la collection TREC WT2g ont montré que :

- Le contenu de la page et le texte d'ancre des liens entrants donne de meilleurs résultats que ceux obtenus avec le texte des liens sortants.
- Les résultats de la combinaison des différentes sources d'information ont montré que le contenu de la page est la source d'information la plus importante lors de la combinaison. De plus, une représentation qui donne des résultats individuels faibles peut améliorer les performances du système en la combinant avec d'autres représentations.
- L'application de l'algorithme *HITS* n'améliore pas les résultats obtenus avec le contenu de la page.

Dans le cadre du modèle de langue, plusieurs travaux ont été proposés pour combiner différentes sources d'information sur un document web **[62] [106] [121] [147] [214]**. Ce point est discuté en détails dans le chapitre suivant (section 3.5).

D'autres modèles basés sur des algorithmes d'apprentissage automatique (en anglais learning to rank) ont été proposés **[124] [206]**.

L'idée de base de ces algorithmes est de faire apprendre une fonction d'ordonnancement en assignant un poids pour chaque source d'information sur un document, puis utiliser la fonction obtenue pour estimer le score de pertinence de chaque document, et en fin ordonner les documents selon leur score de pertinence obtenu.

On distingue trois grandes approches d'algorithmes d'ordonnancement : par point (pointwise), par paire (pairwise) et par liste (listwise). Ces approches diffèrent sur leur façon de considérer le problème d'apprentissage **[123]**.

L'approche par point (pointwise) considère les documents séparément en entrée du système d'apprentissage. A chaque document est associé un score (ou un degré) de pertinence pour une requête donnée. Le problème d'apprentissage est alors assimilé à un problème de régression **[46]** ou de classification **[144]** respectivement.

L'approche par paire (pairwise) considère en entrée du système d'apprentissage des paires de documents ($d_i$, $d_j$) auxquels sont associées des jugements de préférence $r_{i,j}$ à valeur dans {-1,1}. Si $r_{i,j} = 1$ alors le document $d_i$ est préféré au document $d_j$, il doit être mieux classé dans la liste de résultat. La préférence est notée $d_i \succ d_j$. Au contraire, si $r_{i,j} = -1$ alors le document $d_j$ est préféré au document $d_i$ et on note $d_j \succ d_i$. Le problème d'apprentissage est ici

un problème de classification, dans le cas particulier de paires d'instances. Plusieurs techniques ont été proposées pour le classement des documents **[30] [191]**.

Enfin, l'approche par liste (listwise) considère en entrée du système d'apprentissage une liste ordonnée de documents. La fonction d'ordonnancement est apprise par minimisation de la distance entre la liste apprise et la liste de référence **[31]** ou par optimisation d'une mesure de recherche d'information **[208]**.

## 2.5 Conclusion

Nous avons abordé dans ce chapitre la RI sur le web. Particulièrement, les points suivants ont été étudiés. En premier lieu, nous avons énuméré les éléments distinctifs entre la RI classique et la RI sur le web. Ensuite, nous avons identifié et étudié les différentes sources d'information d'un document web. Enfin, nous avons présenté les différentes approches (méthodes) proposées pour combiner ces différentes sources d'information.

Parmi ces méthodes, nous nous intéressons à celles basées sur l'utilisation (exploitation) d'un cadre unique pour la combinaison des sources d'information sur un document web. Plus explicitement, nous utilisons dans notre travail, le modèle de langage comme cadre de combinaison de ces sources d'information.

Dans le chapitre suivant nous détaillons ce modèle, ainsi que les travaux réalisés dans ce cadre pour intégrer les différentes dimensions d'un document.

# Chapitre 3

# Modèles de langue pour la RI

## 3.1 Introduction

Depuis leur première utilisation en recherche d'information (RI) **[151]**, les modèles statistiques de langue ont acquis une grande popularité, en raison de leurs simplicité, efficacité et performance. Ces modèles ont été aussi appliqués avec succès dans différentes tâches de la RI, telles que la RI sur le web **[106]** la RI distribuée **[179]**, la recherche d'expert **[213]**.

Un des atouts de cette nouvelle classe de modèles de RI, concerne leur fondement théorique, basé sur la théorie des probabilités. De plus, ces modèles offrent la possibilité de combiner différentes informations (évidences) sur un document web pouvant être liées à la requête tel que le contenu du document ou non liées à la requête telles que la structure du document, la structure des liens, la date de création d'un document, etc.

Nos contributions exploitent largement ces modèles, notre objectif dans ce chapitre est de donner une vue globale de ces modèles et leur exploitation en RI. Plus précisément, nous décrivons en section 3.2, l'idée de base des modèles de langue en linguistique informatique ainsi que les différentes techniques de lissage. En section 3.3, nous présentons l'exploitation de ces modèle en RI. Nous passons ensuite en revue les différentes approches et travaux de la littérature directement en relation avec nos travaux, relatifs tout d'abord sur la prise en compte de la sémantique (relations entre termes) dans les modèles de langue, en section 3.4. Puis en section 3.5, ceux relatifs à la prise en compte des évidences indépendantes du contenu textuel de document dans le cadre des modèles de langue.

## 3.2 Les modèles de langue en linguistique informatique

Les premiers travaux qui ont porté sur l'utilisation des modèles de langue pour la recherche d'information sont inspirés de l'application avec succès des techniques statistiques de modélisation de langue en informatique linguistique. En effet, les deux domaines ont en commun plusieurs caractéristiques, entre autres, ils manipulent (possèdent) de grandes masses de texte, ce qui permet d'entraîner aisément les modèles statistiques pour des finalités différentes.

### 3.2.1 Idée de base

La modélisation statistique de langue est une distribution de probabilités sur toutes les séquences possibles «$S$» ou autres unités linguistiques dans une langue. Elle peut être vue comme un « processus génératif » qui consiste à estimer la probabilité $P(S|M)$ de générer une séquence de mots « $S$» à partir du modèle de la langue $M$.

Supposons que $S = m_1 m_2 \dots m_k$ est une séquence de $k$ mots, la probabilité de génération de la séquence « $S$ » est formulée ainsi :

$$P(S|M) = \prod_{i=1}^{k} P(m_i|m_{i-n+1} \dots m_{i-1}) \quad (3.1)$$

Dans cette formulation le terme $m_i$ ne dépend que de ses $n-1$ prédécesseurs immédiats, c'est une simplification qui est faite pour réduire le nombre de paramètres à estimer, dans ce cas le modèle est dit modèle *n-gramme.*

Dans le cas où $n = 1$, on parle du modèle *uni-gramme*. On considère dans ce modèle que les mots de la séquence sont générés indépendamment les uns des autres, la formule précédente devient alors :

$$P(S|M) = \prod_{i=1}^{k} P(m_i) \quad (3.2)$$

Quand $n = 2$, le modèle de langue est dit *bi-gramme*. Il est estimé en utilisant des informations sur la cooccurrence de paires de mots, c'est-à-dire chaque mot dépend seulement de son prédécesseur. La probabilité $P(S|M)$ est alors formulée ainsi :

$$P(S|M) = \prod_{i=1}^{k} P(m_i|m_{i-1}) \quad (3.3)$$

Dans le cas ou $n = 3$, on parle du modèle *tri-gramme*.

En pratique, ce sont ces trois modèles qui sont les plus utilisés.

Tous ces modèles nécessitent le calcul de la probabilité d'un *n-gramme (α)*, ce calcul se base sur l'estimation de vraisemblance maximale (MLE), qui revient à calculer la fréquence relative d'un *n-gramme* dans le corpus d'entrainement, soit $P_{ML}(\alpha)$. Cette probabilité est estimée comme suit :

$$P_{ML}(\alpha) = \frac{|\alpha|}{\sum_{\alpha_i \in C} |\alpha_i|} = \frac{|\alpha|}{|C|} \tag{3.4}$$

Où $|\alpha|$ est la fréquence d'occurrence du n-gramme (du mot) $\alpha$ dans le corpus, $\alpha_i$ est un n-gramme de la même longueur que $\alpha$ et le dénominateur $|C|$ correspond à la taille du corpus (c'est-à-dire le nombre total d'occurrences de n-grammes).

Cependant, quelle que soit la taille du corpus d'entrainement utilisé pour l'estimation de ces probabilités, il est fréquent que beaucoup de n-grammes n'apparaissent pas dans le corpus. Ceci entraîne l'attribution d'une probabilité nulle pour toute séquence contenant ces n-grammes.

Pour remédier à ce problème (clairsemence de données) et rendre les modèles plus généraux, des techniques de lissage sont utilisées. Le principe de ces techniques consiste à prélever une quantité de probabilités associées aux n-grammes observés dans le corpus d'entraînement et de la redistribuer sur les n-grammes non observés **[26]**. De cette façon, les n-grammes absents du corpus vont recevoir une probabilité non nulle.

### 3.2.2 Les techniques de lissage

Plusieurs techniques de lissage ont été proposées, nous présentons ci-dessous quelques unes d'entre elles :

**a. Lissage de Laplace :** cette méthode consiste à ajouter la fréquence un (1) à tous les n-grammes, appelé aussi ajouter-un. La probabilité du n-gramme $\alpha$ est estimée ainsi :

$$P_{LL}(\alpha) = \frac{|\alpha|+1}{\sum_{\alpha_i \in C} |\alpha_i|+1} = \frac{|\alpha|+1}{|C|+N} \tag{3.5}$$

Où $N$ est le nombre de n-grammes (distincts) et $|C|$ est la taille du corpus. L'inconvénient de cette méthode est qu'une grande masse de probabilité est distribuée sur les n-grammes non observés dans le corpus, d'où la faible participation des n-grammes observés dans la définition du modèle.

**b. Lissage de Good-Turing :** cette méthode permet l'ajustement de la fréquence « $r$ » d'un n-gramme ($\alpha$) en une fréquence dite corrigée « $r^*$ », exprimée ainsi :

$$r^* = (r + 1) \times \frac{n_r+1}{n_r} \tag{3.6}$$

Où $n_r$ est le nombre de n-grammes de fréquence « $r$ » dans la collection d'apprentissage. Ainsi, pour tout n-gramme ($\alpha$) l'estimation de sa probabilité devient alors :

$$P_{GT}(\alpha) = \frac{r^*}{\sum_{\alpha_i \in C} r_i^*} \tag{3.7}$$

Dans cette méthode la fréquence d'ordre $\frac{r^*}{r}$ pour un n-gramme vu sera redistribuée sur les n-grammes non vus dans le corpus. La méthode de Good-Turing est recommandée pour les n-grammes de faibles fréquences, car elle n'effectue pas de grandes modifications comme c'est le cas pour les n-grammes de grandes fréquences.

**c. Lissage de Backoff :** le principe de cette méthode consiste à utiliser un modèle de langue spécifique d'ordre inférieur, lorsqu'un n-gramme n'est pas observé dans le corpus. C'est le cas de lissage de Katz qui combine le modèle uni-gramme et le modèle bi-gramme, comme suit :

$$P_{Katz}(m_i|m_{i-1}) = \begin{cases} P_{GT}(m_i|m_{i-1}) & si \quad |m_{i-1}m_i| > 0 \\ \alpha(m_{i-1})P_{Katz}(m_i) & sinon \end{cases} \tag{3.8}$$

Dans cette méthode, la diminution de la fréquence utilisée dans $P_{GT}$ est redistribuée au modèle d'ordre inférieur (uni-gramme). $\alpha(m_{i-1})$ est un paramètre qui détermine la part de cette redistribution à $m_i$, déterminée comme suit :

$$\alpha(m_{i-1}) = \frac{1-\sum_{m_i:|m_{i-1}m_i|>0} P_{GT}(m_i|m_{i-1})}{1-\sum_{m_i:|m_{i-1}m_i|>0} P_{ML}(m_i)} \tag{3.9}$$

Le lissage de Katz est proposé pour palier au problème posé par les n-grammes de hautes fréquences.

**d. Lissage par interpolation (Jelinek-Mercer) :** ce type de lissage consiste à combiner le modèle de langue considéré avec un ou plusieurs modèles de références estimés sur d'autres corpus d'apprentissages. Typiquement, dans le cas de collection de documents, on pourrait par exemple estimer le modèle de document en le combinant

avec le modèle de la collection. Dans ce cas, le modèle de document est exprimé ainsi :

$$P_{JM}(m_i|d) = (1-\lambda)P_{ML}(m_i|d) + \lambda P_{ML}(m_i|C) \quad (3.10)$$

Les modèles $P_{ML}(m_i|d)$ et $P_{ML}(m_i|C)$ sont estimés selon le maximum de vraisemblance.

**e. Lissage de Dirichlet :** le lissage précédent ne tient pas compte de la taille des échantillons. pour remédier à cela, le lissage de Dirichlet exploite les valeurs de $\lambda$ (formule (3.10)) en fonction de la taille de l'échantillon. Dans ce cas cette formule s'écrit comme suit:

$$P_{Dir}(m_i|d) = \frac{|d|}{|d|+\mu} P_{ML}(m_i|d) + \frac{\mu}{|d|+\mu} P_{ML}(m_i|C) = \frac{|d|\times P_{ML}(m_i|d)+\mu\times P_{ML}(m_i|C)}{|d|+\mu} = \frac{tf(m_i,d)+\mu P_{ML}(m_i|C)}{|d|+\mu} \quad (3.11)$$

Avec $P_{ML}(m_i|d) = \frac{tf(m_i,d)}{|d|}$

Où $|d|$ est la taille du document (le nombre d'occurrences de mots), $tf(m_i,d)$ est la fréquence du mot $m_i$ dans $d$ et $\mu$ est un paramètre appelé pseudo fréquence.

Plusieurs études en RI ont montré que le choix de la méthode de lissage a un grand impact sur les performances du SRI. Zhai et lafferty **[209]** ont expérimenté plusieurs techniques de lissage en utilisant le modèle de langue uni-gramme et ils ont rapporté que la méthode de lissage de Dirichlet donne de meilleurs résultats que les autres méthodes de lissage. Dans nos travaux nous utilisons cette méthode de lissage.

## 3.3 Modèles de langue et recherche d'information

L'approche de modélisation de langue part d'un principe différent des approches traditionnelles de RI ; on ne tente pas de modéliser directement la notion de pertinence (à l'exception de **[116]**) ; mais on considère que la pertinence d'un document face à une requête est en rapport avec la probabilité que la requête puisse être générée par le modèle de langue d'un document. On suppose en effet qu'à chaque document est associé un modèle de langue, soit $M_d$, le score de pertinence du document vis-à-vis d'une requête $q$ est déterminé par la probabilité de génération de la requête sachant le modèle de ce document, soit $P(q|M_d)$.

La plupart des modèles de langue développés pour la RI se base (utilise) sur ce principe de génération de la requête par un modèle de document, néanmoins, il existe d'autres variantes, qui sont discutées dans la section suivante.

### 3.3.1 Approches d'exploitation des modèles de langue en RI

En se basant sur la représentation des documents et la fonction de « Ranking » les approches de modélisation de langue pour la RI peuvent être classées en trois catégories :

1. Génération de la requête par le modèle de document (Query Likelihood Models) : cette catégorie correspond à ce que nous avons expliqué ci-dessus. Un modèle de langue est associé à chaque document. Les documents sont alors classés selon leurs probabilités de génération de la requête, soit $P(q|M_d)$.
2. Génération de document par le modèle de la requête (Document Likelihood Models) : cette approche procède dans le sens inverse. Ainsi, un modèle de langue est associé à la requête, les documents sont alors classés selon leurs probabilités que leur contenu soit généré par le modèle de la requête, soit $P(d|M_q)$.
3. Similarité document-requête : dans cette catégorie, on considère à chaque document et à chaque requête est associé un modèle de langue. Les documents sont alors ordonnés selon la similarité de leurs modèles avec celui de la requête. Cette similarité est estimée en calculant l'entropie croisée ou la KL (Kullback-Leibler) divergence discuté dans la section 3.3.3.

Nous présentons ci-dessous quelques modèles développés implémentant ces différentes catégories d'approches.

#### 3.3.1.1 Génération de la requête par le modèle de document (Query Likelihood Models)

Ponte et Croft **[151]** furent les premiers à proposer un modèle de langue pour la RI. L'intuition derrière leur modèle est que l'utilisateur est supposé avoir une idée des termes qui sont présents dans le document pertinent ou modèle de ce document $M_d$; à partir de là, l'utilisateur génère la requête en utilisant ces termes. Ainsi, la requête $q$ est censée fournir des indices (des termes) associés au modèle du document (les documents recherchés).

On cherche alors à estimer la probabilité que la requête provienne du modèle ayant généré ce document, soit :

$$score(q,d) = P(q|M_d) \tag{3.12}$$

Ce modèle utilise une distribution de Bernoulli, c'est-à-dire que non seulement les mots présents dans la requête, mais aussi ceux absents de la requête sont pris en compte. Le score du document vis-à-vis de la requête est alors exprimé ainsi :

$$score(q,d) = \prod_{t\in q} P(t|M_d) \times \prod_{t\notin q}\left(1 - P(t|M_d)\right) \tag{3.13}$$

Ce score est composé de deux parties : la probabilité d'observer les termes de la requête dans le document et la probabilité de ne pas observer les termes absents de la requête dans le document.

La probabilité $P(t|M_d)$ est calculée par une méthode non paramétrique qui utilise la probabilité moyenne d'apparition du terme $t$ dans les documents qui le contient ($P_{Avg}(t)$) et un facteur de risque pour un terme observé dans le document ($R(t,d)$). Par contre, la probabilité d'un terme dans la collection est utilisée pour les termes qui n'apparaissent pas dans le document. Le calcul de cette probabilité est exprimé ainsi :

$$P(t|M_d) = \begin{cases} P_{ML}(t|d)^{(1-R(t,d))} \times P_{Avg}(t)^{R(t,d)} & si \quad tf(t,d) > 0 \\ \frac{tf(t,C)}{|C|} & sinon \end{cases} \tag{3.14}$$

Ponte et croft ont obtenu une amélioration de +8.74% de performance (précision moyenne) par rapport au modèle utilisant le schéma de pondération $tf \times idf$ d'Okapi sur des collections TREC.

Dans le modèle proposé par Hiemstra **[94]**, la requête est considérée comme une séquence de termes, $q = (t_1, t_2, \dots, t_n)$. Dans ce modèle on ne considère que les mots présents dans la requête, une distribution multinomial est utilisée. Le score d'un document vis-à-vis d'une requête (la probabilité de générer une requête $q$ sachant le modèle de document $M_d$) est donné par la formule suivante :

$$score(q,d) = P(d)\prod_{i=1}^{n} P(t_i|M_d) \tag{3.15}$$

Où $P(d)$ est la probabilité a priori d'un document.

Pour l'estimation de la probabilité $P(t_i|M_d)$, Hiemstra utilise l'approche par interpolation qui combine le modèle de langue de document ($M_d$) avec le modèle de langue de la collection. Le calcul de cette probabilité est exprimé ainsi :

$$P(t_i|M_d) = \lambda P_{ML}(t_i|M_d) + (1-\lambda)P_{ML}(t_i|C) \quad (3.16)$$

Où $\lambda$ est le poids d'interpolation qui varie entre 0 et 1. Ce paramètre peut être considéré constant ou il peut être estimé d'une manière sophistiquée en utilisant un processus d'optimisation automatique tel que l'algorithme de maximisation d'espérance (EM).
Le calcul des deux probabilités $P_{ML}(t_i|M_d)$ et $P_{ML}(t_i|C)$ est réalisé selon une estimation de vraisemblance maximale, exprimée ainsi :

$$P_{ML}(t_i|M_d) = \frac{tf(t_i,d)}{|d|} \quad \text{et} \quad P_{ML}(t_i|C) = \frac{df(t_i)}{\sum_{t_j \in V} df(t_j)} \quad (3.17)$$

Où $tf(t_i,d)$ est la fréquence du terme $t_i$ dans le document $d$, $df(t_i)$ est le nombre de documents contenant le terme $t_i$ et $V$ est le vocabulaire d'index.

Miller et al **[139]** ont proposé un modèle similaire à celui de Hiemstra. La différence entre les deux modèles réside dans leur implémentation. Miller utilise un modèle de Markov caché à deux états, illustré par la figure 3.1 suivante.

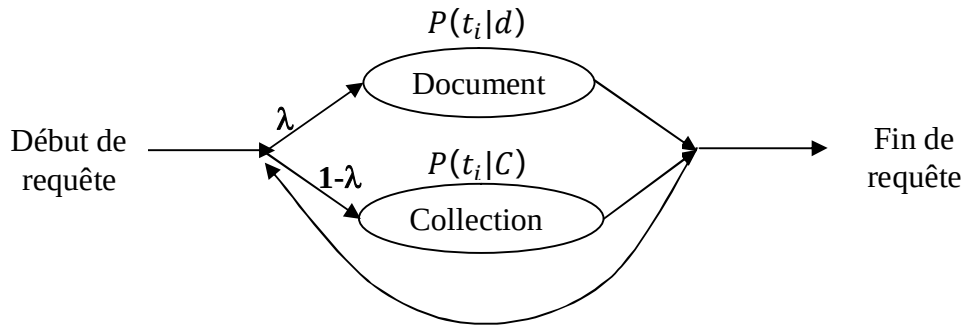



**Figure 3.1 Modèle de Markov à deux états**

Ce modèle correspond à la formule suivante :

$$P(q|M_d) = \prod_{t_i \in q}[\lambda P_{ML}(t_i|M_d) + (1-\lambda)P_{ML}(t_i|C)] \quad (3.18)$$

L'estimation de la probabilité $P(t_i|M_d)$ est réalisée de la même manière que dans le modèle précédent (Hiemstra). Par contre, la probabilité $P(t_i|C)$ est estimée différemment :

$$P_{ML}(t_i|C) = \frac{tf(t_i)}{\sum_{t_j \in V} tf(t_j)} \quad (3.19)$$

Où $tf(t_i)$ est la fréquence du terme $t_i$ dans le corpus et $\sum_{t_j \in V} tf(t_j)$ est le nombre de termes dans le corpus et $V$ est le vocabulaire d'index.

**Similarité avec le modèle $tf \times idf$**

Dans cette section nous présentons le passage de la formule de classement du modèle de langue (formule (3.20)) en une formule similaire au modèle vectoriel $tf \times idf$. Initialement, nous divisons la formulation en deux parties, la première concerne les termes qui apparaissent dans le document et dans la requête ($t \in q \cap d$). La seconde partie contient les autres termes de la requête ($t \in q - d$), c'est adire les termes qui apparaissent dans la requête mais pas dans le document.

$$P(q|M_d) = \prod_{t \in q} (1-\lambda)\, P(t|d) + \lambda P(t|C) \tag{3.20}$$

$$= \prod_{t \in q \cap d} (1-\lambda)\, P(t|d) + \lambda P(t|C) \times \prod_{t \in q-d} \lambda P(t|C)$$

Nous substituons les valeurs des deux probabilités ( $P(t|d)$ et $P(t|C)$) et nous obtenons :

$$P(q|M_d) = \prod_{t \in q \cap d} (1-\lambda)\frac{tf(t,d)}{|d|} + \lambda\frac{tf(t,c)}{|c|} \times \prod_{t \in q-d} \lambda\frac{tf(t,c)}{|c|}$$

$$= \prod_{t \in q \cap d} (1-\lambda)\frac{tf(t,d)}{|d|} + \lambda\frac{tf(t,c)}{|c|} \times \prod_{t \in q-d} \lambda\frac{tf(t,c)}{|c|} \times \prod_{t \in q \cap d} \left( \frac{\lambda\frac{tf(t,c)}{|c|}}{\lambda\frac{tf(t,c)}{|c|}} \right)$$

$$= \prod_{t \in q \cap d} \left( \frac{\lambda\frac{tf(t,c)}{|c|}(1-\lambda)\frac{tf(t,d)}{|d|}}{\lambda\frac{tf(t,c)}{|c|}} + \lambda\frac{tf(t,c)}{|c|} \right) \times \prod_{t \in q \cap d} \frac{1}{\lambda\frac{tf(t,c)}{|c|}} \times \prod_{t \in q-d} \lambda\frac{tf(t,c)}{|c|} \prod_{t \in q \cap d} \lambda\frac{tf(t,c)}{|c|}$$

$$= \prod_{t \in q \cap d} \left( \frac{(1-\lambda)\frac{tf(t,d)}{|d|}}{\lambda\frac{tf(t,c)}{|c|}} + 1 \right) \times \lambda^{|q|} \times \prod_{t \in q} \lambda\frac{tf(t,c)}{|c|}$$

En appliquant le logarithme on obtient :

$$P(q|M_d) = \sum_{t \in q \cap d} \log\left( \frac{1-\lambda}{\lambda} \frac{tf(t,d)}{|d|} \frac{|c|}{tf(t,c)} + 1 \right) + |q| \log \lambda \tag{3.21}$$

A partir de cette formulation, on peut remarquer que le facteur $\frac{tf(t,d)}{|d|}$ agit comme la fréquence du terme ($tf$) et le facteur $\frac{|c|}{tf(t,c)}$ agit comme la fréquence inverse en document ($idf$).

En conclusion, cette catégorie de modèles de langue offre plusieurs possibilités comme :

- Incorporer des informations sur la pertinence a priori du document (la longueur de document, le nombre de liens entrants, le *PageRank*, etc), en utilisant le facteur $P(d)$.
- Incorporer la structure et la sémantique du document.

#### 3.3.1.2 Génération de document à partir du modèle de la requête (Document Likelihood Model)

Au lieu de modéliser la RI comme processus de génération de la requête, Lavrenko et Croft **[116]** ont proposé de modéliser explicitement le modèle de pertinence. Ils ont en effet proposé d'estimer ce modèle à partir du modèle de la requête sans utilisation de données d'entraînement, en faisant le parallèle avec la modélisation de la pertinence proposée dans le modèle probabiliste classique. Ils considèrent en effet que pour chaque requête, il existe un modèle permettant de générer le sujet (thème) abordé par la requête, c'est ce que les auteurs appellent le modèle de pertinence, soit $\theta_R$.

Le but est alors d'estimer la probabilité, $P(t|\theta_R)$, de générer un terme à partir du modèle de pertinence. Comme le modèle de pertinence n'est pas connu, les auteurs ont suggéré d'exploiter les documents retournés les mieux classés (feedback documents) en assumant qu'ils sont générés à partir du modèle de pertinence. Le modèle de pertinence est alors exprimé comme suit :

$$P(t|\theta_R) = \sum_{d \in R} \frac{P(t|d)P(q|d)P(d)}{P(q)} = \sum_{d \in R} P(t|d)P(q|d) \qquad (3.22)$$

Où $R$ dénote l'ensemble de documents feedback.

Ainsi, le modèle de pertinence obtenu est une combinaison pondérée du modèle individuel de chaque document feedback ($P(t|d)$) avec le score de ce document vis-à-vis de la requête ($P(q|d)$).

Les résultats des expérimentations ont montré que cette approche améliore sensiblement les performances de la recherche d'information, de +10% à +29% d'amélioration de précision moyenne par rapport au modèle de langue de base.

#### 3.3.1.3 Comparaison des modèles de requête et du document

Cette approche s'est inspirée de l'approche vectorielle de la RI, dans laquelle la requête et le document sont représentés sous forme de vecteurs, la pertinence est alors interprétée par la similarité des vecteurs associés à la requête et au document.

Lafferty et Zhai **[111]** ont proposé un cadre de minimisation de risque basé sur la théorie de décision bayésienne. Dans ce cadre la requête et les documents sont modélisés par des modèles statistiques. Le modèle de la requête modélise entre autres les préférences et les besoins des utilisateurs, le modèle du document permet de modéliser le processus de génération de document et de capturer les préférences de l'auteur.

La similarité entre documents et requête est mesurée par la fonction de divergence de Kullback-Leibler (*KL)* entre le modèle de la requête ($P(t|q)$) et celui du document ($P(t|d)$) **[111]**, exprimée comme suit :

$$Score(q,d) = -KL(q||d) = -\sum_{t \in V} P(t|q).log\frac{P(t|q)}{P(t|d)} \quad (3.23)$$

Où V est le vocabulaire d'index.

Cette approche peut être vue comme la combinaison de certains aspects des deux approches précédentes. Les expérimentations menées par les auteurs sur des collections TREC ont montré l'utilité de cette méthode. De plus, selon la fonction de risque (perte) choisie, les différents modèles de la RI peuvent s'avérer comme des cas spécifiques de cette approche.

## 3.4 Prise en compte des relations entre termes dans les modèles de langue

La plupart des modèles présentés jusqu'ici, utilisent le modèle uni-gramme (mots simples), justifiée par l'hypothèse d'indépendance entre termes. Il est évident que cette hypothèse est une simplification qui facilite grandement le calcul, néanmoins, elle ne reflète pas la réalité, car les termes dans les documents et la requête sont liés.

Il est assez naturel d'étendre ces modèles (uni-gramme) pour aller au-delà de cette hypothèse d'indépendance des termes ; c'est à dire intégrer les relations potentielles entre termes dans les modèles de langue.

Deux types de relations entre termes peuvent être considérés.

1. Relation de proximité, de surface ou de dépendance entre termes : dans l'objectif de remédier au problème d'ambiguïté des termes.
2. Relation sémantique entre termes : dans le but de pallier au problème de disparité entre termes.

Nous détaillons ci-dessous les travaux ayant été réalisés pour la prise en compte de ces deux types de relations.

### 3.4.1 Prise en compte des relations surfaciques entre termes

Deux directions on été investies pour prendre en compte ce type de relation : la première considère l'utilisation de la dépendance entre termes. Particulièrement, elle suppose que la requête (document) est composée de plusieurs unités de termes (n-grammes, mots composés) et utilise les occurrences de ces unités dans le document en plus des mots simples lors de l'appariement.

La seconde direction se base sur l'utilisation des fonctions de proximité entre termes. Ces fonctions de proximités capturent la mesure dans laquelle les termes de recherche apparaissent proches les uns des autres dans un document. Ces fonctions sont alors utilisées comme un facteur supplémentaire pour le classement des documents.

#### 3.4.1.1 Prise en compte des mots composés dans le modèle de langue

Pour étendre le modèle de langue uni-gramme et aller au delà de l'hypothèse d'indépendance entre termes, plusieurs travaux ont exploité des unités plus complexes formées de n-grammes (mots composés).

Plusieurs travaux en modèle de langue ont exploité les mots composés. Cependant, ils diffèrent dans la manière de leur exploitation et cela selon plusieurs critères ; des critères considérés en RI classique (décrit dans le premier chapitre, section 1.3.1), tels que: la directionnalité (ordre des termes), l'adjacence entre mots simples formant le mot composé, la méthode de sélection des mots composés utilisée (filtrage des n-grammes), et le critère en rapport avec la méthode de lissage utilisée (méthode de combinaison).

Song et Croft **[182]** ont proposé un modèle de langue qui combine le modèle bi-gramme et le modèle uni-gramme en utilisant l'interpolation linéaire. L'estimation du modèle bi-gramme est donnée par :

$$P_{ML}(t_{i-1}, t_i|d) = \frac{f(t_{i-1},t_i|d)}{f(t_{i-1}|d)} \quad (3.24)$$

Où $\lambda$ est un paramètre de lissage et $f(t_{i-1}, t_i|d)$, $f(t_{i-1}|d)$ sont respectivement la fréquence du bi-gramme $(t_{i-1}, t_i)$ et de l'uni-gramme $(t_{i-1})$ dans le document $d$.

Srikanth et Srihari **[185]** ont développé un modèle similaire au modèle de Song et Croft **[182]** où la contrainte d'ordre est ignorée, ce modèle est dit bi-terme. Dans lequel, par exemple, les deux bi-grammes « Acrobat Reader » et «Reader Acrobat » sont représentés par un même bi-terme.

L'estimation des probabilités des bi-termes est réalisée selon l'une des trois approximations suivantes:

$$P_{BT1}(t_{i-1}, t_i|d) \approx \frac{1}{2}\left(P_{ML}(t_{i-1}, t_i|d) + P_{ML}(t_i, t_{i-1}|d)\right) \quad (3.25)$$

Cette première approximation est la moyenne des probabilités des bi-grammes ($t_i t_{i-1}$ et $t_{i-1} t_i$).

$$P_{BT2}(t_{i-1}, t_i|d) \approx \frac{1}{2}\left(\frac{f(t_i,t_{i-1}|d)+f(t_{i-1},t_i|d)}{f(t_{i-1}|d)}\right) \quad (3.26)$$

Cette probabilité est semblable à celle d'un bi-gramme.

$$P_{BT3}(t_{i-1}, t_i|d) \approx \frac{1}{2}\left(\frac{f(t_i,t_{i-1}|d)+f(t_{i-1},t_i|d)}{min(f(t_{i-1}|d)\ ,\ f(t_i|d))}\right) \quad (3.27)$$

Les expérimentations menées avec ces approximations ont montré des gains de performance minimes par rapport au modèle bi-gramme classique ; le meilleur gain de précision est obtenu avec la troisième approximation, il est de l'ordre de +1.93%.

Jiang et al **[101]** ont proposé un modèle de langue pour incorporer des expressions (deux mots) en utilisant la méthode de lissage Backoff. Le modèle proposé est exprimé ainsi :

$$P(t_{i-1}, t_i|d) \approx \begin{cases} P_{dml}(t_{i-1}, t_i|d) & si \quad t_{i-1}t_i \in d \\ \alpha_d P(t_{i-1}|d) \times P(t_i|d) & sinon \end{cases} \quad (3.28)$$

Cette méthode utilise le modèle uni-gramme si l'expression $t_{i-1}t_i$ n'existe pas dans le document. Le facteur $\alpha_d$ est exprimé ainsi :

$$\alpha_d = \frac{1-\sum_{t_{i-1}t_i \in d} P_{dml}(t_{i-1}t_i|d)}{\sum_{t_{i-1}t_i \notin d} P(t_{i-1}|d) \times P(t_i|d)} \quad (3.29)$$

Les résultats obtenus avec ce modèle n'ont pas dépassé ceux de la méthode par interpolation entre bi-gramme et uni-gramme de Song et Croft **[182].**

Alvarez et al **[6]** ont proposé l'incorporation des mots composés sans contraintes d'adjacence ou d'ordre, dans le modèle de langue. La sélection des mots composés (expression) est basée sur les relations lexicales ou statistiques entre termes.
L'estimation de la probabilité des mots composés dans un document est exprimée ainsi :

$$P(t_1t_2|d) = \lambda P_{ML}(t_1t_2|d) + \theta_{t_1t_2}\gamma\big(P(t_1|d) + P(t_2|d)\big) \quad (3.30)$$

Où $\theta_{t_1t_2}$ est égal à 1 si le couple $(t_1t_2)$ n'apparait pas dans le document $d$, 0 sinon et $\gamma$ est un paramètre de normalisation.
Les auteurs proposent deux façons (approches) de combiner les termes individuels (simples) et les mots composés. Dans la première, les mots simples sont utilisés pour le lissage et leurs poids sont identiques dans la description d'un mot composé. Dans la seconde, les mots composés sont considérés comme des unités d'indexation simples est sont par conséquent introduits dans le modèle uni-gramme avec les mots simples.
Les expérimentations menées ont montré que les résultats obtenus avec la première approche ne surpassent pas ni les résultats obtenus avec le modèle vectoriel ni les résultats obtenus avec le modèle uni-gramme ou bi-gramme.
Par contre, les résultats obtenus avec la seconde approche surpassent les meilleurs résultats obtenus avec les modèles uni-gramme et bi-gramme.

Dans leur étude Gao *et al* **[74]**, considèrent une expression (dans la requête ou dans le document) comme un ensemble de termes mais aussi comme un ensemble de liens existants entre ces termes (liens de proximité), qui peuvent être non adjacents. Avec cette interprétation un document pertinent pour une requête doit contenir en plus des termes de la requête les liens existants entre ces termes. Les relations entre termes sont considérées comme des variables cachées qui permettent d'exprimer les dépendances. La génération de la requête est alors un processus en deux étapes. Premièrement, la structure de dépendance $L$ est produite selon une probabilité $P(L|d)$. Puis, la requête $q$ est générée selon la probabilité $P(q|L,d)$, les termes de la requête étant choisis en fonction des termes liés dans $L$. La probabilité de produire la requête $P(q|d)$ sachant toutes les structures de dépendances possibles $Ls$ est alors exprimée ainsi :

$$P(q|d) = \sum_{Ls} P(q,L|d) = log\big(P(L|d)\big) + \sum_{i=1}^{m} P(q_i|d) + \sum_{(i,j)\in L} MI\big(q_i,q_j|L,d\big) \quad (3.31)$$

Où $MI\big(q_i,q_j|L,d\big)$ est l'information mutuelle du couple $(q_i,q_j)$.

Les auteurs supposent que l'ensemble des structures possibles $Ls$ est dominé par une unique structure.

Le modèle ainsi décrit est évalué sur des collections TREC. Les résultats obtenus sont meilleurs que ceux du modèle probabiliste et du modèle uni-gramme. Cependant, la prise en compte des dépendances entre les termes de cette manière est difficile à estimer.

Metzler et Croft **[137]** ont élaboré un cadre formel pour la modélisation des dépendances entre termes en utilisant les champs de Markov, nommé (MRF). Une structure de graphe non orienté est utilisée pour modéliser les distributions jointes. Dans ce cadre, ils ont proposé de modéliser deux types de dépendance: dépendance séquentielle (SD : Sequential Dependency), capturant les relations entre les paires de termes adjacents de la requête, et la dépendance complète (FD : Full Dependency), capturant les relations entre toutes les paires de termes de la requête. Ces deux modèles de dépendance ont été interpolés linéairement avec un modèle uni-gramme, selon la formule suivante:

$$Score_{MRF}(q,d) = \lambda_u \sum_{t_i \epsilon q} log\big(P(q_i|d)\big) + \lambda_s \sum_{t_i \epsilon q} \sum_{\substack{t_i \epsilon q \\ j=i+1}} log\Big(P\big(\langle q_i, q_j\rangle_w|d\big)\Big) + \lambda_f \sum_{t_i \epsilon q} \sum_{\substack{t_i \epsilon q \\ j \neq i}} log\Big(P\big(\langle q_i, q_j\rangle_w|d\big)\Big) \quad (3.32)$$

Où les paramètres $\lambda_u$, $\lambda_s$ et $\lambda_f$ permettent de contrôler respectivement, le poids du modèle uni-gramme, du modèle de dépendance séquentielle et du modèle de dépendance complète et le paramètre $w$ définit la longueur de la fenêtre du texte permettant de compter les occurrences de la paire $\langle q_i, q_j\rangle$ dans le document $d$.

#### 3.4.1.2 Modèles de langue basés sur la proximité entre termes

L'intuition considérée dans ces modèles est la suivante : « un bon document est celui dans lequel les termes de la requête apparaissent proches les uns des autres». Par exemple, considérant la requête « ***moteur de recherche*** » et les deux documents suivants qui contiennent les termes de la requête:

> $d_1$ :« *un* ***moteur de recherche*** *est une application web permettant de retrouver des ressources…*».
>
> $d_2$ : « *le développement* d'*un* ***moteur*** *utilisant uniquement l'énergie solaire nécessite* ***de*** *la* ***recherche*** *…*».

Intuitivement le document $d_1$ est plus approprié à la requête car les termes de la requête y apparaissent d'une manière contiguë dans ce document. Par contre, le document $d_2$ contient les termes de la requête éloignés les uns des autres, et leur combinaison ne fait pas référence au sens du terme recherché « ***moteur de recherche*** ».

Les approches ayant exploité cette proximité se distinguent de la manière dont cette dernière est modélisée et intégrée dans le modèle de langue. Elle est estimée par des mesures de proximité dans [**143**] **[189]**, puis intégrée d'une manière externe (combinaison simple) dans **[189]** et d'une manière interne dans **[212]**. Dans **[129]** la proximité est introduite différemment dans le modèle de langue, les auteurs définissent un modèle de langue à chaque position dans le document, en créant un document virtuel basé sur la propagation de termes. Dans ce qui suit nous détaillons un peu plus ces modèles.

Tao et Zhai **[189]** ont utilisé cinq différentes fonctions de mesure de proximité, elles sont scindées en deux types :

- Les mesures de proximité basées sur le passage du document : deux mesures sont définies. La mesure dite « span », définie comme étant la taille du plus petit passage de document qui contient toutes les occurrences des termes de la requête, incluant les occurrences répétées. La seconde mesure dite « MinCover », définie comme étant la taille du plus petit passage de document qui contient chaque terme de la requête au moins une fois.
- Les mesures de proximité basées sur l'agrégation de distances : cette proximité est calculée en deux étapes. Premièrement, les distances entre les occurrences individuelles des termes de la requête sont calculées. Ensuite, une agrégation de ces distances est effectuée en utilisant l'une des trois mesures d'agrégation suivantes :

« MinDist » définie comme étant la plus petite distance entre toutes les paires des termes de la requête présents dans le document, « AveDist » définie comme étant la distance moyenne entre toutes les paires des termes de la requête présents dans le document et « MaxDist » définie comme étant la plus grande distance entre toutes les paires des termes de la requête présents dans le document.

Le score de proximité obtenu par ces mesures est ensuite intégré dans le modèle de langue, utilisant la mesure *KLD*, de la manière suivante :

$$score(q,d) = KLD(q,d) + \log\left(\gamma + exp\left(-\delta(q,d)\right)\right) \quad (3.33)$$

Où $KLD(q,d)$ est le score du document $d$ vis-à vis de la requête $q$ obtenu avec la fonction de « Ranking » *KLD* et $\delta(q,d)$ est une mesure de distance de proximité du document $d$ vis-à vis de la requête $q$.

Leur modèle a été expérimenté en utilisant cinq collections de test TREC. Les résultats obtenus, montrent que les mesures de proximité basées sur le passage du document ne permettent pas d'obtenir une amélioration par rapport aux modèles ne considérant pas la notion de proximité. Par contre, les mesures basées sur l'agrégation de distances permettent d'obtenir des améliorations dans la plupart des cas. Ils ont constaté aussi que la mesure « MinDist » donne de meilleures performances par rapport aux autres mesures.

A la différence de l'approche proposée par Tao et Zhai **[189]**, Na et al **[143]** ont proposé un modèle de langue bi-gramme qui exploite la contrainte d'adjacence, c'est-à-dire le calcul de la distance n'est pas réalisé pour tous les couples de termes de la requête, mais seulement pour les termes qui apparaissent d'une manière adjacente dans la requête. Les résultats rapportés montrent que l'utilisation de la notion de proximité sur des modèles bi-grammes permet d'améliorer les performances comparativement au modèle bi-gramme antérieur **[182]** et au modèle proposé par Tao et Zhai **[189]**.

Zhao et Yun **[212]** ont proposé un modèle nommé « Proximity Language Model », similaire au modèle de Tao et Zhai **[189]**. Cependant, l'intégration du score de proximité est réalisée de manière interne. Plus spécifiquement, la proximité dans ce modèle est considérée comme un hyper paramètre du modèle de Dirichlet. Cet hyper paramètre permet de pondérer les différents paramètres du modèle de langue uni-gramme. Les auteurs ont rapporté qu'il offre des améliorations par rapport au modèle uni-gramme et au modèle de Tao et Zhai **[189]**.

Lv and Zhai **[129]** ont proposé un modèle de langue nommé « Positional Language Model : PLM ». Dans ce dernier, un modèle de langue pour chaque position du document est créé, il est exprimé ainsi :

$$P(t|d,i) = \frac{c\prime(t;i)}{\sum_{t\prime \in V} c\prime(t\prime;i)} \quad (3.34)$$

Où V est le vocabulaire et $c'(t;\ i)$ est la fréquence virtuelle du terme $t$ à la position $i$ obtenue par la propagation de l'ensemble des occurrences du terme $t$ dans toutes les positions du document. cette propagation est réalisé en utilisant des fonction de densité (Gaussian kernel, Trangle kernel, Circle kernel, Hamming kernel).

Afin de calculer le score d'un document vis-à-vis d'une requête, ils utilisent les scores obtenus sur les différentes positions du document. Différentes stratégies de combinaison de scores ont été utilisées : la stratégie de la meilleure position, la stratégie multi-position et la stratégie Multi-σ.

Le modèle ainsi défini a été expérimenté sur plusieurs collections de test TREC. Les auteurs ont rapporté que ce modèle améliore les résultats du modèle de Tao et Zhai **[189]**.

Les approches utilisant la notion de proximité pour intégrer les relations entre termes ont montré que cette source d'information est utile pour la RI. Cependant, le principal problème de ces approches est l'absence d'une mesure bien reconnue pour le calcul de la proximité entre termes **[212]**.

### 3.4.2 Prise en compte des relations sémantiques entre termes

La prise en compte des relations sémantiques entre termes a été largement étudiée dans le domaine de la RI, depuis ses débuts.

Dans le cadre de modèle de langue, la prise en compte de ces relations est réalisée soit en modifiant la représentation (modèle) de la requête en ajoutant les termes liés, cette opération est nommée expansion du modèle de la requête, ou inversement, en modifiant le modèle du document en attribuant une plus grande probabilité aux termes liés aux termes du document plutôt qu'aux termes non liés. Cette opération est nommée expansion du modèle de document. Les deux types d'expansion peuvent être exploités conjointement.

La fonction de « Ranking » diffère selon l'approche choisie, dans l'expansion du modèle de document le principe de génération de la requête est utilisé **[94] [139]**. Dans les autres

approches (expansion du modèle de la requête et la combinaison des deux types d'expansion) c'est la fonction de divergence (*KLD*) qui est utilisée **[111]**.

#### 3.4.2.1 Expansion du modèle de document

Une bonne estimation du modèle de document améliore les performances de la recherche d'information. En effet, le lissage du modèle de document a donné des résultats prometteurs **[209]**.

La majorité des approches de lissage du modèle de document utilisent le modèle de la collection **[94] [106] [111] [139]**. Cependant, l'utilisation du modèle de la collection pose un problème, à savoir : la quantité de probabilité attribuée à un terme n'apparaissant pas dans un document ne dépend pas de liens de ce terme avec les termes du document mais plutôt de statistiques sur le terme (fréquence du terme dans la collection).

Pour remédier à ce problème des techniques de lissage « sémantique » sont développées. Chacune d'elles utilise une source de données contextuelle. Autrement dit, elles affectent des probabilités plus grandes pour les termes reliés aux termes du document.

Nous présentons ci-dessous quelques travaux réalisés dans ce contexte de lissage « sémantique ».

Liu et Croft **[125]** ont proposé un modèle de langue incorporant la notion d'agglomération (cluster) de documents, qui s'appuie sur l'hypothèse préconisant que les documents similaires répondent aux mêmes besoins d'informations. Chaque cluster est considéré comme un grand document qui traite un thème (sujet) donné ; cette source (cluster) est utilisée pour étendre le modèle du document de telle sorte que les termes présents dans les documents de même cluster que le document concerné auront une plus grande probabilité. Ce modèle est formulé ainsi :

$$P(t|d) = \lambda_1 P(t|d) + \lambda_2 P(t|cluster) + \lambda_3 P(t|C) \qquad (3.35)$$

Où $\lambda_1 + \lambda_2 + \lambda_3 = 1$, $P(t|d)$ est le modèle uni-gramme du document, $P(t|cluster)$ est le modèle du cluster et $P(t|C)$ est le modèle de la collection.

Cette idée a été reprise et étendue par Tao *et al* dans **[188]**, en construisant un cluster pour chaque document, ce cluster contient le voisinage du document, sachant que les documents n'ont pas la même importance dans ce voisinage.

Ces deux approches ont donné des améliorations significatives sur des collections de test TREC par rapport au modèle uni-gramme. Néanmoins, ces approches souffrent d'un inconvénient majeur, qui considère que chaque document traite un seul thème. Pour palier ce problème, Wei et Croft **[201]** ont proposé un modèle de langue basé sur les modèles LDA (Latent Dirichlet Allocation), qui permet de modéliser un document comme une mixture de thèmes (sujet). La formule générale de ce modèle est exprimée ainsi :

$$P(t|d) = \lambda_1 P(t|d) + \lambda_2 P(t|c) + \lambda_3 P_{LDA}(t|d) \quad (3.36)$$

Où $\lambda_1 + \lambda_2 + \lambda_3 = 1$, $P(t|d)$ est le modèle uni-gramme du document, $P(t|c)$ est le modèle de la collection et $P_{LDA}(t|d)$ est le modèle basé *LDA*, qui permet de représenter un document par un ensemble de thèmes.
Les expérimentations menées avec ce modèle ont montré que ce dernier améliore sensiblement les performances de la RI.

Une autre approche de lissage sémantique a été proposée par Berger et Lafferty **[18]**. Cette approche est basée sur un processus de traduction statistique. La traduction est réalisée entre les termes de la requête $q_i$ et les termes de document $m$ dans le but d'étendre le modèle du document. La probabilité de génération de la requête par un modèle du document est exprimée ainsi :

$$P(q|d) = \prod_{i=1}^{n} \lambda P(q_i|d) + (1-\lambda) \sum_{m \in d} t(q_i|m)\, P(m|d) \quad (3.37)$$

Où $P(q_i|d)$ est le modèle uni-gramme et $t(q_i|m)$ est la probabilité de traduction ou le poids de lien entre les termes $m$ et $q_i$. Cette probabilité est calculée en se basant sur une collection artificielle de données d'entraînement. Ce qui est considéré comme une limite à cette méthode. De plus, le modèle de traduction est peu efficace car tous les termes du document sont concernés par la traduction de chaque terme de la requête.

Cao et al **[33]** ont étendu cette méthode, où deux sources sont utilisées pour estimer la probabilité de traduction $t(q_i|m)$. La première est le thésaurus WordNet, dans le but de réduire le bruit et la deuxième est la relation de cooccurrence entre les termes dans la collection, dans le but de réduire le silence. La probabilité de génération de la requête par le modèle du document est exprimée ainsi :

$$P(q_i|d) = \lambda_1 P_U(q_i|d) + \lambda_2 \sum_{m \in d} P_{Coo}(q_i|m)\, P(m|d) + \lambda_3 \sum_{m \in d} P_L(q_i|m)\, P(m|d) \quad (3.38)$$

Où $\lambda_1 + \lambda_2 + \lambda_3 = 1$ , $P_U(q_i|d)$ est le modèle uni-gramme, $P_{Coo}(q_i|d)$ définit la probabilité du terme $q_i$ dans le document $d$ en utilisant la relation de cooccurrence et $P_L(q_i|m)$ définit la probabilité du terme $q_i$ dans le document $d$ en considérant les relations définies dans WordNet.

Le modèle ainsi proposé a été expérimenté et comparé au modèle uni-gramme sur trois collections de test TREC. Les résultats obtenus montrent que ce modèle permet d'avoir des améliorations significatives par rapport au modèle uni-gramme.

#### 3.4.2.2 Expansion du modèle de la requête

Généralement, l'utilisateur ne formule pas son besoin en information d'une manière exacte (requêtes courtes et/ou l'utilisateur ne fournit pas de bons termes). Par conséquent, les performances des systèmes de recherche d'information sont relativement dégradées. Pour prendre en compte cette difficulté, des techniques de reformulation de la requête sont utilisées, afin d'obtenir des requêtes potentiellement meilleures. La modification de la requête peut être : l'ajout de nouveaux termes et/ou la ré-estimation de l'importance des termes de la requête.

Afin de sélectionner les termes d'expansion, plusieurs méthodes et techniques ont été utilisées **[36]**: la relation de cooccurrence **[12][153]**, la relation de cooccurrence et le mécanisme d'inférence « Information Flow » **[12]**, les règles d'association **[77]**, la mesure d'information mutuelle « Mutual Information » **[95]**, la classification de la requête et l'algorithme EM **[11]**, la classification des terme **[32]**, l'estimation maximale **[136]**, les chaines de Markov **[45]**, le modèle de pertinence **[116] [119] [207]**, le modèle mixte **[210]** et les fonctions basées sur l'analyse de la distribution des termes dans les documents pseudo-pertinents, telles que : la distance Kullback-Leibler (*KLD*) **[35]** et Robertson Selection Value (*RSV*) **[159]**.

Nous présentons ci-dessous les méthodes, les plus en vue, d'expansion de la requête dans le cadre du modèle de langue.

Lavrenko et Croft **[116]** ont proposé un modèle de langue de pertinence, décrit dans la section 3.3.2. Dans la même optique, Zhai et Lafferty **[210]** ont proposé un modèle nommé model-based feedback, où le nouveau modèle de la requête est obtenu par l'interpolation du modèle original de la requête avec un modèle de matière de la requête $\theta_T$ (Topic model), obtenu en

utilisant les documents les mieux classés (retour de pertinence). La construction du modèle $\theta_T$ consiste en l'extraction d'une partie des documents retournés pertinents qui est distincte de l'ensemble des documents de la collection. Comme les documents les mieux classés sont susceptibles de contenir à la fois des informations pertinentes et génériques (ou même non pertinentes), ils peuvent être représentés par un modèle génératif mixte qui combine le modèle $\theta_T$ (à estimer) et le modèle de langue de la collection. Le logarithme de la probabilité des documents les mieux classées est donnée comme suit :

$$log\, P(R|\theta_T) = \sum_{d\in R} \sum_t c(t,d)\, log\big((1-\lambda)p(t|\theta_T) + \lambda p(t|C)\big) \qquad (3.39)$$

Où $R$ est l'ensemble des documents les mieux classés, $c(t,d)$ est le nombre d'occurrences du terme $t$ dans le document $d$, et λ est le poids d'interpolation.
L'algorithme EM (Expectation-Maximization) est ensuite utilisé pour extraire le modèle $\theta_T$.

Wei et Croft **[201]** ont proposé une approche basée sur l'utilisation d'un modèle de matière (Topic) (vu comme contexte idéal de l'utilisateur) construit manuellement en utilisant la ressource ODP (Une ontologie ouverte construite par une communauté de volontaires).
La formule de l'expansion de la requête est exprimée ainsi :

$$P(t|q) = \lambda P'(t|q) + (1-\lambda)P'(t|u) \qquad (3.40)$$

Où $P'(t|q)$ est la probabilité du terme $t$ sachant le modèle de la requête initiale et $P'(t|u)$ est la probabilité du terme $t$ sachant le modèle de la matière.
Les expérimentations menées avec cette approche ont montrés des améliorations non significatives comparées aux résultats obtenus par le modèle de pertinence **[116]**.

Bai et al **[12]** ont proposé une approche pour l'expansion de la requête, basée sur deux modèles. L'un est dit HAL (Hyperspace Analog to Language) qui permet de représenter les termes sous forme de vecteurs pondérés dont les dimensions sont les termes de vocabulaire et le poids représente l'importance de la relation de cooccurrence entre termes (qui est inversement liée à leurs distances dans une fenêtre de longueur donnée) ; pour les termes composés leurs vecteurs respectifs sont combinés en appliquant certaines heuristiques.
Sur la base de cette représentation, les auteurs ont utilisé un autre mécanisme d'inférence, « Information Flow », qui permet de suggérer à partir d'un ensemble de termes un autre

terme. L'expansion de la requête dans ce modèle se fait en tenant compte de toute la requête. Le même principe a été adopté, dans des approches non centrées modèle de langue **[16] [153]**.

Cependant, tous les modèles d'expansion de la requête présentés jusqu'ici ont tendance à ignorer une évidence importante qui est la dépendance entre les termes de la requête (ou document).
En effet, Il existe relativement peu de travaux où l'expansion de la requête a été réalisée en se basant sur un modèle considérant les relations de dépendance ou de proximité entre les termes **[113] [127] [136] [138][80]**.

Metzler et Croft **[136]** ont développé un modèle nommé «Latent Concept Expansion : LCE » basé sur le modèle MRF **[137]** (décrit dans la section 3.4.1.1). Le modèle LCE permet de retrouver les concepts non exprimés par la requête d'un utilisateur (les concepts latents), en se basant sur leurs cooccurrences dans les documents pertinents ou pseudo-pertinents, avec les concepts explicitement exprimés dans la requête originale.
Spécifiquement, le modèle LCE ajoute au graphe MRF, contenant initialement les termes de la requête et le nœud document, des nœuds correspondant aux concepts latents. En utilisant le graphe ainsi construit, la distribution jointe au travers de la requête ($q$), d'un concept latent ($e$), et d'un document ($d$) peut être définie. La probabilité conditionnelle d'un concept latent sachant une requête peut être estimée ainsi :

$$P(e|q) = \frac{\sum_{d \in R_q} P(q,e,d)}{\sum_{d \in R_q} \sum_{e} P(q,e,d)} \tag{3.41}$$

Où $P(q,e,d)$ est la distribution jointe et $R_q$ l'ensemble des documents pertinents ou pseudo-pertinents pour la requête $q$. Une fois le calcul des probabilités conditionnelles effectué, les $k$ concepts latents $e$, ayant les plus grandes valeurs de probabilités conditionnelles sont retenus pour l'expansion de la requête.
Cependant, le modèle LCE suppose que les termes de la requête et les concepts latents sont conditionnellement indépendants sachant le document. Lang *et al* **[113]** ont étendu le modèle LCE pour combler cette limite.

Lv and Zhai **[127]** ont utilisé leur modèle « Positional Language Model » proposé dans **[129]** pour étendre le modèle de pertinence **[116]**. Plus explicitement ils ont intégré la notion de

proximité dans le modèle de pertinence, et cela en surpondérant les termes candidats se trouvant à proximité des termes de la requête.

Ce qui est à noter à propos de l'expansion de la requête dans toutes les approches présentées dans le contexte du modèle de langue (et en général dans la littérature de la RI) est que des améliorations conséquentes des performances de la RI sont obtenues. Afin d'améliorer la robustesse de ces techniques, plusieurs directions de recherche ont été investies, parmi elles on peut citer : l'expansion de requête sélective en utilisant par exemple la mesure de clarté **[53] [207]** et la combinaison d'évidences **[11]**.

### 3.4.3 Positionnement de nos approches

La prise en compte des relations (surfaciques et sémantiques) entre termes est importante pour la RI.

Pour notre part, nous exploitons la relation de cooccurrence entre termes, car cette relation est une source d'information qui ne nécessite pas de ressource externe et elle a montré son efficacité dans beaucoup de travaux **[4] [12] [153].**

Nous avons exploité cette relation pour répondre à différents objectifs.

- Le premier, pour identifier et extraire les mots composés (bi-grammes qui apparaissent fréquemment dans la collection), dont les mots composants sont adjacents, par exemple, « g*énie logiciel* » et « *microsoft word* ». Ces termes seront utilisés comme unités d'indexation. L'utilisation de ces unités d'indexation supplémentaires est dictée par le fait qu'elles sont moins ambiguës et plus précises que les mots qui les composent. De ce fait, le contenu sémantique des documents et des requêtes sera plus précis, ce qui améliore par conséquent la pertinence de la RI. Nous exposerons en détails dans le chapitre 4 la solution proposée.
- Le second, pour l'expansion de la requête. Il consiste en l'extraction des termes de voisinage d'un terme donné (simple ou composé) et les utiliser pour étendre la requête initiale. Par exemple, pour le terme « g*énie logiciel* », les termes probables de son voisinage sont : (*uml, cycle de vie, merise*). Cette deuxième utilisation de la relation de cooccurrence est importante car elle permet de désambiguïser les termes du fait qu'elle permet de capturer la thématique (sémantique) de la requête. Le chapitre 5 est consacré à la présentation de notre méthode d'expansion de la requête.

## 3.5 Incorporation d'évidences indépendantes du contenu de document dans le modèle de langue

Un document web n'est pas seulement décrit par son contenu textuel, mais aussi par des évidences (sources d'information) indépendantes du contenu de document, telles que la structure interne du document (balise), la structure des liens, etc.

Le modèle de langue permet de combiner ces différentes évidences sur un document, suivant la formule ci-dessous :

$$P(q|d) = P(d) \sum_{i=1}^{k} P(q_i|d) \quad (3.42)$$

La pertinence d'un document $d$ vis-à-vis d'une requête $q$ résulte alors de l'évaluation de la combinaison de deux types d'évidences :

- Évidences indépendantes du contenu de document, représentées par le facteur $P(d)$.
- Évidences dépendantes du contenu de document, représentées par la probabilité $P(q_i|d)$.

Nous avons vu dans la section 3.4 les différentes approches proposées permettant d'intégrer la dimension sémantique d'un document afin d'estimer le modèle du document, $P(q_i|d)$. D'autres travaux ont intégré la structure du document dans l'estimation de ce même modèle **[106] [146]**.

Le facteur $P(d)$ représente la probabilité de pertinence a priori du document $d$. Ainsi, selon l'approche adoptée, les propriétés (taille de document, nombre de liens entrants, etc.) indépendantes du contenu de document peuvent être utilisées ou pas pour conditionner cette probabilité. Si cette probabilité n'est pas conditionnée par l'une de ces propriétés (évidences) alors les documents sont équiprobables dans la collection, donc la probabilité a priori de pertinence de document peut être ignorée lors du classement des documents.

Par contre, si la probabilité a priori est conditionnée par l'une de ces caractéristiques, alors les documents de la collection n'ont pas la même probabilité a priori. Par exemple, si la caractéristique utilisée est le score de popularité du document alors un document populaire est plus probable d'être pertinent qu'un document moins populaire.

Plusieurs caractéristiques on étés utilisées pour estimer la probabilité a priori d'un document, comme : la longueur du document **[106]**, la structures des liens **[106] [90]**, le facteur temps **[62] [121]**, le rapport information/bruit **[214]**.

Nous présentons ci-dessous quelques unes des caractéristiques utilisées.

- **La taille du document** :

  Dans ce cas, la probabilité de pertinence a priori est proportionnelle à la taille du document, exprimée ainsi:

$$P(d) = \frac{|d|}{|C|} \tag{3.43}$$

  Où $|d|$ est la taille du document et $|C|$ est la taille de la collection.

  L'intuition de l'utilisation d'une telle caractéristique est qu'un document plus long tend à contenir plus d'informations et par conséquent, il est plus probable d'être pertinent. Les résultats obtenus avec l'utilisation de cette caractéristique ont été mixtes et cela selon la collection utilisée **[90] [106]**.

  Parapar et al **[149]** ont proposé d'estimer cette probabilité $P(d)$ en utilisant la taille compressée d'un document. La formule utilisée est exprimée ainsi :

$$P(d) = \frac{comp(d)}{\sum_{d_i \in C} comp(d_i)} \tag{3.44}$$

  Où $comp(d)$ est la taille en octets du document $d$ compressé (zippé) divisée sur la taille originale en octets du document $d$. Ce nouveau facteur a été évalué et comparé au facteur taille originale du document en utilisant quatre collections TREC. Les résultats présentés montrent que la taille compressée d'un document donne des améliorations de précision (MAP) allant de +0,4% à +3,1% par rapport à l'utilisation de la taille originale du document.

- **La date de création du document :**

  D'autres travaux utilisent l'intuition suivante : « les documents récents tendent à être plus pertinents que les documents anciens », pour estimer la probabilité a priori d'un document.

  Li et Croft **[121]** ont proposé un modèle de langue qui permet d'intégrer la notion de « temps » dans l'évaluation de la pertinence d'un document vis-à vis d'une requête, où ils assignent une plus grande probabilité de pertinence pour les documents ayant une date de création récente. Ainsi, ils expriment la probabilité de pertinence a priori d'un document sachant sa date de création, comme une distribution exponentielle, exprimée comme suit:

$$P(d|T_d) = \lambda e^{-\lambda(T_C - T_d)} \quad (3.45)$$

Où $T_C$ est la date la plus récente dans toute la collection (exprimée en mois) et $T_d$ est la date de création du document $d$.

Les évaluations réalisées sur un ensemble particulier de requêtes montrent que l'incorporation de la notion de temps en utilisant la distribution exponentielle est bénéfique pour la RI.

- **La structure des liens :**

  L'intuition derrière l'utilisation de la structure des liens est que les documents populaires ou les plus cités tendent à être plus pertinents. La méthode la plus simple d'exploitation de la structure des liens est l'utilisation du nombre de liens entrants. La probabilité de pertinence a priori est alors exprimée comme suit :

$$P(d) = \frac{n(l,d)}{\sum_{d_i} n(l,d_i)} \quad (3.46)$$

  Où $n(l,d)$ est le nombre de liens entrants dans le document $d$.

  D'autres facteurs plus sophistiqués peuvent être utilisés comme : le *HITS*, le *PageRank*, etc.

- **Le rapport information/bruit :**

  Il est définit comme le rapport entre la taille du document après prétraitement (élimination des mots vides et des balises HTML) et la taille du document sans prétraitement **[214]**. La probabilité de pertinence a priori est alors exprimée ainsi :

$$P(d) = \frac{l_{token}}{l_{document}} \quad (3.47)$$

  Où $l_{token}$ est la taille du document après prétraitement et $l_{document}$ est la taille du document avant le prétraitement. Ainsi, un document avec moins de mots vides et peu de balises HTML produit un haut rapport information/ bruit, ce qui signifie que le document est de « bonne » qualité.

- **Type d'URL du document :**

  Kraaij et al **[106]** ont utilisé la forme (type) de l'URL pour estimer la probabilité qu'une page soit une page d'entrée, elle définie ainsi :

$$P(d) = P\left(PE|url_{type}(d)\right) = \frac{c(PE,t_i)}{c(t_i)} \quad (3.48)$$

  Où $url_{type}$ est le type de l'URL du document $d$, $c(PE, t_i)$ est le nombre de documents de type d'URL « $t_i$» qui sont des pages d'entrée « *PE* » pour un site web (obtenu a partir des évaluations de pertinence) et $c(t_i)$ est le nombre de documents de type d'URL « $t_i$»

  Quatre types de catégories d'URL ont été définis :

  *Racine* : contenant le nom du domaine seulement, exemple : www.sigir.org

  *Sous-racine* : contenant le nom du domaine suivi d'un seul répertoire, exemple : www.sigir.org/sigirlist/

  *Chemin* (répertoire) : contenant le nom du domaine suivi d'un ou plusieurs répertoires, exemple: www.sigir.org/sigirlist/issues/

  *Fichier* : tous les URL se terminant avec un nom de fichier autre que « index.html ».

  Sur la base de ces quatre types d'URL, ils ont mené des expérimentations sur la collection WT10g (collection utilisée dans TREC 2001), pour estimer la probabilité qu'une page soit une page d'entrée sachant son type d'URL. Ils ont constaté que cette source d'information est un bon indicateur pour prévoir la pertinence d'une page.

## 3.6 Conclusion

Nous avons présenté dans ce chapitre un état de l'art sur les modèles de langue. Nous avons particulièrement, mis l'accent sur les travaux incorporant d'une part les relations entre termes dans ces modèles, d'autre part les évidences indépendantes du contenu de document.

Deux types de relation entre termes ont été exploités, les relations surfaciques (proximité) et les relations sémantiques.

La prise en compte du premier type de relation a pour objectif d'établir une bonne représentation (modèle) du document, en utilisant des unités plus complexes telles que les n-grammes à coté des mots simples (uni-grammes) ou par l'utilisation de la proximité entre les termes de la requête. Nous nous intéressons dans le cadre de notre travail à l'utilisation des mots composés, qui sont des n-grammes spécifiques. Nous présentons dans le chapitre 4 les

éléments fondamentaux de l'exploitation de ces mots composés à savoir leurs identifications, combinaisons avec les mots simples et le schéma de pondération adopté.
L'exploitation du second type de relation consiste quant à elle, à ré-estimer le modèle de la requête ou/et du document afin d'ajouter des termes à la représentation initiale. Cette opération se nomme expansion du modèle de la requête ou/et du document. Dans le chapitre 5 nous exposons notre approche d'expansion de la requête, basée sur un modèle combinant les mots simples et les mots composés (présenté dans le chapitre 4).

Plusieurs caractéristiques (évidences) d'un document, indépendantes du contenu de document ont été utilisées pour estimer la probabilité a priori d'un document. Cependant, toutes les caractéristiques utilisées jusqu'ici dépendent du document uniquement. Or, dans le contexte du web, un document (page web) fait partie en général d'un site lequel fait partie du web. Nous décrivons dans le chapitre 6, notre approche prenant en compte cette réalité.

# Partie II

# Contributions

# Chapitre 4

# Modèle de langue mixte pour la RI

## 4.1 Introduction

La majorité des modèles de RI se basent sur l'hypothèse d'indépendance des mots, ce qui mène au problème d'ambiguïté. Même si, comme nous l'avons étudié dans le chapitre précédent, des travaux ont tenté d'aller au-delà de cette hypothèse, en prenant par exemple des mots composés. À notre connaissance, force est de constater qu'aujourd'hui, il n'existe aucun cadre théorique permettant de justifier et de démontrer de manière claire l'intérêt des mots composés. Une des raisons, largement discutée dans le domaine est l'absence de modèle de pondération ou de mesure de fréquence adéquate pour ce type de terme. Des mesures de pondération ont été proposées dans des approches en dehors du modèle de langue; ces mesures se basent généralement sur un simple comptage de mots, tel que réalisés pour les mots simples. Cependant, ces méthodes sont sans réel apport. Pour prendre en compte la spécificité d'un mot composé nous avons proposé une nouvelle approche de pondération qui intègre la notion de dominance entre les mots composants le mot composé. Plus spécifiquement la fréquence d'un mot composé est revisitée en prenant en compte à la fois le nombre d'occurrence de ce mot composé dans le document ainsi que le nombre d'occurrence de ses mots composants relativement à leur dominance dans ce mot composé.

Cette nouvelle approche de pondération de mot composé est modélisée dans le cadre d'un modèle de langue mixte combinant les mots simples et les mots composés. En plus de cette nouvelle méthode de pondération de mots composés, notre modèle se distingue des modèles existants par les points suivants:

1. La majorité des modèles de langue décrits dans le chapitre précédent [**137**] [**139**] [**182**] [**185**]; prennent en compte toutes les dépendances entre mots adjacents (bi-grammes). Cependant, nous considérons, pour notre part, que seules quelques dépendances sont utiles en RI. Dans notre modèle, nous ne considérons que les bi-grammes « pertinents » (mots composés) comme il a été réalisé dans [**74**]. Toutefois,

le choix des mots composés dans [**74**] nécessite le calcul d'une structure de lien au moment d'appariement document-requête, ce qui augmente le temps de réponse. Dans notre cas, la sélection des bi-grammes « pertinents » se fait à la phase d'indexation, ce qui n'affecte pas le temps de réponse.

2. Notre approche diffère des autres approches dans la méthode utilisée pour l'estimation des modèles de langue des mots simples et des mots composés. Alors que les approches de l'état de l'art estiment les deux modèles d'une façon indépendante. Nous proposons dans notre approche d'estimer le modèle de document des mots composés (voir mots simples) comme une mixture des deux modèles : le modèle de document des mots simples et le modèle de document des mots composés.

Le reste de ce chapitre est organisé comme suit: la section 4.2 présente notre modèle de langue combinant les mots simples et les mots composés. Cette section est divisée en quatre sous-sections. La sous-section 4.2.1 présente la méthode d'estimation du modèle de langue mixte. Dans la sous-section 4.2.2 le concept de « dominance » entre mots est discuté et formalisé. La sous-section 4.2.3 est dédiée à la présentation du schéma de pondération des mots composés, basé sur la notion de dominance. Dans la section 4.2.4 l'estimation d'une composante (probabilité d'un mot simple sachant le modèle du document des mots composés) du modèle de langue mixte est détaillée. Dans la section 4.3 nous rapportons les résultats expérimentaux de notre modèle. La dernière section fait la synthèse de ce chapitre.

Les principaux résultats de ce chapitre sont publiés dans **[78] [81] [83] [87]**.

## 4.2 Un modèle de langue mixte pour la RI

L'objectif de notre approche est de mieux représenter le contenu sémantique des documents et des requêtes en introduisant une certaine sémantique dans leurs représentations. À cette fin, nous proposons un modèle de langue mixte, noté *LM-TC*, pour la recherche d'information qui combine les mots simples et les mots composés.

Une des contributions de ce modèle est la manière dont le modèle de langue des mots composés est estimé. Plus précisément, dans la plupart des travaux existants sur les modèles de langue, un modèle de document est estimé par le comptage, soit des mots simples ou des mots composés. Nous pensons, que le simple comptage des n-grammes est susceptible de biaiser l'importance réelle (poids) des mots composés (n-grammes) dans les documents. En effet, deux intuitions ont guidé notre réflexion, nous pensons d'abord que les mots composants d'un mot composé n'apportent pas la même contribution dans le poids final de mots composés. Nous introduisons à cet effet la notion de dominance de mot composant dans un mot composé. Nous considérons que les mots composants dominants contribuent plus que les autres mots. Deuxièmement, nous supposant que l'auteur d'un document utilise les mots composants isolément pour exprimer le mot composé comme abréviation après un nombre d'occurrences de mot composé. Nous proposons alors une nouvelle formule de calcul de fréquence des mots composés qui prend en compte aussi la fréquence de ses mots composants.

Enfin, nous pensons que, la prise en compte de tous les n-grammes (n> 1), (un n-gramme est composé de *n* mots non vides adjacents) peut introduire du bruit, en effet tous les n-grammes ne sont pas de mots composés réels. Nous proposons de considérer que les n-grammes qui sont fréquents dans la collection.

### 4.2.1 Description du modèle

Nous considérons une requête $Q$ et un document $D$ représentés dans le vocabulaire $V = \{T_1, \dots, T_m, t_1, \dots, t_n\}$ composé de mots simples $t_i$ et de mots composés $T_j$. Un mot composé peut être formé de deux ou plusieurs mots simples adjacents non vides. Ils sont extraits des documents et utilisés comme unités d'indexation supplémentaires.

En suivant la logique du modèle de langue et en considérant que le contenu d'un document comporte à la fois des mots simples et des mots composés, on peut considérer ainsi, que le document a été généré par un modèle mixte. Chacun produisant un type de terme.

Nous supposons donc que le modèle de document peut être estimé à l'aide de deux modèles : un modèle des mots simples ($M_{D_t}$) et un modèle des mots composés ($M_{D_T}$). Ainsi, étant donné une requête $Q$, exprimée par des mots simples et des mots composés, le modèle d'appariement document-requête que nous proposons combine les deux modèles de la manière suivante:

$$P(Q|D) = \prod_{t_i \in Q} P(t_i|D) \times \prod_{T_j \in Q} P(T_j|D) \quad (4.1)$$

Chaque mot simple et mot composé dans la requête est estimé en combinant les deux modèles du document. Formellement, on l'exprime comme suit:

$$P(t_i|D) = \lambda P(t_i|M_{D_t}) + (1-\lambda) P(t_i|M_{D_T}) \quad (4.2)$$

$$P(T_j|D) = \alpha P(T_j|M_{D_T}) + (1-\alpha) \prod_{t_k \in T_j} P(t_k|M_{D_t}) \quad (4.3)$$

Où λ et $\alpha$ ϵ [0, 1] sont des paramètres de lissage, $P(T_j|M_{D_T})$ et $P(t_i|M_{D_t})$ peuvent être évaluées en utilisant n'importe quel modèle de langue uni-gramme. Nous avons pour notre part opté pour le lissage de Dirichlet, car il a donné de meilleurs résultats que les autres méthodes de lissage **[209]**. Les deux modèles sont estimés comme suit:

$$P_{Dir}(t_i|M_{D_t}) = \frac{F(t_i, D_t) + \mu P(t_i|C_t)}{|D_t| + \mu} \quad (4.4)$$

Où $F(t_i, D_t)$ est la fréquence du mot simple $t_i$ dans le document $D$, $P(t_i|C_t)$ est le modèle de langue de la collection (la fréquence globale du terme est utilisée), $|D_t|$ est la longueur du document exprimée avec des mots simples, μ est le paramètre de lissage.
De la même manière on a:

$$P_{Dir}(T_j|M_{D_T}) = \frac{F(T_j, D_T) + \mu P(T_j|C_T)}{|D_T| + \mu} \quad (4.5)$$

Où $P(T_j|C_T)$ est le modèle de langue de la collection, $F(T_j, D_T)$ est la fréquence du mot composé $T_j$ dans le document $D$, elle peut être calculée par un simple comptage des occurrences du mot composé ou en utilisant notre méthode de comptage décrite dans section 4.2.3, et $|D_T|$ est la longueur du document représenté par des mots composés.

Nous détaillons dans les sections suivantes comment la fréquence du mot composé et la probabilité $P(t_i|M_{D_T})$ sont calculées. Nous introduisons tout d'abord dans la section suivante la notion de dominance d'un terme.

### 4.2.2 Dominance d'un terme

La pondération des mots composés demeure un problème ouvert en RI. En effet, il n'existe pas de schéma bien accepté pour la pondération des mots composés. La manière la plus simple de pondération des mots composés est l'utilisation du facteur de pondération $tf$ (en comptant le nombre d'occurrences d'un mot composé dans un document). Des alternatives ont été proposées; parmi elles l'adaptation du schéma de pondération $tf \times idf$ [**50**] [**96**], mais aucune amélioration notable n'est constatée.

Cependant, ces schémas de pondération ne tiennent pas compte d'un facteur important qui est l'importance des mots composants dans le mot composé; dans les schémas précédents cette importance est considérée identique. Or, dans la réalité un des mots composants peut être plus important que les autres termes, nous les nommons les termes dominants. Par exemple, le terme *«ordinateur»* est plus important que le terme *«personnel»* dans le mot composé *«ordinateur personnel»*.

Nous considérons intuitivement que la dominance d'un terme est déterminée par sa spécificité, nous proposons de l'estimer de la manière suivante :

$$imp(t) = {}^{N}/_{df} \tag{4.6}$$

Où $df$ est le nombre de documents où le terme $t$ apparaît, et $N$ le nombre de documents dans la collection $C$.

Nous attribuons à chaque mot simple $t$ sa probabilité de dominance dans son mot composé $T$ comme suit :

$$P(t|T) = \frac{imp(t)}{\sum_{t_i \in T}(imp(t_i))} \tag{4.7}$$

### 4.2.3 La fréquence des mots composés revisitée

Notre deuxième intuition dans la pondération des mots composés est la suivante : « nous supposons que l'auteur d'un document utilise les composants d'un mot composé isolément pour faire référence à ce mot composé comme abréviation après un certain nombre d'occurrences du mot composé ». Par exemple, un document contenant le mot composé *«énergie électrique»*, l'auteur utilise le terme *«énergie»* simplement pour désigner le mot composé *«énergie électrique»*. Afin de prendre en compte cette hypothèse, nous proposons de lisser la fréquence du mot composé en tenant compte de la fréquence de ses mots composants relativement à leur dominance dans le mot composé. La nouvelle fréquence d'un mot composé est exprimée alors comme suit:

$$F^n(T) = F(T) + \sum_{i=1}^{\#T} P(t_i|T) \times F(t_i) \quad (4.8)$$

Où $F^n(T)$ représente la nouvelle fréquence (revisitée) du mot composé $T$, $F(T)$ est la fréquence initiale du mot composé $T$, $P(t_i|T)$ est la probabilité de dominance du terme $t_i$ dans le mot composé $T$, $F(t_i)$ est la fréquence du terme $t_i$ dans le document, et $\#T$ est la taille du mot composé $T$ (nombre de mots simples).

Dans l'objectif d'illustrer cette seconde intuition, nous avons pris un ensemble de mots composés apparaissant dans les requêtes comme suit : (*protection measures* : 255, *cigarette consumption* : 257, *computer security* : 258, *seasonal affective* : 262, *genetic code* : 281), et nous avons examiné la validité de cette hypothèse dans tous les documents de la collection TREC *AP88* (décrite dans la section 4.3.2) où ces termes apparaissent avec leurs mots simples seuls. La table 4.1 présente les résultats obtenus. Nous avons adopté la notation suivante : un «+ » pour designer que l'hypothèse est vérifiée et un «-» pour signifier que l'hypothèse n'est pas vérifiée.

| Mots composés | Documents | Mots simples seuls | Vérifiée |
|---|---|---|---|
| | *AP881125-0098* | *seasonal , affective* | -+ |
| | *AP881119-0062* | *affective* | - |
| *Protection measures* | *AP880427-0119* | *measures* | + |
| | *AP880728-0136* | *protection* | + |
| | *AP880801-0124* | *protection* | + |
| | *AP881031-0224* | *protection* | + |
| *Cigarette consumption* | *AP880401-0010* | *cigarette , consumption* | + + |
| | *AP880401-0182* | *cigarette* | + |
| | ***AP880418-0130*** | ***cigarette*** | **+** |
| | *AP880516-0160* | *cigarette* | + |
| | *AP880521-0203* | *cigarette , consumption* | + |
| | *AP880620-0159* | *cigarette* | + |
| | *AP880921-0141* | *cigarette* | + |
| | *AP880922-0093* | *cigarette* | - |
| | *AP881121-0014* | *cigarette* | + |
| *Computer security* | *AP880417-0001* | *computer , security* | ++ |
| | *AP880524-0075* | *computer* | + |
| | *AP880602-0271* | *computer , security* | -- |
| | *AP880606-0236* | *computer* | + |
| | *AP880616-0002* | *computer* | + |
| | *AP880617-0066* | *computer , security* | -- |
| | *AP880708-0177* | *computer , security* | ++ |
| | *AP880624-0254* | *computer , security* | -- |
| | *AP880705-0051* | *computer* | + |
| | *AP880831-0247* | *computer , security* | ++ |
| | *AP880923-0108* | *security* | + |
| | *AP881104-0021* | *computer* | + |
| | *AP881104-0234* | *computer* | + |
| | *AP881104-0315* | *computer* | + |
| | *AP881105-0021* | *computer* | + |
| | *AP881105-0034* | *computer* | + |
| | *AP881105-0035* | *computer* | + |
| | *AP881105-0083* | *computer , security* | ++ |
| | *AP881105-0161* | *computer* | + |
| | *AP881105-0162* | *computer , security* | ++ |
| *Genetic code* | *AP880215-0223* | *genetic , code* | ++ |
| | *AP880309-0243* | *genetic, code* | ++ |
| | *AP880419-0161* | *genetic* | - |
| | *AP880513-0058* | *code* | + |
| | *AP880601-0230* | *code* | + |
| | *AP881223-0010* | *code* | + |
| | *AP880601-0272* | *code* | + |
| | *AP881021-0141* | *code* | + |

**Table 4.1 Exemples de mots simples référençant les mots composés**

À partir des exemples ci-dessus, nous pouvons noter que notre hypothèse est satisfaite dans la majorité des cas. La vérification de cette hypothèse d'une manière automatique nécessite une analyse syntaxique et sémantique des textes.

Pour illustrer la méthode de pondération des mots composés, nous prenons l'exemple suivant :

Soit le document *D*= « ***AP880418-0130 »***, le mot composé *T*= « ***cigarette consumption »***, contenant les deux mots simples $t_1$ = « ***cigarette*** » et $t_2$ = « ***consumption »***. Les statistiques sur ces termes sont les suivantes :

*F(T)*=1 : est la fréquence initiale du mot composé *T* dans le document *D*.

$F(t_1)$=3 : est la fréquence du terme $t_1$ apparaissant seul dans le document *D*.

$F(t_2)$=0 : est la fréquence du terme $t_2$ apparaissant seul dans le document *D*.

$df(t_1)$=817 : est le nombre de documents dans la collection contenant le terme $t_1$ .

$df(t_2)$=*586* : est le nombre de documents dans la collection contenant le terme $t_2$.

*P(t1|T)*=( 586)/( 586+817)= 0.42 : est la probabilité de dominance du terme $t_1$ dans *T*.

*P(t2|T)*=( 817)/( 586+817)= 0.58 : est la probabilité de dominance du terme $t_2$ dans *T*

La fréquence revisitée du mot composé «***cigarette consumption»*** dans le document ***« AP880418-0130 »*** est calculée de la manière suivante :

$$F^n(T) = F(T) + P(t_1|T) \times F(t_1) + P(t_2|T) \times F(t_2)$$
$$= 1 + 0.42 \times 3 + 0.58 \times 0 = 2.253$$

Ainsi, la fréquence revisitée du mot composé «***cigarette consumption***» dans le document « ***AP880418-0130 »*** est égale à 2.253, alors que sa fréquence calculée par simple comptage du nombre d'occurrence est de 1.

### 4.2.4 Estimation de la probabilité $P(t_i|M_{D_T})$

Afin d'estimer cette probabilité, nous proposons un modèle similaire au modèle de traduction [**18**]. Par conséquent, nous exprimons cette probabilité comme suit:

$$P(t_i|M_{D_T}) = \sum_T (P(t_i|T) \times P_{Dir}(T|M_{D_T})) \quad (4.9)$$

Dans cette formule, le passage d'un mot simple vers un document *D* est réalisé à travers tous les mots composés contenant le mot simple.

Cependant, comme nous l'avons mentionné précédemment, nous supposons que, l'auteur lorsqu'il utilise un mot simple dans un document, il ne peut renvoyer qu'à un mot composé donné. Nous considérons que ce mot composé est le plus fréquent qui contient ce mot simple dans le document. Ce mot composé noté $\hat{T}$ est sélectionné par la formule suivante:

$$\hat{T} = argmax_{\forall\, T \in D_T \,\wedge\, t_i \in T} \left( P(t_i|T) \times P_{Dir}\left(T \middle| M_{D_T}\right) \right) \tag{4.10}$$

En introduisant le facteur $\hat{T}$dans la formule (4.9), elle peut être réécrite de la manière suivante :

$$P\left(t_i \middle| M_{D_T}\right) = P(t_i|\hat{T}) \times P_{Dir}\left(\hat{T} \middle| M_{D_T}\right) \tag{4.11}$$

## 4.3 Expérimentations et résultats

Dans cette section nous présentons initialement notre environnement d'expérimentation : les détails d'implémentation de notre approche, les outils utilisés, les collections et topics (requêtes) considérés et les métriques d'évaluation adoptées. Nous passons ensuite à la présentation des résultats expérimentaux obtenus et à l'analyse et l'évaluation de l'apport de chaque élément de notre modèle, puis nous comparons notre modèle avec deux modèles présentés dans l'état de l'art **[129] [137]**. Enfin, nous analysons la robustesse de notre modèle.

### 4.3.1 Implantation de l'approche

Nous décrivons dans ce qui suit les différentes étapes que nous avons suivis pour l'indexation et l'évaluation des requêtes.

1- ***Le prétraitement de la collection***: en premier lieu, nous procédons au prétraitement de chaque collection (nous avons travaillé sur différentes types de collections décrites en section 4.3.2). Nous parsons les documents, nous éliminons les mots vides et nous appliquons l'algorithme de Porter [**152**]. Les documents traités obtenus sont ensuite utilisés comme entrées par l'outil Text-NSP [**13**] pour l'extraction des mots composés.

2- ***L'extraction des mots composés*** : pour l'extraction des mots composés nous avons utilisé l'outil Text-NSP. Le package Text-NSP est un outil permettant l'identification et la sélection de n-grammes ou séquence de mots dans une collection de texte. Dans le processus d'extraction des mots composés, nous tenons compte des paramètres suivants:

***La directionalité entre mots simples*** : dans certains cas la préservation de l'ordre des mots est important pour garder le sens de l'unité d'indexation, ceci est vrai par exemple pour le terme «*système d'exploitation* », dans d'autres cas l'ordre n'est pas important, par exemple le terme «*système et organisation*». La plupart des travaux réalisés en RI utilisant les mots composés comme unité d'indexation sont basés sur la non directionnalité des termes; dans notre cas la contrainte d'ordre est respectée lors de l'identification des mots composés.

***La distance*** : la distance entre les termes formant le mot composé (ou l'adjacence ou la non- adjacence des termes) : l'intensité de liens entre termes – opérationnalisée à travers la distance- reflète la proximité sémantique entre termes. La capture de cette proximité est importante pour la recherche d'information. Les études réalisées en RI sur l'extraction des mots composés suppose que la cooccurrence des mots dans les éléments fortement structurés (c.-à-d., une phrase) est plus significative que dans les éléments moins structurés (c.-à-d., des paragraphes ou des sections). Ainsi, la recherche sur l'extraction des mots composés a été dominée par l'analyse de phrases. Dans notre cas, nous avons adopté l'adjacence entre termes, un mot composé est reconnu si et seulement s'il est composé de mots simples adjacents.

***La taille des mots composés*** **:** en principe les mots composés peuvent être de n'importe quelle longueur (supérieure ou égale à 2). Dans notre cas, nous restreindrons la taille des mots composés à deux, qui est une pratique commune et scalable pour de grandes collections hétérogènes.

Text-NSP offre deux modules pour la sélection des mots composés: « *count.pl* » et « *statistic.pl* ». Premièrement, nous avons utilisé le module « *count.pl* » pour compter les bi-grammes. Nous ne gardons dans une liste intermédiaire que les bi-grammes ayant une fréquence supérieure à un seuil noté « *seuil_freq* ». Cette liste est ensuite utilisée par le deuxième module « *statistic.pl* », qui permet de calculer un score pour chaque bi-gramme de la liste et cela en utilisant une mesure statistique. Dans notre cas, nous avons utilisé la mesure de Pointwise Mutual Information (PMI). L'étude menée par Petrovic et al [**150**] a montré que la mesure PMI permet l'identification de mots composés pertinents pour la RI. Nous ne gardons dans la liste finale que les bi-

grammes ayant un score supérieur à un seuil noté « *seuil_PMI* ». Cette liste est ensuite utilisée dans les étapes d'indexation et de recherche.

3- ***Indexation, recherche et évaluation*** : nous avons utilisé la plate-forme de RI Terrier 2.1 [**131**] pour l'indexation, la recherche et l'évaluation de notre approche. Les documents sont indexés en utilisant les mots composés reconnus dans l'étape 2 et les mots simples utilisés traditionnellement en RI.

### 4.3.2 Les collections et les requêtes utilisées

Nous avons évalué notre modèle en utilisant trois collections TREC; plus précisément deux collections de journaux *AP88* (Associated Press News, 1988) et *WSJ90-92*(Wall Street Journal, 1990-92) et une collection de type web, la *WT10g*. Pour la recherche nous avons utilisé 100 requêtes pour chaque type de collection. Pour les collections de type journaux nous avons utilisé les requêtes numérotées « *201-300* » dans TREC, où les champs « *title* » et « *description* » sont considérés. En ce qui concerne la collection *WT10g* nous avons utilisé les requêtes numérotées « *451-550 »*, où seul le champ « *title* » est considéré, cette restriction est due au fait que les requêtes sur le web sont courtes.

La figure 4.1 ci-dessous montre un exemple de requête utilisée sur la collection *WT10g*. Cette requête est composée de trois champs. Seul le champ « *title* » est utilisé pour construire la requête. La requête construite par notre système est «*Bengals cat* » (reconnu comme un mot composé).

```
<top>
<num> Number: 451
<title> What is a Bengals cat?
<desc> Description:
Provide information on the Bengal cat breed.
<narr> Narrative:
Item should include any information on the
Bengal cat breed, including description, origin,
characteristics, breeding program, names of
breeders and catteries carrying bengals.
References which discuss bengal clubs only are
not relevant.  Discussions of bengal tigers are
not relevant.
</top>
```

**Figure 4.1 Exemple de requête (451) de la collection *WT10g*.**

La table suivante montre quelques statistiques sur les collections et requêtes utilisées

| Collection | #documents | Topics |
|---|---|---|
| *WSJ90-92* | *74,520* | *201-300* |
| *AP88* | *79,919* | *201-300* |
| *WT10g* | *1,692,096* | *451-550* |

**Table 4.2** Aperçu sur les collections et requêtes utilisées

### 4.3.3 Evaluation

Dans cette partie nous présentons l'évaluation des performances de notre modèle (*LM-TC*). Nous avons procédé par étape.

En premier lieu, nous avons évalué l'impact de chaque élément caractérisant notre modèle, à savoir :

- Le filtrage des bi-grammes : cette version de notre modèle, notée *LM-TC_0*, est basé sur le filtrage des bi-grammes et l'utilisation de la fréquence initiale pour estimer le poids des bi-grammes sélectionnés (mots composés), et pour calculer la probabilité $P(t_i|M_{D_T})$ la formule (4.9) est utilisée, en d'autres termes le facteur $\hat{T}$ n'est pas pris en compte.
- La nouvelle méthode de pondération des mots composés proposée : cette version de notre modèle, notée *LM-TC_1*, est similaire au modèle précédent (*LM-TC_0*), sauf que, la fréquence revisitée est utilisée pour estimer le poids des mots composés, c'est-a-dire la formule (4.8) est exploitée.
- L'utilisation du facteur $\hat{T}$ introduit dans la formule (4.11) : ce modèle, noté *LM-TC*, correspond à la version *LM-TC_1* incluant le facteur $\hat{T}$.

En second lieu, nous avons comparé notre modèle (*LM-TC*) avec deux modèles de l'état l'art, à savoir, le modèle *MRF* [**137**] et le modèle *PLM* [**129**].

Enfin, nous avons évalué la robustesse de notre modèle.

Pour l'évaluation des performances des différents modèles, nous avons utilisé la mesure de la précision moyenne, MAP (Mean Average Precision), qui est une mesure largement utilisée pour l'évaluation de l'efficacité des Systèmes de Recherche d'Information.
La précision moyenne donne une vue globale des performances d'un modèle de recherche d'information au travers d'un ensemble de requêtes. Cependant, la moyenne sur un ensemble de requêtes peut cacher beaucoup de détails. Il n'est pas alors évident de déterminer ce qui

conduit à l'augmentation ou à la diminution de la précision moyenne. Pour obtenir une meilleure explication, et mieux comprendre la différence entre les différents modèles, nous avons effectué une analyse requête-par-requête.

Afin, de vérifier la significativité des résultats obtenus, nous avons effectué le test de Student et nous avons joint « $^{+}$ » et « $^{++}$ » pour l'indice de performance dans les différents tables des résultats lorsque le test passe respectivement 95% et 99%.

Notre modèle a plusieurs paramètres de contrôle, ceux de filtrage des bi-grammes (*seuil_fre* et *seuil_PMI*), les paramètres de combinaison des mots simples et mots composés ($\lambda$ : formule (4.2) et $\alpha$ : formule (4.3)) et le paramètre de la méthode de lissage de Dirichlet ($\mu$).

Afin de trouver les valeurs de ces paramètres optimisant les performances de notre modèle, nous avons utilisé différentes valeurs pour les deux seuils de filtrage (*seuil_fre* et *seuil_PMI*). Pour chaque valeur de *seuil_fre* (allant de 0 à 30 avec un pas de 5) nous avons utilisé différentes valeurs de *seuil_PMI* (allant de 0 à 3 avec un pas de 1). Pour chaque couple (*seuil_fre*, *seuil_PMI*) obtenu nous varions la valeur des deux paramètres $\alpha$ et $\lambda$ allant de 0 à 1 avec un pas de 0.1.

Concernant le paramètre du modèle de langue, la valeur du paramètre de Dirichlet ($\mu$) est empiriquement fixée à 2500 sur toutes les collections.

La table 4.3 ci-dessous illustre les valeurs des différents paramètres optimisant les performances (Précision Moyenne) de notre modèle sur chacune des collections.

| Modèle | *LM-TC* | | |
|---|---|---|---|
| Collections / Paramètres | *AP88* | *WSJ90-92* | *WT10g* |
| *seuil_fre* | 10 | 10 | 15 |
| *seuil_PMI* | 1 | 1 | 2 |
| $\lambda$ (formule (4.2)) | 0.6 | 0.8 | 0.7 |
| $\alpha$ (formule (4.3)) | 0.4 | 0.6 | 0.3 |
| $\mu$ | 2500 | 2500 | 2500 |

**Table 4.3 Valeurs des paramètres utilisées dans notre modèle (*LM-TC*)**

#### 4.3.3.1 Apport de filtrage des bi-grammes

Pour évaluer l'impact d'utilisation des mots composés (les bi-grammes filtrés) et les mots simples comme unités d'indexation et de recherche, nous avons comparé notre modèle noté *LM-TC_0* avec deux modèles : le premier est le modèle uni-gramme basé sur les mots simples

noté *ULM*, le second est similaire au modèle *LM-TC_0*, à l'exception que dans ce modèle tous les bi-grammes sont considérés, ce second modèle est noté *BGM*.

Il faut noter que dans le modèle uni-gramme (*ULM*) nous avons utilisé le modèle de Dirichlet décrit par la formule (4.4).

Nous rapportons dans la table suivante les résultats obtenus (Précision Moyenne).

| | *ULM* | *BGM* | *BGM %ULM* | *LM-TC_0* | *LM-TC_0 %ULM* | *LM-TC_0 %BGM* |
|---|---|---|---|---|---|---|
| *WSJ90-92* | 0.1852 | 0.1935 | +4.48%$^{+}$ | **0.1978** | **+6.8%**$^{+}$ | **+2.22%** |
| *AP88* | 0.2338 | 0.2409 | +3.04%$^{++}$ | **0.2464** | **+5.39%** | **+2.28%** |
| *WT10g* | 0.2085 | 0.2202 | +5.61%$^{++}$ | **0.2275** | **+9.11%**$^{++}$ | **+3.31%**$^{+}$ |

**Table 4.4 Comparaison des différents Modèles (ULM, BGM, LM-TC_0)**

A partir de cette table, nous pouvons tirer les remarques et les conclusions suivantes :

- Le modèle basé sur les bi-grammes (*BGM*) améliore les résultats du modèle uni-gramme ne tenant en compte que des mots simples et cela sur les trois collections utilisées. Ces améliorations significatives vont de +3.04 % (sur la collection *AP88*) à +5.61% (sur la collection *WT10g*). Cela montre, que l'usage des bi-grammes, comme une seconde évidence, à coté des mots simples peut améliorer significativement les performances de la recherche d'information.
- Notre modèle basé sur le filtrage des bi-grammes (*LM-TC_0*) améliore les résultats du modèle prenant en compte tous les bi-grammes (*BGM*). Les améliorations constatées vont de +2.22% (sur la collection *WSJ90-92*) à +3.31% (sur la collection *WT10g*), cette dernière amélioration est significative. Ces résultats montrent que l'utilisation des mots composés à côté des mots simples peut être bénéfique pour la RI, et cela lorsque les mots composés sont bien extraits en appliquant les deux filtres (*seuil_fre* et *seuil_PMI*) et lorsque la combinaison des mots composés et des mots simples et réalisée d'une manière appropriée.

**Analyse requête-par-requête**

Nous illustrons dans les figures ci-dessous la comparaison des résultats requête-par-requête du modèle *LM-TC_0* avec les modèles *ULM* et *BGM* sur les trois collections.

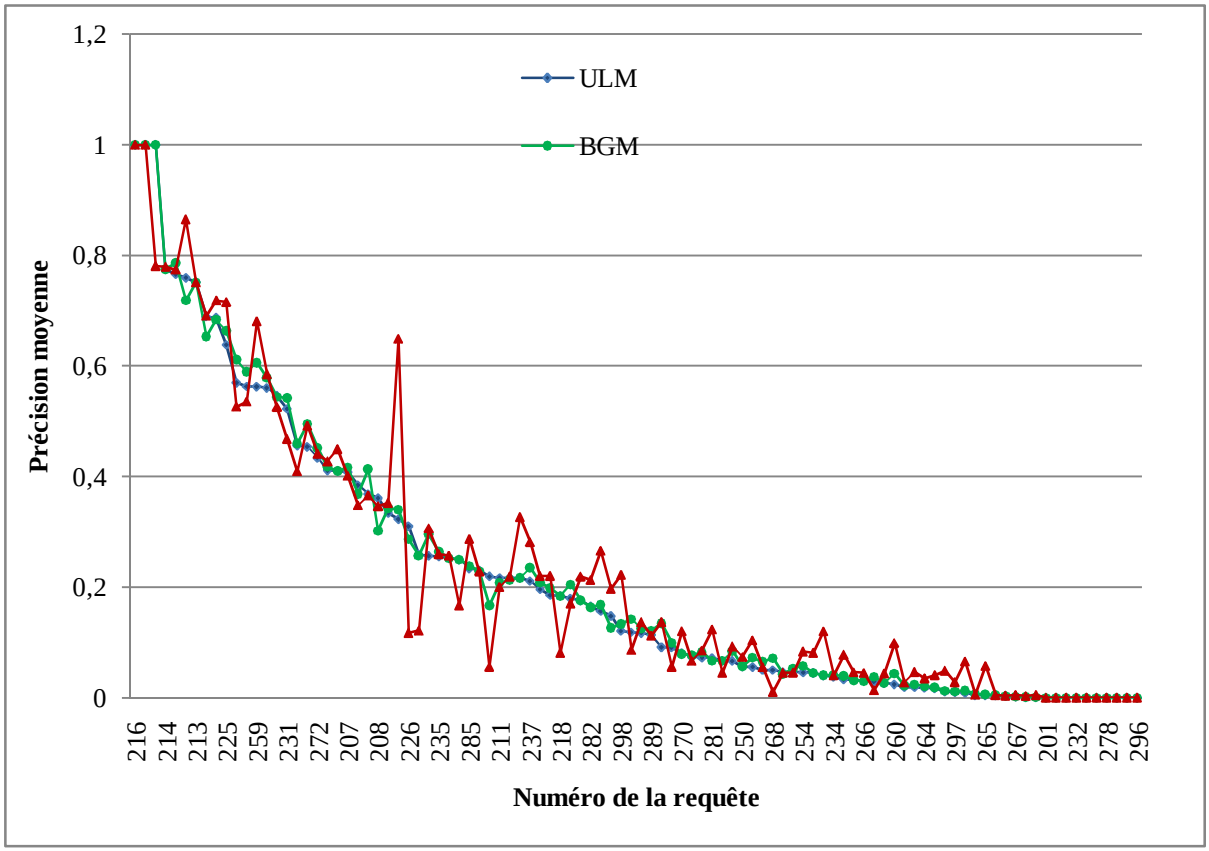



**Figure 4.2 Analyse requête-par-requête des différents modèles sur la collection *AP88*.**

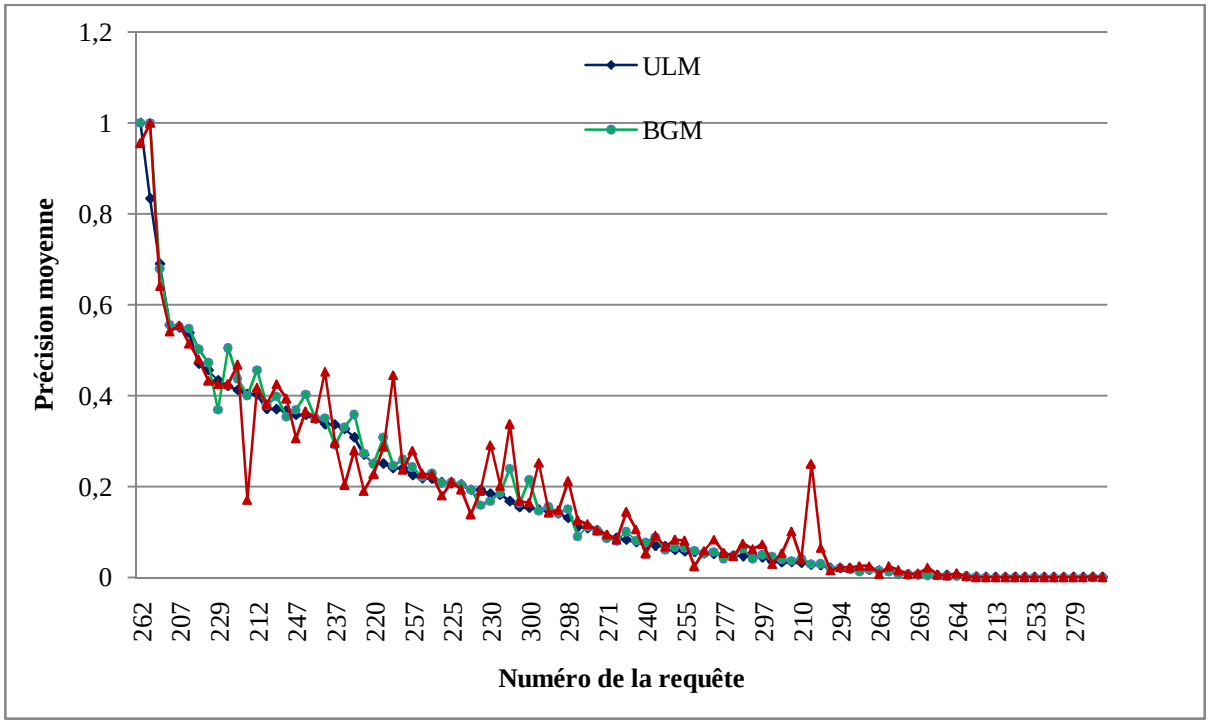



**Figure 4.3 Analyse requête-par-requête des différents modèles sur la collection *WSJ90-92*.**

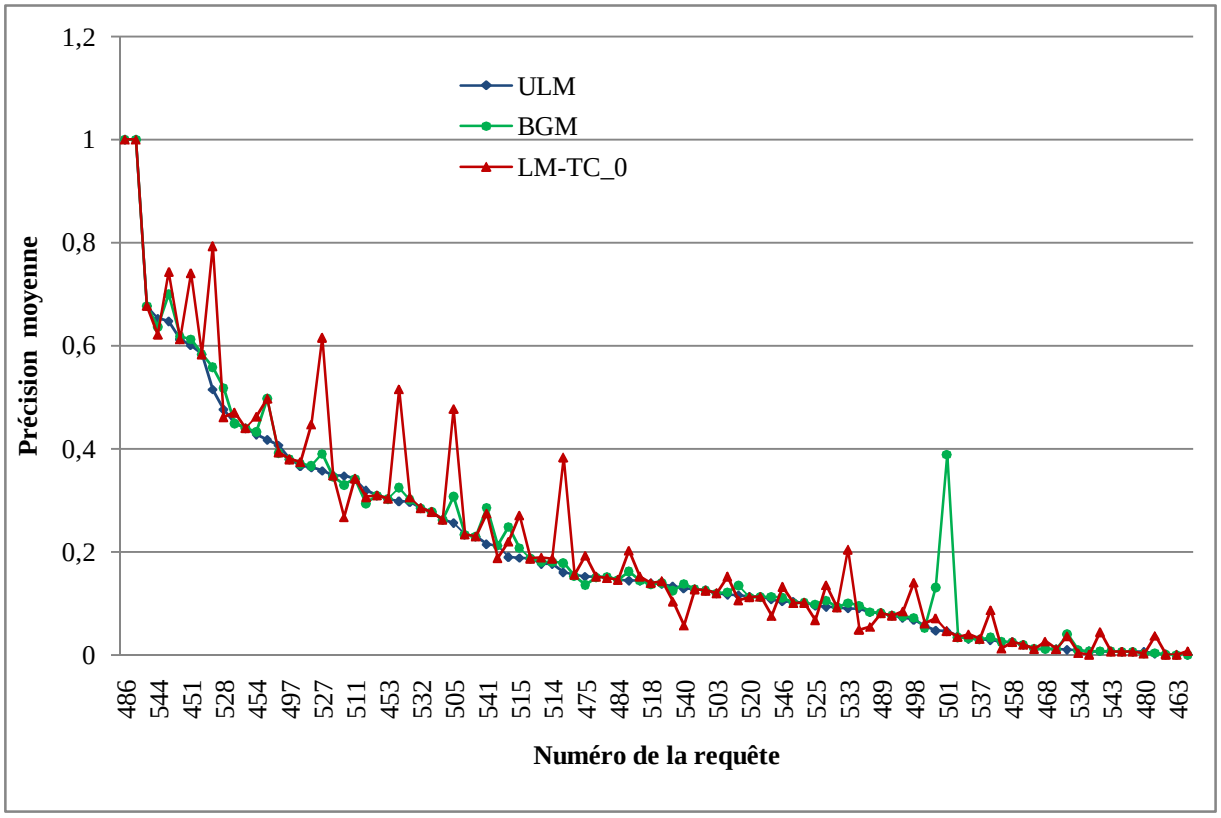



**Figure 4.4 Analyse requête-par-requête des différents modèles sur la collection *WT10g*.**

Pour bien analyser l'apport de filtrage des bi-grammes nous avons divisé les requêtes en deux classes : la première notée *QTC* concerne les requêtes contenant au moins un mot composé, la seconde classe notée *Qts* concerne les requêtes ne contenant aucun mot composé.

A partir de l'analyse des résultats obtenus nous avons noté les remarques suivantes :

- Sur la collection *WSJ90-92* nous avons 88 requêtes de type *QTC* et seulement 12 requêtes de type *Qts* ; sur les 88 requêtes de type *QTC* notre modèle présente des améliorations dans 51 requêtes vis-à-vis du modèle *BGM*, ce dernier outrepasse notre modèle sur 28 requêtes. Les deux modèles présentent des résultats égaux sur 9 requêtes. En considérant uniquement les requêtes de type *QTC* nous constatons une amélioration de la précision moyenne (MAP) de l'ordre de +2.42%. Cette amélioration est légèrement supérieure à l'amélioration constatée en prenant en compte toutes les requêtes, qui est de l'ordre de +2.22%.

  Sur les 12 requêtes de type *Qts*, notre modèle améliore le modèle *BGM* dans 4 requêtes. Les deux modèles donnent les mêmes résultats dans 4 requêtes. Sur les 4 requêtes restantes le modèle *BGM* outrepasse notre modèle. Sur ce type de requête

l'amélioration moyenne apportée par notre modèle est très minime, elle est de l'ordre de +0.88.

- Les améliorations obtenues par notre modèle par rapport au modèle *BGM* sur la collection *AP88*, en considérant les deux types de requêtes : *QTC* (89 requêtes) et *Qts* (11 requêtes), restent dans les mêmes proportions que celles obtenues sur la collection *WSJ90-92*. Ces améliorations sont de l'ordre de +2.77% (amélioration significative) avec les requêtes de type *QTC* et de +0.26% avec les requêtes de type *Qts*.
- Sur la collection *WT10g*, les résultats obtenus avec notre modèle sont significativement importants en utilisant les requêtes de type *QTC*, +6.38% (amélioration significative) par rapport à ceux du modèle *BGM*. En termes de nombre de requêtes nous avons 40 requêtes où notre modèle donne de meilleurs résultats sur 61 requêtes de type *QTC*. Nous notons aussi que sur cette collection le nombre de requêtes de type *QTC* est moins important que sur les deux autres collections, cela est dû au fait que seul le champ titre des requêtes est utilisé (requêtes courtes). L'amélioration obtenue avec le deuxième type de requête reste dans la même proportion que celles obtenues sur les deux autres collections, elle est de l'ordre de +1.88%.

La table 4.5 ci-dessous récapitule les résultats obtenus, où les colonnes (+,=,-) listent le nombre de requêtes pour lesquelles notre modèle obtient de (meilleur, égal, moindre) précision que les deux autres modèles (*ULM*, *BGM*).

| Collection | Type | *LM-CT_0% ULM* | | | | *LM-CT_0%BGM* | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | + | = | - | Amelioration | + | = | - | Amélioration |
| *WSJ90-92* | 88(*QCT*) | 57 | 8 | 23 | +7.7%$^{+}$ | 51 | 9 | 28 | +2.42% |
| | 12(*Qst*) | 2 | 4 | 6 | +0.90% | 4 | 4 | 4 | +0.88 |
| *AP88* | 89(*QCT*) | 55 | 9 | 25 | +5.11%$^{+}$ | 49 | 9 | 31 | +2.77%$^{+}$ |
| | 11(*Qst*) | 5 | 3 | 3 | +1.50% | 5 | 3 | 3 | +0.26% |
| *WT10g* | 61(*QCT*) | 39 | 4 | 18 | +14.54%$^{++}$ | 40 | 3 | 18 | +6.38%$^{+}$ |
| | 37(*Qst*) | 4 | 30 | 3 | +3.15% | 14 | 19 | 4 | +1.88% |

**Table 4.5 Résultats par type de requêtes des différents modèles (*ULM, BGM, LM-TC_0*)**

A partir de ces résultats, nous pouvons tirer les conclusions suivantes :

**1.** Les améliorations obtenues, même si elles sont minimes, avec les requêtes de type *Qts* sont dues à l'utilisation dans la formule (4.2) d'un modèle mixte combinant les mots simples et composés pour l'estimation du modèle des mots simples.

**2.** Les améliorations constatées avec les requêtes de type *QTC* sont plus importantes que celles obtenues avec les requêtes de type *Qts*. Ceci est dû principalement à l'utilisation de filtrage des bi-grammes (mots composés).

Pour bien comprendre les raisons de ces améliorations sur ce type de requêtes (*QTC)*, nous avons examiné manuellement quelques requêtes. Nous illustrons ci-dessous un exemple de requête. Sur la collection *WSJ90-92*, la requête numéro 258 (***computer security*** *identify instances of illegal entry into sensitive* ***computer networks*** *by non authorized personnel*), où les termes « ***computer security*** **»** et «***computer networks*** **»** sont sélectionnés comme des mots composés. Sur cette requête notre modèle (*LM-TC_0*) obtient une précision moyenne de l'ordre de 0.0649, et permet de retrouver cinq (5) documents pertinents. Les résultats correspondants au modèle *BGM* pour cette requête sont : 0.0301 de précision et 5 documents pertinents retrouvés. Sachant que le nombre total de documents pertinents pour cette requête est cinq (5). La table 4.6 ci-dessous montre le rang des documents pertinents dans la liste retournée de documents (1000 documents) pour les deux modèles.

| Documents pertinents | Mots composés | Rang du Document | |
|---|---|---|---|
| | | *BGM* | *LM-TC_0* |
| ***WSJ900921-0017*** | ***computer security*** | **178** | **13** |
| *WSJ910315-0028* | / | 12 | 28 |
| ***WSJ900507-0106*** | ***computer security*** | **539** | **46** |
| ***WSJ910610-0091*** | ***computer network*** | **61** | **55** |
| *WSJ900817-0032* | / | 486 | 156 |

**Table 4.6 Le rang des documents pertinents avec les deux modèles (*BGM, LM-TC_0*)**

A partir de cette table nous remarquons les points suivants:

- Les documents pertinents *WSJ900921-0017* et *WSJ900507-0106* qui contiennent le terme «***computer security***» ont vu leur rang passé de 178 et 539 en utilisant le modèle *BGM* à 13 et 46 respectivement, en utilisant notre modèle *LM-TC_0*.
- Le document *WSJ910610-0091* qui contient le mot composé «***computer network***» a été promu du rang 61 au rang 55, cette progression est moins importante que celle obtenue avec les documents contenant le mot composé «***computer security***», car la requête est plus centrée sur la recherche de documents qui couvrent le mot composé «***computer security***» que sur les documents qui couvrent le mot composé «***computer***

***network***». Ce point reste l'un de nos perspectives à l'avenir. C'est-à-dire comment pondérer les mots composés dans une requête.

Cependant, dans certaines requêtes, le filtrage des bi-grammes échoue. Par exemple, la requête numéro 239 (*Are there certain regions in the **United States** where specific cancers seem to be concentrated? What **conditions exist** that might **cause t**his **problem**?*). Dans cette requête « ***United States* », « *conditions exist* »** et **« *cause problem*** » sont choisis comme mots composés. Sur cette requête, le modèle *BGM* permet d'avoir une précision moyenne de l'ordre de 0,0359, tandis que notre modèle (*LM-TC_0*) atteint une précision moyenne de l'ordre de 0,029. La raison de cette régression est due au fait que des bi-grammes pertinents n'ont pas été choisis par notre méthode tels que le bi-gramme: « ***specific cancers***».

#### 4.3.3.2 Impact de la fréquence revisitée des mots composés

Pour évaluer l'impact de la fréquence revisitée des mots composés, nous avons comparé la version de notre modèle *LM-TC_1*, avec la version *LM-TC_0* et le modèle de langue uni-gramme. Le modèle *LM-TC_1* est basé sur la fréquence revisitée d'un mot composé.

La table 4.7 montre la comparaison entre les différents modèles en termes de précision moyenne (MAP).

| | *ULM* | *LM-TC_0* | *LM-TC_1* | *LM-TC_1% ULM* | *LM-TC_1% LM-TC_0* |
|---|---|---|---|---|---|
| *WSJ90-92* | 0.1852 | 0.1978 | 0.2017 | **+8.90%**$^{++}$ | **+1.97%**$^{+}$ |
| *AP88* | 0.2338 | 0.2459 | 0.2508 | **+7.27%**$^{+}$ | **+1.99%**$^{+}$ |
| *WT10g* | 0.2085 | 0.2271 | 0.2328 | **+11.65%**$^{++}$ | **+2.51%**$^{+}$ |

**Table 4.7 Comparaison des performances des modèles (*ULM, LM-TC_0, LM-TC_1*)**

Nous pouvons noter à travers les résultats illustrés dans la table précédente que :

- La version de notre modèle (*LM-TC_1*), implémentant la nouvelle méthode de pondération des mots composés proposée (fréquence revisitée), améliore la version de notre modèle se basant sur l'utilisation de la fréquence initiale pour la pondération des mots composés. Cette amélioration passe de +1.97% (amélioration significative) sur la collection *WSJ90-92* à +2.51% (amélioration significative) sur la collection *WT10g*.

**Analyse requête-par-requête**

Afin de mieux comprendre l'apport cette nouvelle méthode de pondération des mots composés, nous avons examiné les requêtes de type *QTC*.

Nous récapitulons les résultats de la comparaison entre les deux versions de notre modèle (*LM-TC_0* et *LM-TC_1*) dans les figures ci-dessous :

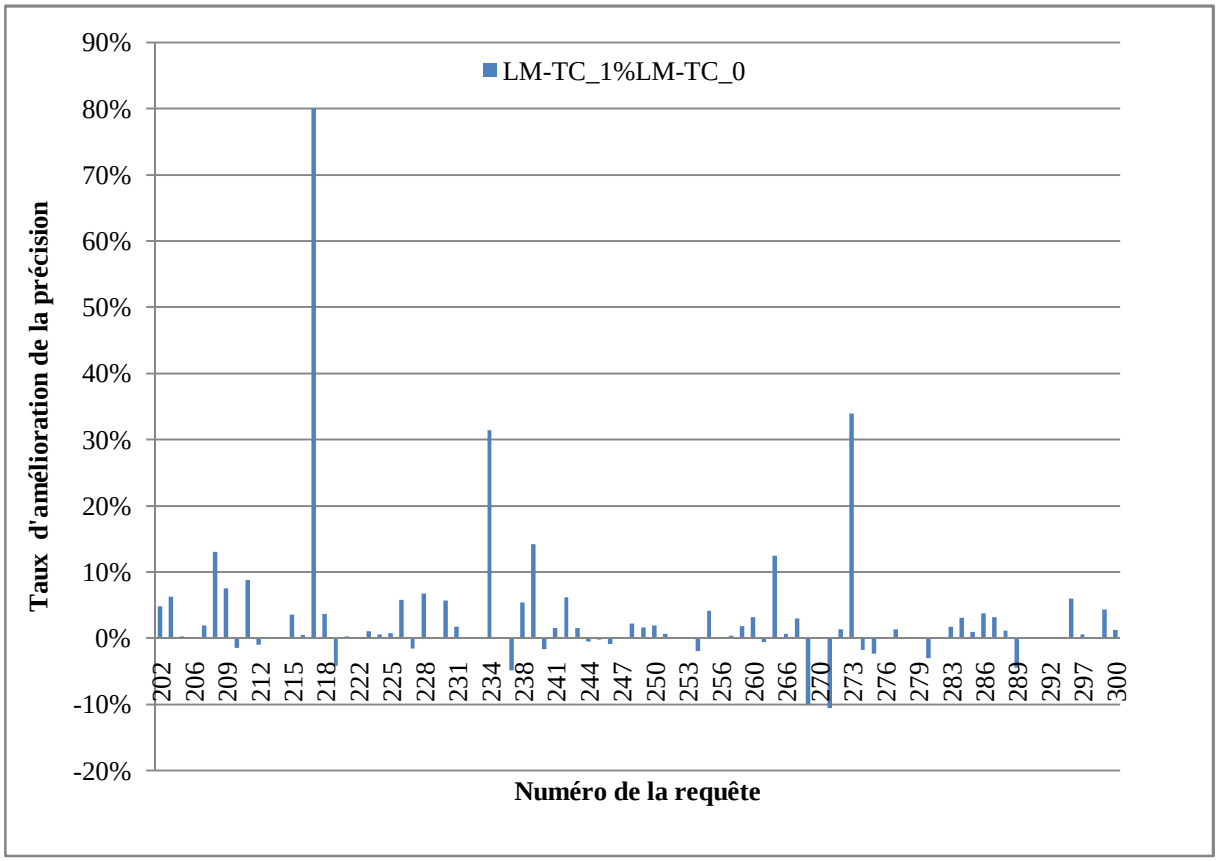



**Figure 4.5 Analyse requête-par-requête sur la collection *WSJ90-92*.**

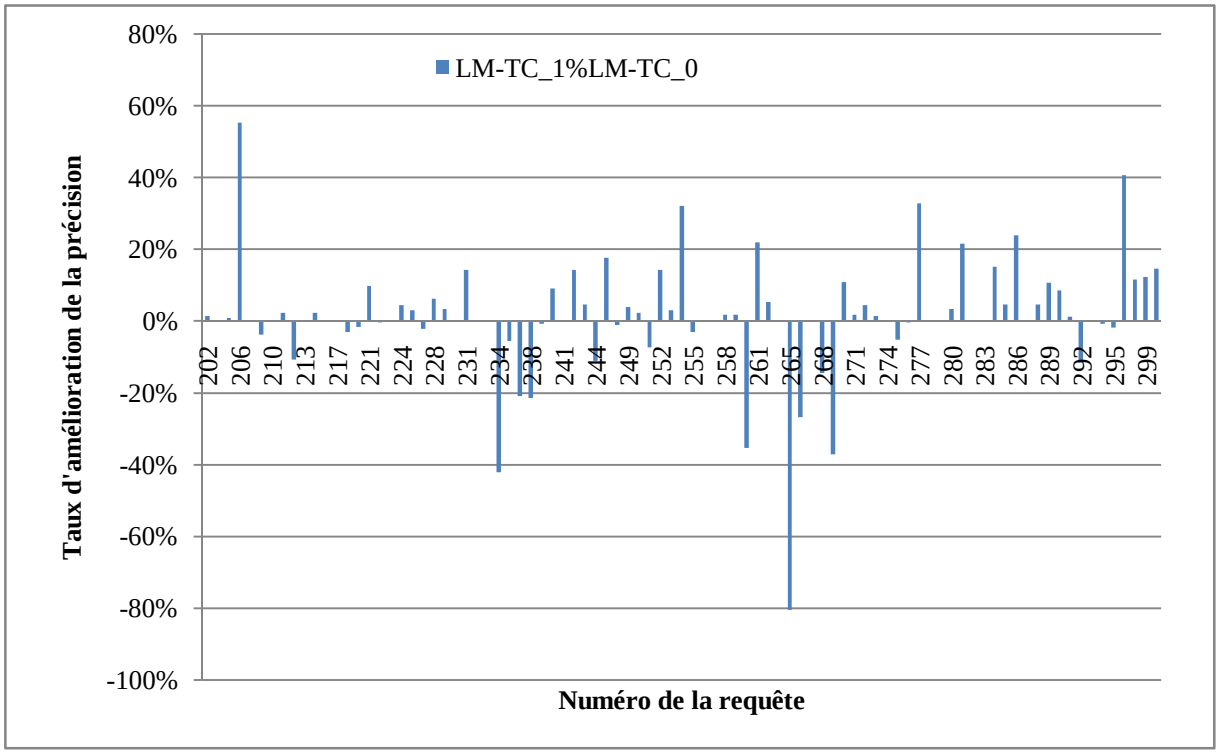



**Figure 4.6 Analyse requête-par-requête sur la collection *AP88*.**

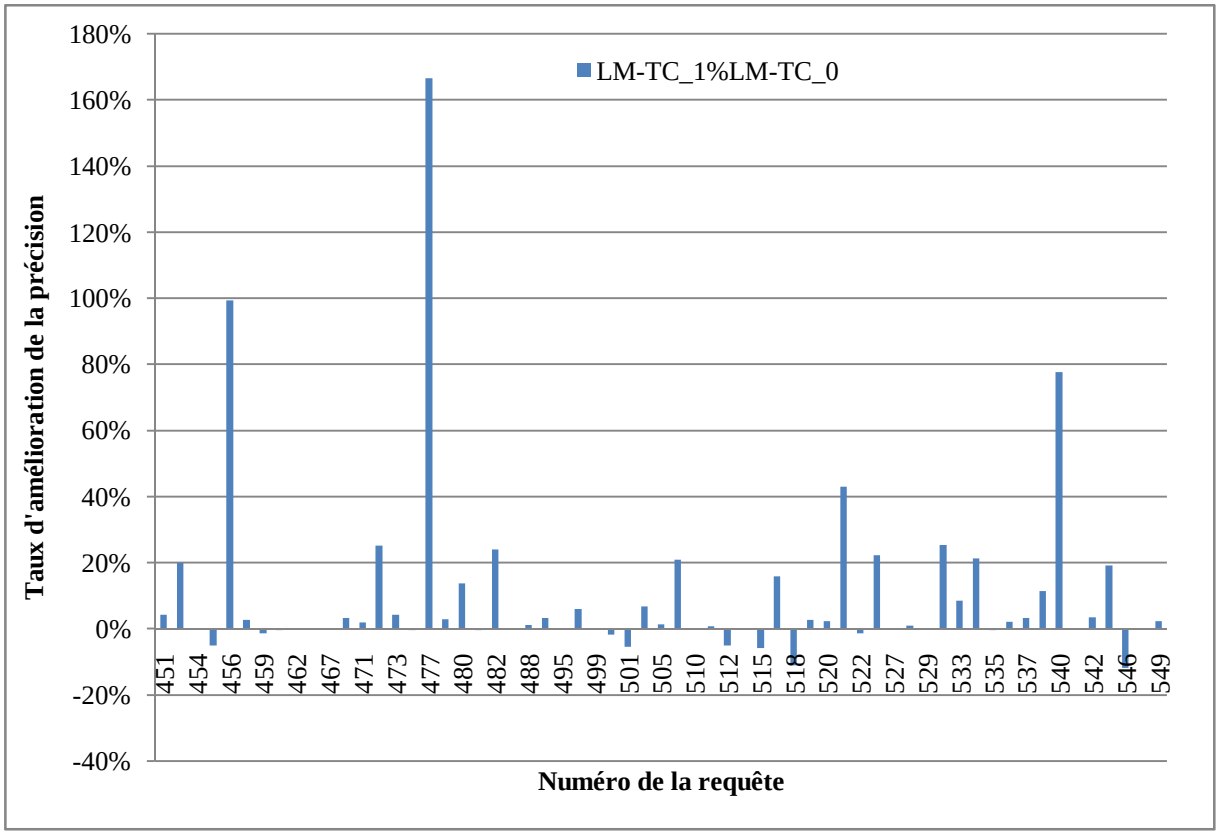



**Figure 4.7 Analyse requête-par-requête sur la collection *WT10g*.**

A partir des résultats obtenus, nous notons les points suivants :

- Le nombre de requêtes de type *QTC* est de 88 dans la collection *WSJ90-92*. Sur cet ensemble de requêtes le modèle *LM-TC_1* donne de meilleurs résultats par rapport au modèle *LM-TC_0* dans 49 requêtes. Ce dernier modèle outrepasse le modèle *LM-TC_1* dans 22 requêtes. Les deux modèles présentent les mêmes résultats sur 17 requêtes. La moyenne d'amélioration obtenue par le modèle *LM-TC_1* par rapport au modèle *LM-TC_0* sur ce type de requêtes est de l'ordre de +2,28%.
- Dans la collection *AP88*, nous avons 89 requêtes de type *QTC*. Sur 43 requêtes le modèle *LM-TC_1* offre de meilleurs résultats que le modèle *LM-TC_0*. Ce dernier donne de meilleurs résultats que le modèle *LM-TC_1* sur 29 requêtes. Les deux modèles donnent des résultats équivalents sur 16 requêtes.
- Finalement, sur la collection *WT10g*, qui présente 61 requêtes de type *QTC*, nous avons obtenu les résultats suivants : le modèle *LM-TC_1* améliore les résultats du modèle *LM-TC_0* sur 38 requêtes. Ce dernier modèle outrepasse le modèle *LM-TC_1* dans 14 requêtes. Enfin, les deux modèles réalisent les mêmes performances sur 9 requêtes. La moyenne d'amélioration obtenue par le modèle *LM-TC_1* par rapport au modèle *LM-TC_0* sur cet ensemble de requêtes de type *QTC* est de l'ordre de +1,81%.

Nous avons également examiné manuellement quelques requêtes, afin de voir d'une manière claire l'impact de la nouvelle méthode de pondération proposée. Par exemple, la requête numéro 451 (*What is a **Bengals cat***) sur la collection *WT10g*. Cette requête est réduite aux termes « ***Bengals cat*** » après élimination de termes vides. Le terme « ***Bengals cat*** » est reconnu comme mot composé dans cette requête. Avec cette requête, le modèle *LM-TC_1* obtient une précision moyenne de 0,7722, les modèles *LM-TC_0* et *ULM* quant à eux obtiennent respectivement 0.7408 et 0.6006 de précision moyenne.

La table 4.8 ci-dessous présente le rang de quelques documents pertinents obtenus avec les trois modèles (sur les 1000 documents retournés). De plus, on illustre la fréquence initiale (*freq-ini*) et revisitée (*freq-rev*) du mot composé « ***Bengals cat*** » dans ces documents pertinents.

| Document | Rang des documents & fréquence de mot composé | | | | |
|---|---|---|---|---|---|
| | *LM-TC_1* | *freq-rev* | *LM-TC_0* | *freq-ini* | *ULM* |
| *WTX003-B26-249* | **2** | **11.30** | 3 | 2 | 1 |
| *WTX059-B30-262* | 15 | 1.037 | 15 | 1 | 147 |
| *WTX097-B19-147* | 19 | 1.037 | 19 | 1 | 352 |
| *WTX020-B24-89* | **23** | **2.37** | 31 | 1 | Non retrouvé |
| *WTX092-B36-89* | **24** | **2.37** | 32 | 1 | Non retrouvé |
| *WTX049-B12-27* | **25** | **2.37** | 33 | 1 | Non retrouvé |

**Table 4.8 Le rang des documents pertinents avec les modèles (*ULM, LM-TC_0, LM-TC_1*)**

A partir de cette table on peut noter les points suivants :

- Premièrement, le modèle uni-gramme ne permet pas de retrouver trois documents pertinents (*WTX020-B24-89, WTX092-B36-89* et *WTX049-B12-27*).
- Deuxièmement, quatre documents pertinents ont vu leur rang amélioré avec le modèle *LM-TC_1* comparativement au modèle *LM-TC_0*. Par exemple, le document *WTX020-B24-89* est passé du rang 31 avec le modèle *LM-TC_0* à la 23$^{ème}$ place avec le modèle *LM-TC_1*. Cette amélioration est expliquée principalement par le fait que la fréquence du mot composé « ***Bengals cat*** » dans ce document passe de la valeur 1 (fréquence initiale) à la valeur 2,37 en utilisant la fréquence revisitée.

#### 4.3.3.3 Impact du facteur $\widehat{T}$

Nous avons évalué l'impact du facteur $\widehat{T}$ introduit dans la formule (4.11). Seulement trois requêtes comprennent un mot simple contenu dans plus d'un mot composé. Les expérimentations ont été réalisées uniquement sur la collection *WT10g*. Les résultats obtenus en utilisant uniquement ces requêtes donnent une amélioration de l'ordre de +2.75% en incluant le facteur $\widehat{T}$. Mais, compte tenu du nombre réduit des requêtes utilisées, nous ne pouvons tirer aucune conclusion de cette expérimentation.

#### 4.3.3.4 Comparaison avec d'autres modèles

Nous avons aussi comparé notre modèle nommé *LM-TC*, correspondant à la version *LM-TC_1* incluant le facteur $\widehat{T}$, avec deux modèles; le premier est le modèle *MRF* (*M*arkov *R*andom *F*ield) présenté dans [**137**], plus précisément nous avons utilisé la version *S*equential *D*ependency (*MRF-SD*) de ce modèle. Le deuxième est le modèle *PLM* (*P*ositional Language *M*odels) présenté dans [**129**].

Nous avons attribué des valeurs aux paramètres de ces modèles de façon à optimiser la précision moyenne.

La table 4.9 montre la comparaison de la précision moyenne entre les différents modèles de recherche utilisés.

| | *MRF-SD* | *PLM* | *LM-TC* | *LM-TC% MRF-SD* | *LM-TC % PLM* |
|---|---|---|---|---|---|
| *WSJ90-92* | 0.1976 | 0.1987 | **0.2018** | **+2.12%** | **+1.56%** |
| *AP88* | 0.2461 | 0.2454 | **0.2519** | **+2.36%** | **+2.65%** |
| *WT10g* | 0.2215 | 0.2192 | **0.2341** | **+5.69%**$^{+}$ | **+6.79%**$^{+}$ |

**Table 4.9 Comparaison des performances des modèles (*MRF-SD, PLM et LM-TC*)**

En se basant sur nos expérimentations réalisées sur trois collections de test TREC, nous avons noté les points suivants :

- Premièrement, les deux modèles *MRF-SD* et *PLM* donnent des résultats similaires sur les trois collections;
- Deuxièmement, notre modèle donne de meilleurs résultats que les deux autres modèles (*MRF-SD* et *PLM)* et cela sur toutes les collections. Nous pouvons déduire que notre modèle améliore et le modèle bi-gramme et le modèle bi-terme [**185**], car le modèle *MRF-SD* peut générer les deux modèles (bi-gramme et bi-terme). Ceci montre que la sélection de bi-grammes pertinents et leur pondération d'une manière non uniforme peut être utile pour la RI.

**Analyse requête-par-requête**

Les figures suivantes montrent les résultats de l'analyse des performances requête-par-requête entre notre modèle (*LM-TC*) et les deux modèles : *MRF-SD* et *PLM*.

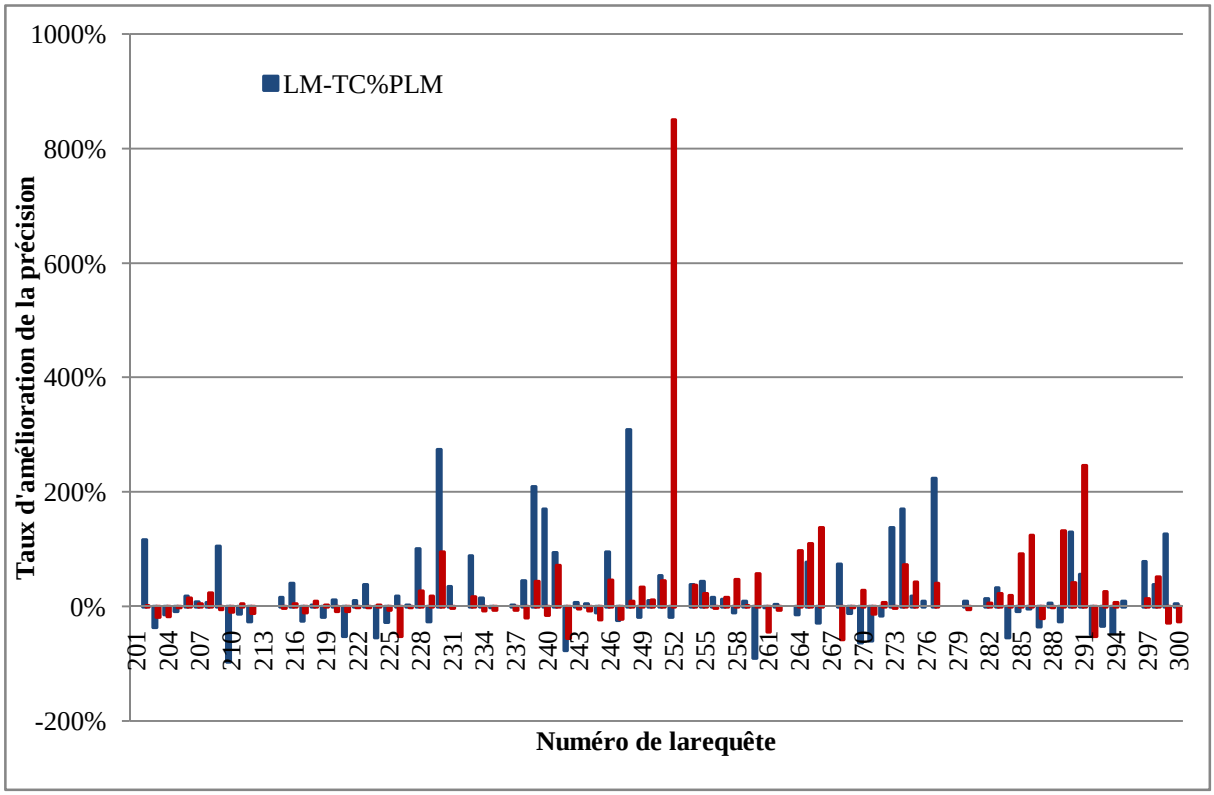



**Figure 4.8 Analyse requête-par-requête sur la collection *WSJ90-92*.**

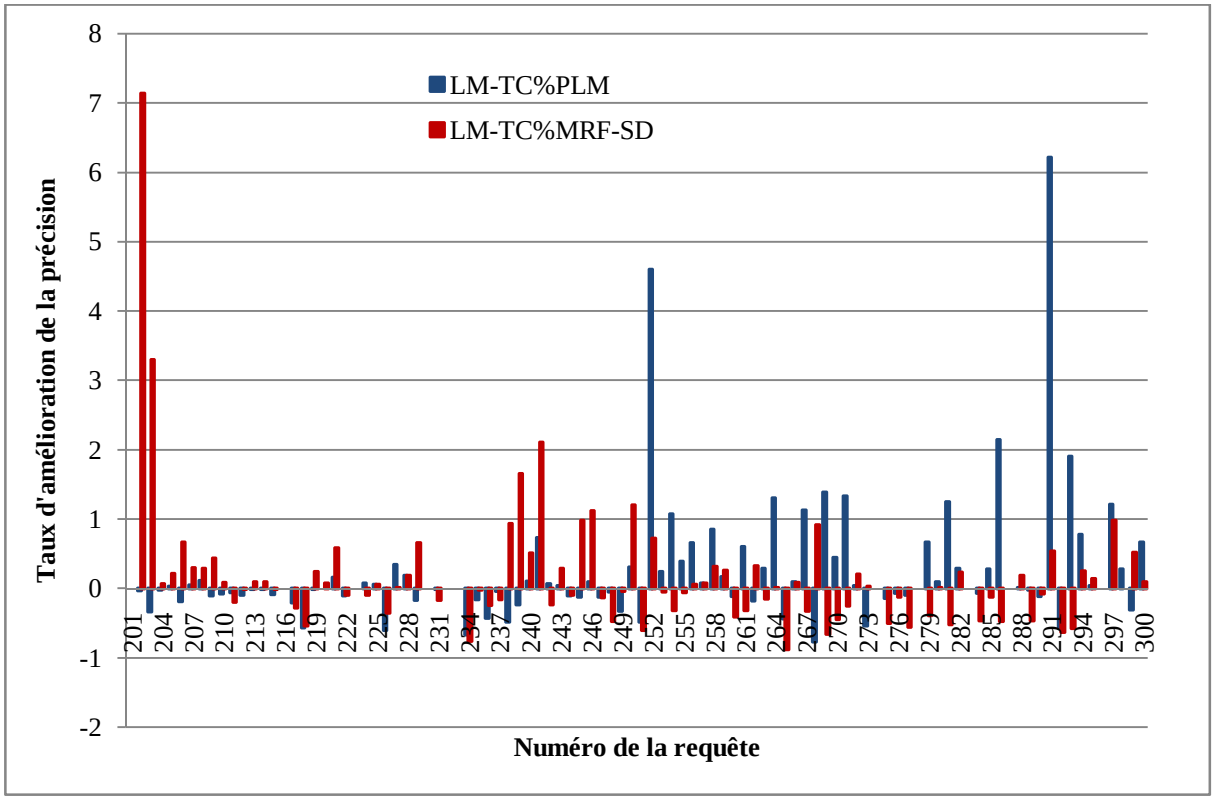



**Figure 4.9 Analyse requête-par-requête sur la collection *AP88*.**

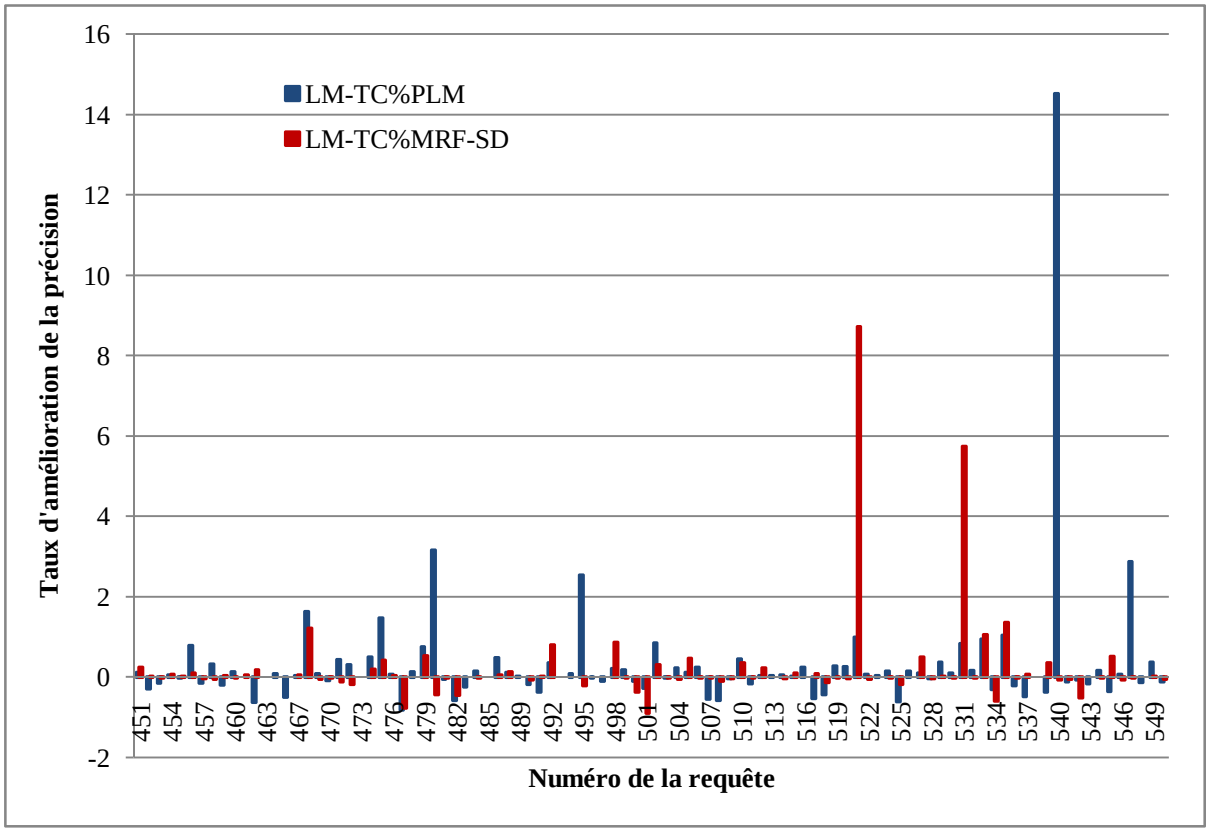



**Figure 4.10 Analyse requête-par-requête sur la collection *WT10g*.**

La comparaison des résultats de notre modèle par rapport à ceux des deux autres modèles (*MRF-SD* et *PLM*) montre que :

- Sur les collections *WSJ90-92* et *AP88* et en considérant les requêtes de type *QTC* notre modèle *LM-TC* affiche des améliorations de l'ordre de +2.4%, +2.3% (sur *WSJ90-92*) et +1.84%, +3.15% (sur *AP88*) respectivement par rapport aux modèles *MRF-SD* et *PLM*. Par contre, en utilisant les requêtes de types *Qts* notre modèle donne des résultats légèrement inférieurs que les deux autres modèles -0.76%, -0.87% (sur *AP88*) et -0.2%, -0.47% (sur *WSJ90-92*) respectivement.
- Sur la collection *WT10g* les améliorations apportées par notre modèle par rapport aux autres modèles sont conséquentes +8.46% et +12.83% et cela en utilisant les requêtes de type *QTC*. Par contre avec les requêtes de type *Qts*, notre modèle affiche des résultats légèrement inférieurs que ceux des deux autres modèles, allant de -0.01% à -3.51%. Ceci est principalement expliqué par le fait que notre modèle ne capture pas les mots composés non adjacents, ce qui est fait par les deux autres modèles.

#### 4.3.3.5 Analyse de la robustesse de notre modèle

Dans nos expérimentations, nous avons utilisé différentes valeurs pour les paramètres de notre modèle, optimisant ses performances (MAP), sur chacune des trois collections, comme le montre la table 4.3.

Afin d'évaluer la robustesse de notre modèle, nous avons appliqué les valeurs des paramètres utilisées dans la collection *AP88*, sur les deux autres collections (*WSJ90-92 et WT10g*). La table 4.10 ci-dessous illustre les résultats obtenus.

| | *MRF-SD* | *PLM* | *LM-TC* |
|---|---|---|---|
| *AP88* | 0.2461 | 0.2454 | **0.2519** |
| *WSJ90-92* | 0.1976 | 0.1987 | **0.1995** |
| *WT10g* | 0.2215 | 0.2192 | **0.2286** |

**Table 4.10 Evaluation de la robustesse de notre modèle (*LM-TC*)**

Nous constatons, comme le montre la table ci-dessus, que les résultats de notre modèle restent supérieurs à ceux des deux modèles de l'état de l'art (*MRF-SD* et *PLM*). Cela montre la robustesse de notre modèle face au choix des valeurs de ses paramètres.

## 4.4 Conclusion

Dans ce chapitre nous avons décrit notre approche qui permet de combiner les mots composés et les mots simples dans le cadre du modèle de langue. A partir des résultats d'expérimentations obtenus sur trois collections de TREC, nous pouvons tirer les conclusions suivantes :

- En se basant sur la comparaison des résultats entre la version *LM-TC_0* de notre modèle, permettant le filtrage des bi-grammes, et le modèle *BGM*, considérant tous les bi-grammes, nous pouvons conclure que le filtrage des bi-grammes est utile pour l'amélioration des performances de la recherche d'information. Cependant, ce filtrage doit être réalisé d'une manière adéquate pour assurer un apport positif.
- En se basant sur la comparaison des résultats entre la version *LM-TC*_1 de notre modèle, implémentant la méthode de pondération proposée qui se base sur la fréquence revisitée du mot composé utilisant la notion de dominance entre termes, et la version *LM-TC*_0, se basant sur l'utilisation de la fréquence initiale des mots composés pour la pondération, nous pouvons conclure que l'introduction de cette nouvelle méthode de pondération de mot composé basée sur le facteur de dominance est effective pour l'amélioration de la RI.

- L'introduction du facteur de dominance au niveau de l'appariement, donne de légères améliorations.
- L'évaluation de notre modèle *LM-TC* par rapport à deux modèles de l'état de l'art (*MRF-SD* et *PLM*) affiche de meilleurs résultats.

# Chapitre 5

# Réinjection de pertinence basée sur un modèle de langue mixte

## 5.1 Introduction

Nous avons présenté dans le chapitre précédent, un modèle de langue combinant les mots simples et les mots composés, permettant une meilleure représentation du document. Ce modèle permet de pallier au problème d'ambiguïté des termes, de fait qu'un mot composé est moins ambigu que les mots qui le compose.

Cependant, ce modèle ne permet pas de pallier au problème de disparité de termes, posé particulièrement sur le web, où le besoin en information de l'utilisateur est exprimé avec une requête contenant peu de termes [**184**]. Pour remédier à ce problème une bonne représentation de la requête est nécessaire. Dans l'objectif de construire une telle représentation, l'expansion de requête par réinjection de pertinence, est la méthode la plus utilisée.

Nous proposons dans ce chapitre une extension de notre modèle initial, pour la prise en compte de l'expansion de la requête par réinjection de pertinence. Ce nouveau modèle ainsi construit, permet prendre en charge les problèmes d'ambiguïté et de disparité des termes. Particulièrement, notre approche d'expansion de la requête est caractérisée par les points suivants **[80]**:

1. Notre approche propose une méthode originale pour le choix des termes d'expansion, qui consiste, d'une part, à ajouter non seulement les termes en relation avec un mot composé de la requête, mais aussi les termes en relation avec les mots simples qui le compose. D'autre part, à ajouter les termes en relation avec les mots composés qui contiennent un mot simple de la requête, en plus des termes liés à ce dernier.

2. Notre approche pondère les relations entre termes en utilisant la relation de cooccurrence, dans le but d'extraire (sélectionner) les termes d'expansion, comme cela est réalisé dans **[153].** Cependant, notre approche d'expansion de requêtes est basée sur la réinjection de pertinence. De plus, elle est modélisée dans le cadre du modèle de langue.
3. Dans la majorité des méthodes antérieures d'expansion de requêtes, le type de terme d'expansion utilisé est le mot simple. Cependant, l'utilisation d'unité composée lors de la phase de recherche initiale a montré qu'elle est bénéfique pour faire de la haute précision, utile dans le cas de la pseudo réinjection. Ainsi, nous explorons dans notre approche, l'utilisation d'unités composées lors de la phase d'expansion de la requête.

Le reste de ce chapitre est organisé comme suit : dans la section 5.2, nous présentons notre méthode d'expansion de requêtes basée sur le modèle de langue mixte combinant les mots simples et les mots composés et nous détaillons la façon dont les termes d'expansion sont extraits et pondérés. Nous rapportons les résultats expérimentaux dans la section 5.3. Enfin, dans la section 5.4, nous concluons notre travail.

## 5.2 Réinjection de pertinence basée sur un modèle de langue mixte

Dans cette section, nous présentons notre approche d'expansion de requêtes basée sur un modèle de langue mixte combinant les mots simples et les mots composés. Premièrement, nous introduisons brièvement la fonction d'appariement utilisée. Ensuite, nous décrivons en détails l'estimation du modèle de la requête.

Nous nous sommes intéressés dans notre modèle initial (*LM-TC)*, uniquement à l'estimation du modèle de document. Nous avons alors utilisé l'approche de génération de la requête par le modèle de document (Query Likelihood Models), comme fonction d'appariement. Cependant, cette approche ne supporte pas la modélisation de la requête. C'est pour quoi, nous avons utilisé dans notre nouveau modèle, nommé *LM-TC-QE*, l'approche de comparaison des modèles de la requête et du document. Particulièrement, la mesure de divergence de Kullback-Leibler (KL-divergence) est utilisée, et exprimée ainsi :

$$Score(Q,D) \;=\; \sum_{w \in V} P(w|Q) log P(w|D) \qquad (5.1)$$

L'estimation du modèle de document $P(w|D)$, a fait l'objet de chapitre précédent. Nous nous intéressons dans ce qui suit à l'estimation du modèle de la requête $P(w|Q)$.

### 5.2.1 Modèle de langue de la requête

La manière la plus simple pour estimer le modèle de la requête $P(w|Q)$ est l'utilisation de l'estimation du maximum de vraisemblance (*ML*) $P_{ML}(w|Q)$. Cependant, comme la requête est courte (quelques termes), il faut estimer son modèle en exploitant une source plus riche. Afin d'y parvenir, non seulement les termes exprimés dans la requête initiale, mais également les termes des documents pertinents retournés, et qui sont en relation avec ces derniers, auront une probabilité non nulle.

Pour considérer cela dans notre approche, nous proposons une méthode similaire à celle développée dans **[12]**, dans laquelle nous combinons le modèle de la requête initiale noté $P_{org}(w|Q)$ avec un autre modèle, considérant les relations entre termes noté $P_R(w|Q)$. Ainsi, le nouveau modèle de la requête est exprimé comme suit:

$$P(w|Q) = \varphi \times P_{org}(w|Q) + (1-\varphi) \times P_R(w|Q) \quad (5.2)$$

Où φ est un paramètre de lissage, permettant de pondérer l'apport de la requête initiale dans la nouvelle requête construite.

En remplaçant ce modèle dans la formule d'appariement, elle devient ainsi:

$$score(Q,D) = \sum_{w \in V}(\varphi \times P_{org}(w|Q) + (1-\varphi) \times P_R(w|Q))logP(w|D) \quad (5.3)$$

Cette formule est récrite ainsi :

$$score(Q,D) = \varphi \times \sum_{w \in Q} P_{org}(w|Q)logP(w|D) + (1-\varphi) \times \sum_{w \in DP} P_R(w|Q)logP(w|D) \quad (5.4)$$

Où $DP$ est l'ensemble de documents de réinjection.

On note que le premier terme de la formule est une sommation à travers les termes de la requête (non pas à travers tous les termes du vocabulaire), car $P_{org}(w|Q) = 0$ pour tout terme n'appartenant pas à la requête initiale.

Nous assumons que les termes reliés à la requête originale sont ceux qui apparaissent seulement dans les documents de réinjection $DP$. Nous ne considérons qu'un sous-ensemble de $DP$, noté $G$ ; où $|G| = N$ ; où $N$ est le nombre de termes à ajouter à la requête originale (i.e. les termes qui sont fortement en relation avec l'ensemble de termes de la requête initiale). Par conséquent, la dernière formule devient ainsi :

$$score(Q,D) =$$

$$\varphi \times \sum_{w \in Q} P_{org}(w|Q)logP(w|D) + (1-\varphi) \times \sum_{w \in G} P_R(w|Q)logP(w|D) \quad (5.5)$$

Nous décrivons ci-dessous, l'estimation des deux modèles $P_{org}(w|Q)$ et $P_R(w|Q)$, respectivement, le modèle de la requête initiale et le modèle prenant en compte les relations entre les termes des documents pertinents retournés et ceux de la requête initiale.

#### 5.2.1.1 Estimation du modèle de la requête initiale $P_{org}(w|Q)$

Pour estimer cette probabilité, nous supposons que la contribution d'un terme $w$ dans la requête initiale est liée à deux facteurs: (1) la fréquence du terme dans la requête, (2) le type

du terme, qui peut être simple $t$, ou composé $T$. Afin de calculer ce dernier facteur, nous considérons qu'un mot composé ajoute plus de précision à la requête qu'un mot simple, nous lions cet apport à la taille du terme. Nous exprimons alors les contributions d'un mot composé et d'un mot simple dans la requête initiale par ces deux probabilités, comme suit:

$$P_{org}(t|Q) = \frac{F(t,Q)}{|Q|}$$

et

$$P_{org}(T|Q) = \frac{F(T,Q)\times|T|}{|Q|} \quad (5.6)$$

Où $|T|$ est la longueur du mot composé $T$ et $|Q| = \sum_{T\in Q} F(T,Q) \times |T| + \sum_{t\in Q} F(t,Q)$ est la longueur de la requête.

#### 5.2.1.2 Le modèle de la requête considérant les relations entre termes $P_R(w|Q)$

Maintenant, nous présentons l'estimation du modèle de la requête utilisant les relations entre termes défini dans la formule (5.5).

Sachant que la requête $Q$ est considérée comme un ensemble de mots composés $T$ et de mots simples $t$, on peut estimer la probabilité $P_R(w|Q)$ comme suit :

$$P_R(w|Q) = \sum_{t\in Q} P_R(w|t) \times P_{org}(t|Q) + \sum_{T\in Q} P_R(w|T) \times P_{org}(T|Q) \quad (5.7)$$

Le principe de cette formule est le même que celui du modèle de traduction **[18]**. Cependant, dans **[18]**, il est utilisé pour l'expansion du modèle de document, dans notre cas il utilisé dans le contexte d'expansion de la requête. Plus précisément, le poids (probabilité) de la relation d'un terme $w$ avec l'ensemble des termes de la requête initiale, $P_R(w|Q)$, est obtenu par une sommation du produit de poids de la relation entre chaque terme (simple , $P_R(w|t)$ ou composé, $P_R(w|T)$) de la requête initiale et la contribution de ce terme (simple , $P_{org}(t|Q)$ ou composé, $P_{org}(T|Q)$) dans la requête initiale. Dans ce qui suit, nous présentons l'estimation des deux probabilités $P_R(w|t)$ et $P_R(w|T)$).

#### 5.2.1.3 Estimation de la probabilité $P_R(w|T)$

Le calcul de cette probabilité est basé sur l'hypothèse énoncée dans la section 4.2.3 (chapitre 4), considérant que l'auteur d'un document utilise les mots composants (simples) isolément

pour exprimer le mot composé comme abréviation après un nombre d'occurrences du mot composé, où elle a été utilisée pour revoir la pondération des mots composés. Ici, nous utilisons cette hypothèse pour la sélection des termes en relation avec un mot composé.

Sur la base de cette hypothèse on peut statuer que si un terme est lié à un mot composé, il peut être aussi lié à ses mots composants. Par exemple, le terme «*entropie*» lié au mot composé «*compression de données*», il est également lié au mot composant «*compression*».

Afin de prendre en compte cette hypothèse, nous proposons d'estimer la probabilité $P_R(w|T)$ comme une combinaison entre la probabilité de relation du terme $w$ avec le mot composé $T$, notée $P_{R\_directe}(w|T)$, et la probabilité de relation entre le terme $w$ avec l'ensemble des mots qui composent le terme $T$. La probabilité $P_R(w|T)$ est alors exprimée comme suit :

$$P_R(w|T) = \alpha P_{R\_directe}(w|T) + (1-\alpha)P_R\big(w|c(T)\big) \quad (5.8)$$

Où $c(T)$ est l'ensemble de mots simples composants le terme $T$. $\alpha$ est un facteur d'interpolation pour contrôler l'apport d'un mot composé par rapport à ses mots composants.

Afin d'estimer la seconde partie de la formule, nous proposons une méthode similaire à celle développée dans le modèle de traduction **[18]**. Par conséquent, la formule précédente est réécrite comme suit :

$$P_R(w|T) = \alpha P_{R\_directe}(w|T) + (1-\alpha) \times \sum_{t \in T} P_{R\_directe}(w|t)P(t|T) \quad (5.9)$$

Le calcul des deux probabilités $P_{R\_directe}(w|T)$ et $P_{R\_directe}(w|t)$ et réalisé en utilisant la formule (5.14), ces deux probabilités expriment le poids de la relation directe entre le terme $w$ et les termes $t$ et $T$ (respectivement).

L'estimation de la probabilité de dominance d'un mot simple dans son mot composé, $P(t|T)$, est formalisée dans le chapitre précédent, et exprimée par la formule (4.7), donnée en section 4.2.2.

#### 5.2.1.4 Estimation de la probabilité $P_R(w|t)$

Nous nous sommes basés sur l'hypothèse suivante pour calculer la probabilité $P_R(w|t)$ : « nous supposons que l'utilisateur lorsqu'il utilise un mot simple dans sa requête, généralement, il fait référence à un ou plusieurs mots composés ».

Par exemple, une requête contenant le terme «*énergie*», peut faire référence au mot composé «*énergie solaire*». Ainsi, nous proposons d'étendre la requête originale non seulement avec les termes liés au mot simple, mais aussi avec les termes liés aux mots composés qui contiennent ce mot simple, et cela relativement à la dominance de ce mot simple dans le mot composé et la fréquence de ce dernier dans la collection.

Afin de prendre en compte cette hypothèse, nous proposons de lisser la probabilité de relation du terme $w$ avec le mot simple $t$, notée $P_{R\_directe}(w|t)$, avec la probabilité de relation du terme $w$ dans l'ensemble des mots composés auxquels le terme $t$ appartient. La probabilité $P_R(w|t)$ est alors exprimée comme suit :

$$P_R(w|t) = \beta P_{R\_directe}(w|t) + (1-\beta)P_R\big(w|C(t)\big) \tag{5.10}$$

Où $C(t)$ est l'ensemble des mots composés auxquels le mot simple $t$ appartient et β est un paramètre de lissage qui contrôle la contribution du mot simple $t$, relativement à ses mots composés. De même que pour la formule (5.9), nous utilisons une méthode de traduction **[18]**. Nous obtenons alors la formulation suivante :

$$P_R(w|t) = \beta P_{R\_directe}(w|t) + (1-\beta) \times \sum_{t\,\in T} P_{R\_directe}(w|T)P(T|t) \tag{5.11}$$

Le calcul des deux probabilités $P_{R\_directe}(w|t)$ et $P_{R\_directe}(w|T)$ et fait en utilisant la formule (5.14). Pour estimer la probabilité $P(T|t)$ nous appliquons le théorème de Bayes, et on obtient :

$$P(T|t) = \frac{P(t|T)P(T)}{P(t)} \tag{5.12}$$

Où $P(t|T)$ est calculée en utilisant la formule (4.7), et $P(T)$ est estimée comme suit :

$$P(T) = \frac{df(T)}{\sum_{T_m \in C(t)}\big(df(T_m)\big)} \tag{5.13}$$

Où $df(T)$ est le nombre de documents contenant le mot composé $T$. $C(t)$ est l'ensemble des mots composés auxquels le terme $t$ appartient.

#### 5.2.1.5 Estimation du poids de la relation entre termes $P_{R\_directe}(w|w_j)$

L'estimation du poids de la relation entre termes consiste à calculer la probabilité $P_{R\_directe}(w|w_j)$. Comme dans de nombreuses études antérieures, nous exploitons dans notre modèle la relation de cooccurrence, cette probabilité est alors exprimée ainsi:

$$P_{R\_directe}(w|w_j) = \frac{Count(w,w_j)}{\sum_{w_i} Count(w,w_i)} \quad (5.14)$$

Où $w$, $w_i, w_j \in Q \cup DP$ et $Count(w, w_j)$ est la fréquence de cooccurrence de couple $(w, w_j)$ dans une fenêtre de texte de taille $F$.

## 5.3 Expérimentation et résultats

### 5.3.1 Collections de test et configuration expérimentale

Pour évaluer notre modèle (*LM-TC-QE*) décrit dans les sections précédentes, nous avons utilisé deux collections de test TREC *WSJ90-92* (Wall Street Journal, 1990-92) et *AP88* (Associated Press, 1988). La Table 5.1 ci-dessous montre quelques statistiques sur les collections et les requêtes utilisées. Seule la partie titre des requêtes est utilisée.

| Collection | #Documents | Requêtes d'apprentissage | Requêtes de test |
|---|---|---|---|
| *WSJ90-92* | 74520 | 101-150 | 51-100 |
| *AP88* | 79919 | 101-150 | 51-100 |

**Table 5.1 Statistiques sur les collections et les requêtes utilisées**

Pour la mise en œuvre de notre modèle nous avons suivi les mêmes étapes décrites dans le chapitre précédent, section 4.3.1. De plus, nous avons implémenté la phase d'expansion de requêtes par réinjection de pertinence.

### 5.3.2 Évaluation

Dans nos expérimentations, les modèles suivants ont été comparés:

*ULM*: modèle de langue uni-gramme.

*LM-TC*: modèle de langue mixte, présenté dans le chapitre précédent.

*LM-TC-QE*: notre modèle d'expansion de requêtes basée sur le modèle de langue mixte.

*LM-TC-QE-α=1*: est le modèle *LM-TC-QE* avec la non prise en compte du lissage dans la formule (5.9).

*LM-TC-QE-β=1*: est le modèle *LM-TC-QE* avec la non prise en compte de lissage dans la formule (5.11).

*LM-TC-QE-α=β=1*: est le modèle *LM-TC-QE* avec la non prise en compte du lissage dans les formules (5.9) et (5.11).

*KLD* : est une méthode populaire de pondération des termes d'expansion. La méthode d'expansion basée *KLD* a obtenu la meilleure valeur de Précision (MAP) sur un ensemble de méthodes standards dans TREC 2009 **[128]**.

Afin d'évaluer notre modèle et de le comparer aux autres modèles nous utilisons la mesure MAP (Mean Average Precision). En outre, nous utilisons également la précision à 10 et 20 documents (P@10, P@20) et le rappel (le nombre de documents pertinents retrouvés) comme mesures supplémentaires.

Notre modèle possède plusieurs paramètres de contrôle à affiner. Dans l'objectif de trouver les valeurs optimales de ces paramètres et une comparaison équitable entre notre modèle et les modèles Baseline, nous avons utilisé des requêtes d'apprentissage (101-150). Nous avons estimé les différents paramètres des modèles d'une manière empirique de façon à optimiser la valeur de MAP. La table 5.2 ci-dessous illustre les valeurs de ces paramètres pour les deux modèles.

| Modèles | *LM-TC-QE* | | *KLD* | |
|---|---|---|---|---|
| Collections / Paramètres | *AP88* | *WSJ90-92* | *AP88* | *WSJ90-92* |
| $\varphi$ (formule(5.5)) | 0,3 | 0,3 | - | - |
| $\alpha$ (formule(5.9)) | 0,5 | 0,3 | - | - |
| $\beta$ (formule(5.11)) | 0,5 | 0,6 | - | - |
| *DR* (nombre de documents de réinjection) | 3 | 3 | 10 | 14 |
| *N* (nombre de termes d'expansion) | 50 | 30 | 50 | 50 |
| *F* (taille de la fenêtre de texte, formule (5.14)) | 20 | 50 | - | - |

**Table 5.2 Valeurs des paramètres des deux modèles (*LM-TC-QE et KLD)***

Les tables 5.3 et 5.4 montrent la comparaison entre les différentes modèles de recherche. Afin de vérifier la significativité des résultats obtenus, nous avons effectué le test de Student et nous avons joint ($^{+}$) et ($^{++}$) pour l'indice de performance dans les différents tables des résultats lorsque le test passe respectivement 95% et 99%.

| Modèles | P @ 10 | P @ 20 | Rappel | MAP | Amélioration |
|---|---|---|---|---|---|
| *ULM* | 0,3286 | 0,3041 | 1755/2797 | 0,2471 | |
| *LM-TC* | 0, 3367 | 0, 3041 | 1840 /2797 | 0,2684 | 8,62 % $^{++}$ |
| *KLD* | 0,3612 | 0,3500 | 1 949/2797 | 0,3085 | 24,85% $^{++}$ |
| *LM-TC-QE-β=1* | 0,3980 | 0,3714 | 1978/2797 | 0,3246 | 31,36% $^{++}$ |
| *LM-TC-QE-α=1* | **0.4143** | 0.3786 | 1990/2797 | 0,3250 | 31,53% $^{++}$ |
| *LM-TC-QE-α=β=1* | 0,4041 | 0,3684 | 1974/2797 | 0,3223 | 30,43% $^{++}$ |
| *LM-TC-QE* | 0,4082 | **0,3867** | **2013/2797** | **0,3289** | **33,10%** $^{++}$ |

**Table 5.3 Résultats des différents modèles sur la collection *AP88*.**

| Modèles | P@10 | P@20 | Rappel | MAP | Amélioration |
|---|---|---|---|---|---|
| *ULM* | 0,283 3 | 0,2635 | 1461/2172 1 | 0,1971 | |
| *LM-TC* | 0,2917 | 0,2708 | 1467/2172 | 0,2063 | 4,67% |
| *KLD* | 0,3083 | 0,2760 | **1554/2172** | 0,2310 | 17,20% $^{++}$ |
| *LM-TC-QE-β=1* | 0.3167 | 0.2833 | 1533/2172 | 0,2338 | 18,62% $^{++}$ |
| *LM-TC-QE-α=1* | 0,3229 | 0,2802 | 1540/2172 | 0,2424 | 22,98 % $^{++}$ |
| *LM-TC-QE-α=β=1* | 0,3104 | 0,2813 | 1534/2172 | 0,2327 | 18,06% $^{+}$ |
| *LM-TC-QE* | **0,3292** | **0,2844** | 1540/2797 | **0,2449** | **24,25 %** $^{++}$ |

**Tableau 5.4 Résultats des différents modèles sur la collection *WSJ90-92*.**

D'après les tables ci-dessus, nous pouvons tirer les remarques et conclusions suivantes:

- Le modèle de langue mixte (*LM-TC*) améliore le modèle uni-gramme (*ULM*) en termes de précision et de rappel. Ces résultats rejoignent ceux obtenus dans le chapitre précédent et affirment que l'utilisation d'un modèle combinant les mots composés et les mots simples à l'étape recherche initiale est utile pour la RI.
- Les modèles *KLD* et *LM-TC-QE* améliorent significativement le modèle uni-gramme (*ULM*), cela reconfirme que l'expansion de requêtes a un apport effectif pour la recherche d'information. Nous avons obtenu une amélioration de précision de l'ordre de +24,25% sur la collection *WSJ90-92* et de +33,10% sur la collection *AP88*, par rapport au modèle *ULM*. De plus, notre modèle surpasse le modèle *KLD*. Ces résultats montrent que l'expansion de requête basée sur un modèle de langue mixte peut sélectionner et pondérer les termes d'expansion mieux que le modèle d'expansion *KLD*.

- Enfin, le modèle *LM-TC-QE* donne de meilleurs résultats que ses trois autres versions (*LM-TC-QE*-α=1, *LM-TC-QE*-β=1, *LM-TC-QE*-α=β=1) où le lissage est ignoré dans les formules (5.9), (5.11) et (5.9) (5.11), respectivement. Ceci montre qu'il est intéressant d'ajouter non seulement les termes qui sont liés au mot simple de la requête, mais aussi les termes qui sont liés aux mots composés qui contiennent ce mot simple. Et inversement ajouter non seulement les termes qui sont liés à un mot composé de la requête, mais aussi les termes qui sont liés à ses mots composants est aussi intéressant.

**Analyse requête-par-requête**

Afin de mieux comprendre l'apport cette nouvelle méthode d'expansion de requêtes, nous avons effectué une analyse requête-par-requête. Nous récapitulons les résultats de cette analyse sur les graphiques ci-dessous :

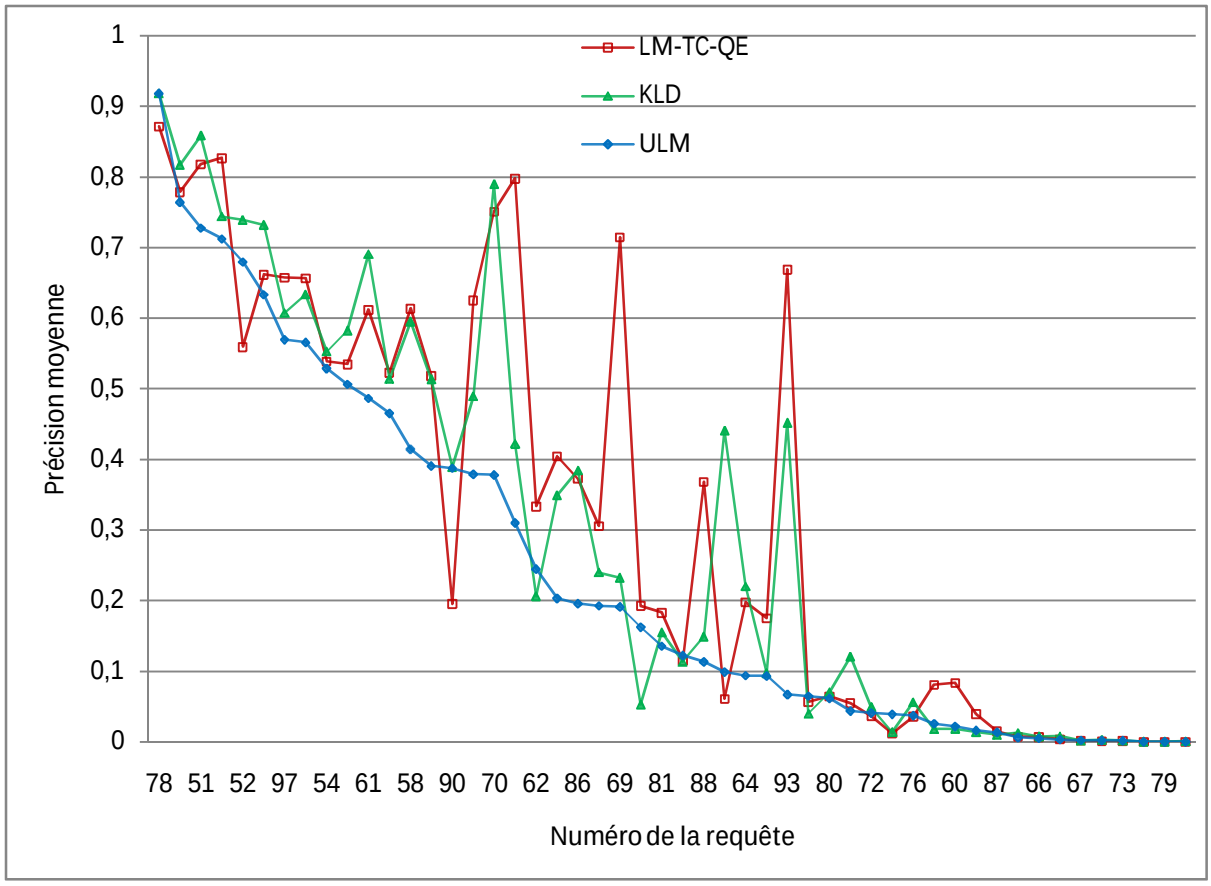



**Figure 5.1 Résultats requête-par-requête sur la collection *AP88*.**

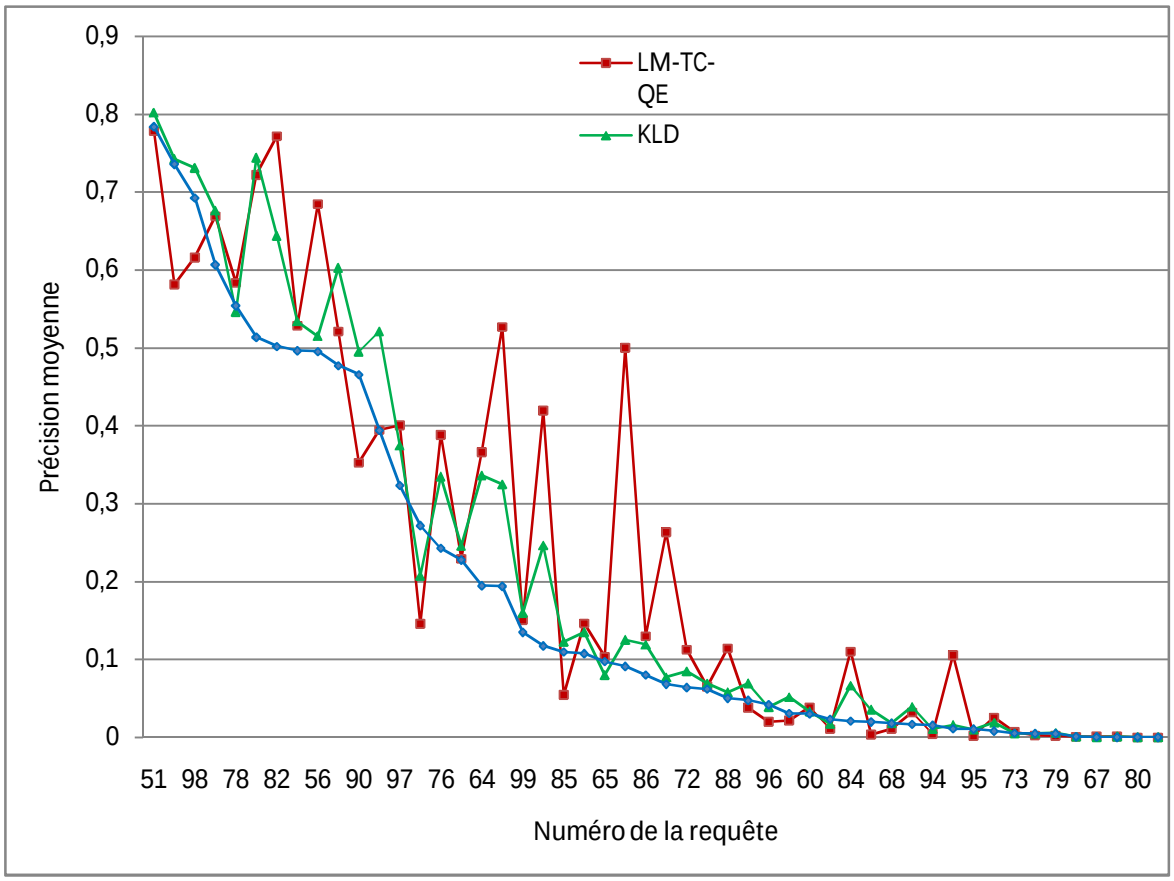



**Figure 5.2 Résultats requête-par-requête sur la collection *WS90-92*.**

A partir de ces deux graphiques nous constatons les remarques suivantes :

1. Les résultats obtenus sur la collection *AP88* montrent que :
    - Notre modèle (*LM-TC-QE*) donne de meilleures précisions sur 35 requêtes par rapport au modèle uni-gramme. Ce dernier obtient de meilleurs résultats sur 12 requêtes. Enfin, les deux modèles réalisent la même précision sur 3 requêtes.
    - Notre modèle améliore le modèle *KLD* sur 24 requêtes. Ce dernier donne de meilleurs résultats sur 24 requêtes, enfin, sur les deux(2) requêtes restantes, les deux modèles réalisent le même résultat.
2. Les résultats obtenus sur la collection *WSJ90-92* montrent que :
    - Notre modèle donne de meilleures précisions sur 31 requêtes par rapport au modèle uni-gramme. Ce dernier obtient de meilleurs résultats sur 17 requêtes. Enfin, les deux modèles réalisent la même précision sur deux(2) requêtes.

- Notre modèle améliore le modèle *KLD* sur 21 requêtes. Ce dernier donne de meilleurs résultats sur 26 requêtes, sur les 3 requêtes restantes. Enfin, les deux modèles réalisent le même résultat.

Dans l'objectif d'avoir une vue plus fine et plus précise de l'impact de la nouvelle méthode d'expansion proposée, Nous présentons ci-dessous un exemple de requête numéro 88: « *Topic: Crude Oil Price Trends* », où nous montrons les premiers termes d'expansion de la requête sélectionnés par les deux modèles, sur la collection *WSJ90-92*.

| **Modèle *LM-TC-QE*** | **Modèle *KLD*** |
|---|---|
| *opec* | *opec* |
| *joint meet* | *barrel* |
| ***opec member*** | *produc* |
| *price committe* | *meet* |
| ***opec produc*** | *cartel* |
| ***iran iraq*** | *cent* |
| *barrel* | *output* |
| ***world oil*** | ***petroleum*** |
| *sourc* | *lukman,* |
| *ink* | *suppli,* |
| *group* | *committe* |
| *newspap* | ***energi*** |
| ***oil produc*** | ***gallon*** |
| …….. | …….. |

**Table 5.5 Termes d'expansion (lemmatisés) générés par les modèles *LM-TC-QE* et *KLD***

Comme on peut le voir dans la table 5.5, notre modèle d'expansion permet de sélectionner de nouveaux termes pertinents (**en gras**) que ceux sélectionnés par le modèle *KLD*. Par exemple, le mot composé «***opec production***» est plus précis que les termes «*opec*» et «*production*», sélectionnés séparément par le modèle *KLD*. Par conséquent, en utilisant cette requête notre modèle obtient une meilleure précision que le modèle *KLD*. Précisément, notre modèle obtient une précision moyenne égale à 0.1143, et les modèles *KLD* et *ULM* obtiennent respectivement : 0.0577 et 0.0501 de précision moyenne.

## 5.4 Conclusion

Dans ce chapitre, nous avons décrit une nouvelle approche d'expansion de la requête basée sur un modèle de langue mixte combinant les mots simples et les mots composés. Les expérimentations effectuées sur deux collections TREC ont montré que, d'une part, le modèle mixte combinant les mots composés et les mots simples améliore les résultats du modèle unigramme. Ce qui rejoint les résultats obtenus dans le chapitre précédent. D'autre part, notre modèle d'expansion basé sur le modèle mixte améliore significativement le modèle unigramme et affiche de meilleurs résultats que l'un des modèles de l'état de l'art, nommément la méthode d'expansion *KLD*.

# Chapitre 6

# Introduction de l'importance d'un site dans le calcul de la pertinence a priori d'une page

## 6.1 Introduction

La plus part des études menées sur l'estimation de la probabilité a priori d'une page web $P(d)$ (formule (6.1)), se sont concentrées sur les caractéristiques liées à ces pages uniquement **[62] [90] [106] [121] [214]**. Cela, néglige le fait qu'une page web fait partie d'un site web qui lui à son tour fait partie du web **[79][85]**. Nous explorons dans ce chapitre l'idée de l'intégration des caractéristiques du site dans le calcul de la probabilité a priori de la page web, sous l'hypothèse que dans la plupart des cas les auteurs des pages web référencent la page principale (site) au lieu de référencer la page exacte (la page concernée). Les facteurs tels que, le nombre de liens entrants et le *PageRank* ne suffise pas alors à refléter l'importance réelle de la page dans l'espace web. Autrement dit, les pages provenant de sites importants doivent avoir une plus grande priorité que celles provenant de sites moins importants.

Dans ce chapitre nous proposons un modèle qui permet d'intégrer une caractéristique du site (nombre de liens entrants) dans l'estimation de la probabilité a priori d'une page web. Une fois cette probabilité est calculée, nous la combinons avec le score obtenu par le contenu de la page web. Cette combinaison des deux évidences est réalisée sous le cadre du modèle de langue. Afin de valider notre idée, nous avons effectué des tests sur la collection TREC « *.GOV* » ; où nous avons comparé les différentes versions de notre modèle avec deux modèles : le modèle uni-gramme qui ne considère que le contenu de la page et le modèle combinant le contenu d'une page web et la probabilité à priori de la page obtenu en utilisant le nombre de liens entrants a cette page. Les résultats obtenus montrent que notre modèle est prometteur.

Le reste de ce chapitre est structuré comme suit : dans la section 6.2, nous décrivons notre méthode d'introduction de l'importance d'un site web dans le calcul de la probabilité a priori d'une page web, où trois versions sont définies. Les détails de l'implémentation de notre approche, ainsi que les résultats de nos expérimentations sont donnés dans la section 6.3. Enfin, la section 6.4 fait la synthèse de ce chapitre.

## 6.2 Introduction de l'importance d'un site dans le calcul de la pertinence a priori d'une page

L'estimation du score d'un document $d$ vis-à-vis d'une requête $q$ est réalisée dans le modèle de langue comme suit :

$$P(q|d) = P(d) \prod_{t_i \in Q} P(t_i|d) \quad (6.1)$$

Où le facteur $P(d)$ représente la probabilité a priori de pertinence d'un document. Cette probabilité est ignorée dans le classement des documents si aucune caractéristique sur le document n'est utilisée pour l'estimer. Par contre, si une caractéristique sur un document est utilisée pour estimer sa probabilité a priori de pertinence alors les documents de la collection n'ont pas la même probabilité a priori. Ce score est alors utilisé dans le classement des documents à coté du score obtenu en utilisant le contenu de ce document ; i.e. $P(t_i|d)$.

L'évaluation du modèle de document $P(t_i|d)$ peut être réalisée en utilisant n'importe quel modèle de langue uni-gramme. Dans ce travail le lissage Dirichlet est utilisé.

En partant de l'intuition qu'un site important (référencé par beaucoup d'autres sites) procure une information plus pertinente qu'un site moins important. Nous introduisons le facteur importance du site dans le calcul de la probabilité a priori de pertinence d'une page web $P(d)$.

Avant de décrire notre modèle, nous définissons ci-dessous les termes clés utilisés.

***Un site web (page d'entrée)*** : est une page dont l'URL contient uniquement, soit le nom du domaine ou un de sous domaine. Par exemple :

www.nist.gov/ et expect.nist.gov/ : sont deux URL de site web.

***Une page web*** : est une page dont l'URL contient un nom du domaine ou un nom du domaine ou sous domaine suivi d'un ou plusieurs répertoires et se terminant (ou non) par un nom de fichier. Par exemple, les deux sites web (URL) illustrés dans l'exemple précédent contiennent respectivement les pages suivantes :

www.nist.gov/srd/jpcrd_28.htm et expect.nist.gov/scripts/faxstat

***Importance d'une page (site) web*** : l'importance d'une page web est mesurée par le nombre de liens pointant cette page. Tous les liens sont pris en compte, les liens inter-site et liens intra-site.

### 6.2.1 Première version

Dans cette version de notre modèle, nous nous sommes basé sur le postulat suivant : « les page d'un site web permettent de détailler le contenu de la page d'accueil (site) », pour estimer la probabilité a priori d'une page web . Nous proposons alors que l'importance d'un site web est héritée équitablement par l'ensemble des pages web de ce site. De ce fait on estime la probabilité a priori de pertinence d'une page web $P(d)$ ainsi :

$$P(d) = C[\lambda((Nb1)/N) + (1-\lambda)Nb2] \quad (6.2)$$

Où :

$Nb1$ est le nombre de liens pointant le site « page principale » ;

$Nb2$ est le nombre de liens pointant la page concernée (p) ;

$N$ est le nombre de pages dans le site contenant la page (p) ;

$C$ est une constante ;

λ est un paramètre de lissage compris entre 0 et 1.

***Exemple illustratif :***

Soit quatre (4) pages, P1, P2, P3 et P4 appartenant respectivement aux sites S1, S1, S2, S3. La table ci-dessous montre le calcul de la probabilité a priori d'une page suivant la formule (6.2), et l'impact de l'exploitation de cette probabilité dans le classement a priori des pages web.

| Pages | Sites | ***Nb1*** | ***Nb2*** | ***N*** | λ | ***P(d)*** | ***Rang1*** | ***Rang2*** |
|---|---|---|---|---|---|---|---|---|
| P1 | S1 | 5 | 5 | 10 | 0.5 | 2.75 | 2 | 3 |
| P2 | S1 | 5 | 2 | 10 | 0.5 | 1.25 | 3 | 4 |
| P3 | S2 | 40 | 6 | 10 | 0.5 | 5 | 1 | 2 |
| P4 | S3 | 200 | 1 | 20 | 0.5 | 5.5 | 4 | 1 |

Tel que Rang1 est le rang de la page en considérant uniquement son importance ($Nb2$).
Rang2 est le rang de la page en utilisant la formule (6.2).

Nous pouvons remarquer que le rang des pages change ; par exemple la page P4 classée 4ème a passé à la 1ère place, car elle appartient à un site important (200 liens entrants).

### 6.2.2 Seconde version

Le deuxième cas que nous allons explorer est basé sur l'hypothèse que l'information dans la page d'accueil (site) est souvent détaillée dans les pages descendantes directes. Ainsi, la

formule (6.2) est réécrite comme suit et appliquée uniquement pour estimer la probabilité a priori de pertinence des pages se trouvant au niveau un (descendantes directes du site):

$$P(d) = C[\lambda((Nb1)/N_p) + (1-\lambda)Nb2] \quad (6.3)$$

Où $N_p$ est le nombre de pages descendantes directes du site. Les autres paramètres sont identiques à ceux de la formule (6.2).

### 6.2.3 Troisième version

En partant de l'hypothèse que deux pages ayant un contenu sémantique similaire devraient avoir une pertinence assez proche pour un même sujet. Alors le fait qu'une page a un contenu informationnel similaire à celui de son site lui permet de bénéficier de l'importance de site plus que les autres pages. Ainsi nous formulons la probabilité a priori de pertinence d'une page comme suit :

$$P(d) = C[\left(\frac{sim(d,S)}{\sum_{d_i \in S} sim(d_i,S)}\right) Nb1 + (1 - \frac{sim(d,S)}{\sum_{d_i \in S} sim(d_i,S)})Nb2] \quad (6.4)$$

Où $sim(P_i, S)$ est la similarité sémantique entre la page $P_i$ et le site auquel elle appartient.

***Remarque :*** pour le calcul de la similarité entre une page et son site, nous utilisons la mesure de cosinus, exprimée comme suit:

$$sim(P_i, S) = \frac{S.P_i}{\|S\|.\|P_i\|} = \frac{\sum_{j=1}^{|T|} w_{Sj}.w_{P_ij}}{\sqrt{\sum_{j=1}^{|T|} {w_{Sj}}^2 \times \sum_{j=1}^{|T|} {w_{ij}}^2}} \quad (6.5)$$

Où $w_{Sj}$ est le poids du terme $t_j$ dans le document (site web $S$), $w_{P_ij}$ est le poids du terme $t_j$ dans le document (page web $P_i$) et $|T|$ est le nombre de termes dans la collection.

## 6.3 Expérimentations et résultats

Nous présentons dans cette section notre environnement d'expérimentation : les étapes d'implémentation du système, les outils utilisés, les collections et topics considérés et les mesures d'évaluation adoptées. Ensuite, Nous présentons les résultats expérimentaux obtenus pour chaque une des versions de notre modèle, ainsi que l'analyse et l'évaluation de ces résultats.

### 6.3.1 Environnement d'évaluation

Pour l'implémentation de notre modèle nous avons développé des programmes structurés ainsi :

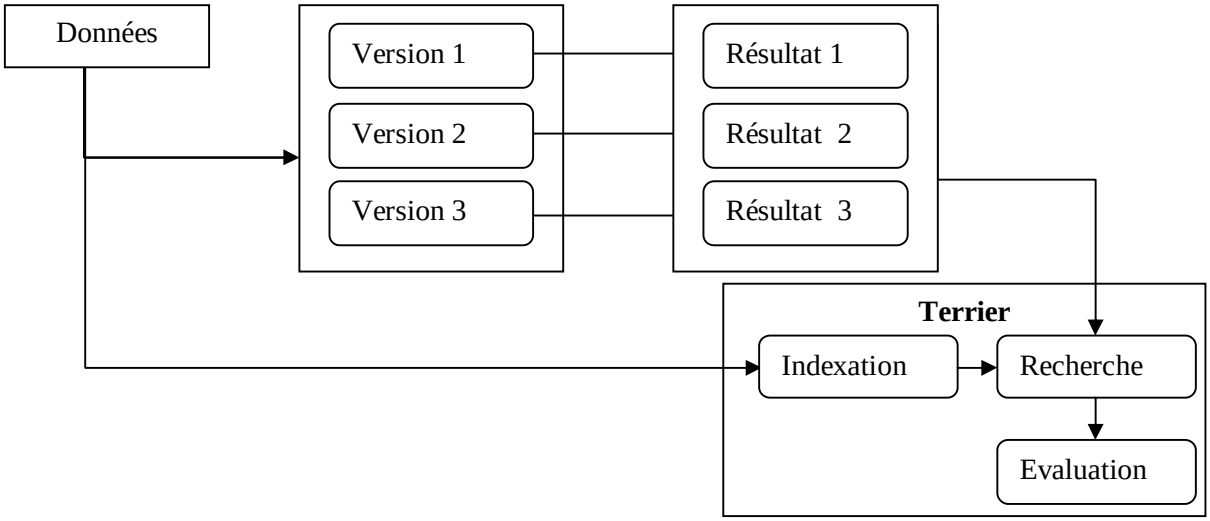



**Figure 6.1 Architecture globale de notre système**

#### 6.3.1.1 Les Données manipulées

Pour chaque une des versions décrites dans la section précédente nous avons utilisé les données suivantes :

- ***La collection .GOV*** : est une collection web de taille de vingt giga octets et comporte 1,25 millions de documents. La structure de liens entre pages de cette collection est exprimée à l'aide de deux fichiers décrit ci-dessous. Ces deux fichiers sont fournis avec cette collection.
- ***Le fichier url_id*** : ce fichier comporte sur chaque ligne l'URL de la page web et son identificateur, comme illustré sur la figure suivante.


**Figure 6.2 Un extrait du fichier *url_id***

- ***Le fichier links_id :*** ce fichier comporte sur chaque ligne l'identificateur de la page web, et l'identificateur de la page web pointée par la première (la première page possède un lien sortant vers la deuxième). la figure suivante illustre un extrait de ce fichier.

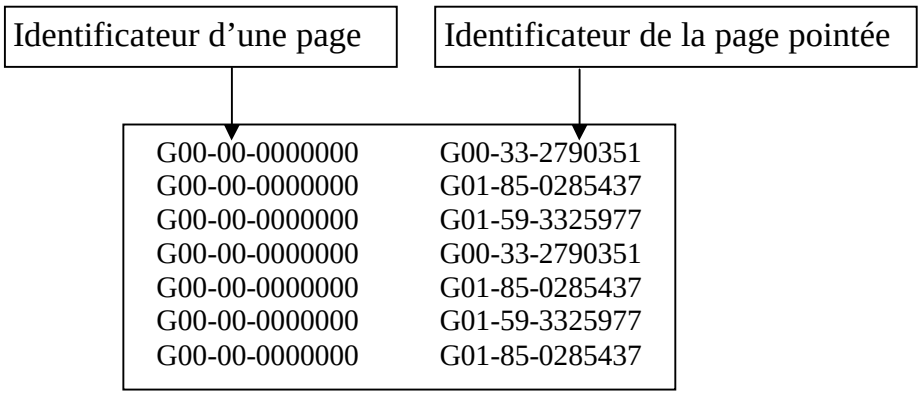



**Figure 6.3 Un extrait du fichier *links_id***

#### 6.3.1.2 Implémentation de la première version

Pour implémenter la première version de notre modèle nous avons développé deux programmes. Le premier programme permet de calculer pour chaque page web le nombre de liens entrants à cette page (NbrLp), le nombre de liens entrants à son site (NbrLs), le nombre de pages dans le site (NbrPs) et la profondeur de la page dans le site (Profp). Nous avons sauvegardé le résultat de ce premier programme dans une table à cinq champs, comme illustrée ci-dessous :

| doc_id | NbrLp | NbrLs | NbrPs | Profp |
|---|---|---|---|---|
| G17-43-1663516 | 1 | 1020 | 1495 | 4 |
| G17-43-1698865 | 2 | 414 | 16214 | 4 |
| G17-43-1710495 | 2 | 373 | 915 | 2 |
| G17-43-1736807 | 1 | 6996 | 2010 | 1 |
| G17-43-1749009 | 1 | 41 | 1218 | 4 |
| G17-43-1806271 | 2 | 555 | 910 | 4 |
| G17-43-1818987 | 1 | 57 | 28 | 2 |
| G17-43-1833640 | 1 | 8672 | 14545 | 4 |

**Table 6.1 Un extrait de la table (nombre de liens, nombre de pages, etc.)**

Le second programme prend en entrée la table retournée par le premier programme et calcule la pertinence a priori de chaque page web. La formule utilisée pour le calcul de la pertinence a priori est la formule (6.2) à laquelle, on applique le logarithme (log) ; et on fait varier le facteur λ entre 0.1 et 0.9. Comme résultat nous obtenons neuf (9) fichiers « Prior ». Chaque fichier contient l'identificateur de la page et la pertinence a priori de la page.

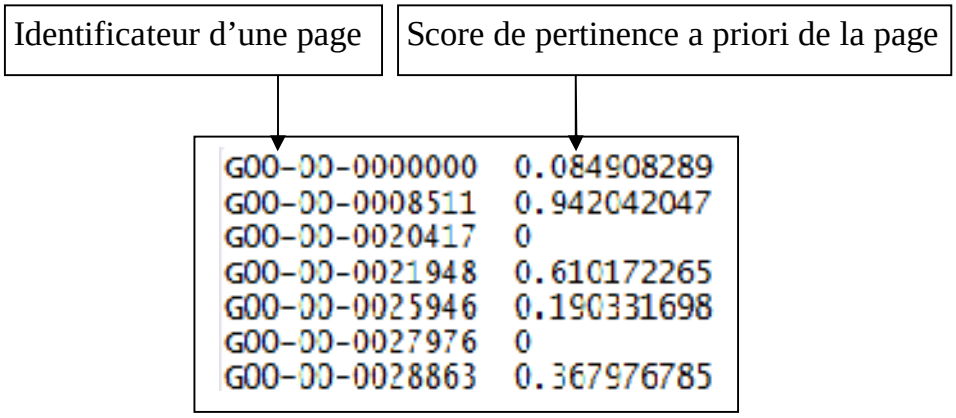



**Figure 6.4 Un extrait du fichier Prior0.1.**

#### 6.3.1.3 Implémentation de la seconde version

Deux programmes ont été développés pour mettre œuvre cette seconde version de notre modèle. Le premier programme utilise les données citées auparavant, il permet de calculer pour chaque page le nombre de pages de niveau un (1) dans son site (Nbrpsp1). Le résultat de ce programme est une table à deux champs contenant les identificateurs des pages web et le nombre de pages web de niveau un dans le site pour chaque page. La figure suivante illustre un extrait du la table obtenue.

| doc_id | Nbrpsp1 |
|---|---|
| G00-00-0000000 | 15 |
| G00-00-0008511 | 18 |
| G00-00-0020417 | 0 |
| G00-00-0021948 | 26 |
| G00-00-0025946 | 0 |

**Table 6.2 un extrait de la table (nombre de pages au niveau un du site).**

Le second programme calcule la pertinence à priori de chaque page, pour y faire on utilise les tables (Table 6.1 et Table 6.2) et on applique la formule (6.3). Ce programme retourne neuf fichiers Prior (selon la valeur de λ comprise entre 0.1 et 0.9). Chaque fichier contient sur chaque ligne l'identificateur de la page, et la pertinence à priori de cette page.

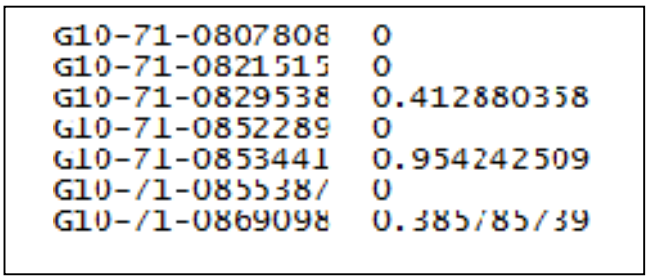



**Figure 6.5 Un extrait du fichier Prior_0.3.**

#### 6.3.1.4 Implémentation de la troisième version

Pour mettre en œuvre cette troisième version de notre modèle, on a mis au point deux programmes ; le premier permet de calculer la similarité sémantique entre les pages et leur site. Le résultat de ce programme est une table à deux champs : l'identificateur de la page et la valeur de sa similarité avec son site, comme illustré dans la table suivante :

| doc_id | Sim |
|---|---|
| G01-75-3567356 | 0.318661586966504 |
| G01-75-3575967 | 0.168041591079845 |
| G01-75-3579205 | 0.04675737879600171 |
| G01-75-3589312 | 0.476062712568625 |
| G01-75-3596710 | 0.436708513129001 |
| G01-75-3619167 | 0.196933532515712 |
| G01-75-3632288 | 0.662479807615892 |

**Table 6.3 Un extrait de la table de similarité (page/site)**

Le second programme permet de construire neuf (9) fichiers Prior (selon la valeur de λ). Chaque fichier contient sur chaque ligne l'identificateur de la page et la pertinence a priori de la page. Ce programme utilise les tables (Table 6.1 et Table 6.3) et applique la formule (6.4). La figure suivante illustre un extrait d'un fichier Prior.

```
G26-71-1565555 1.2604167801415278
G26-71-1567753 1.270039635546034
G26-71-1575072 0.9162907318741551
G26-71-1579867 0.9162907318741551
G26-71-1581520 0.4054651081081644
G26-71-1613126 1.4683245920723798
G26-71-1633708 0.9264576626791408
G26-71-1736116 0.924066281693018
G26-71-1740612 1.0954465024869418
G26-71-1756655 2.351478949486156
G26-71-1805102 0.9300256161833313
G26-71-1833790 1.4433828577429766
G26-71-1838333 0.9623958236455686
```

**Figure 6.6 Un extrait du fichier Prior_0.5.**

### 6.3.2 Résultats expérimentaux

#### 6.3.2.1 Collections de test et configuration expérimentale

L'évaluation de notre modèle décrit dans la section précédente est réalisée en utilisant la collection de test TREC *.GOV*. La Table 6.4 ci-dessous montre quelques statistiques sur la collection et les requêtes utilisées. Seule la partie titre des requêtes est utilisée.

| Collection | #Documents | Requêtes de test |
|---|---|---|
| *.GOV* | 1247753 | 1-225 (web query 2004) |

**Table 6.4 Statistiques sur les collections et les requêtes utilisées**

L'indexation, la recherche et l'évaluation sont réalisées en utilisant la plate-forme Terrier **[131]**, où les mots vides sont éliminés et l'algorithme de Porter [**152**] est utilisé en indexation et en recherche.

#### 6.3.2.2 Evaluation

Nous avons utilisé les modèles suivants dans nos expérimentations :

*ULM* : dans ce modèle la probabilité a priori de pertinence est ignorée. Le modèle de document est un modèle uni-gramme basé sur le lissage de Dirichlet.

*ULM_IN* : modèle uni-gramme basé sur le lissage de Dirichlet, utilisant le nombre de liens entrants (*IN*) pour estimer la probabilité a priori de document.

*ULM_APP1* : modèle uni-gramme basé sur le lissage de Dirichlet, utilisant la formule (6.2) (première version) pour estimer la probabilité a priori de document.

*ULM_APP2* : modèle uni-gramme basé sur le lissage de Dirichlet, utilisant la formule (6.3) (seconde version) pour estimer la probabilité a priori de document.

*ULM_APP3* : modèle uni-gramme basé sur le lissage de Dirichlet, utilisant la formule (6.4) (troisième version) pour estimer la probabilité a priori de document.

L'estimation de la pertinence finale (score) d'un document est réalisée, pour tous les modèles intégrant la probabilité a priori comme suit : initialement, on effectue un classement des documents en tenant compte uniquement de leurs contenus, ensuite on effectue un reclassement des mille (1000) premiers documents retournés et cela en utilisant les deux évidences : le contenu de document et sa probabilité a priori.

Afin d'évaluer notre modèle et de le comparer aux autres modèles nous utilisons la mesure MAP (Mean Average Precision). Afin, de vérifier la significativité des résultats obtenus, nous avons effectué le test de Student et nous avons joint « $^{+}$ » et « $^{++}$ » pour l'indice de performance dans la table des résultats lorsque le test passe respectivement 95% et 99%.

La table ci-dessous montre les résultats obtenus avec les différents modèles cités précédemment.

| Modèle | MAP | Amélioration |
|---|---|---|
| *ULM* | 0,2404 | |
| *ULM_IN* | 0,2709 | 12,69%$^{++}$ |
| *ULM_APP1* | 0,2775 | 15,43% $^{++}$ |
| *ULM_APP2* | 0,2727 | 13,44% $^{++}$ |
| *ULM_APP3* | **0,2801** | **16,51 %** $^{++}$ |

**Table 6.5 Résultats des différents modèles sur la collection *.GOV*.**

A partir de la table suivante nous tirons les remarques et conclusions suivantes :

- Premièrement, on remarque que l'utilisation de nombre de liens entrants comme seconde évidence a amélioré significativement le résultat obtenu avec le modèle uni-gramme (*ULM*), qui considère uniquement le contenu textuel du document dans le classement des documents, l'amélioration constatée est de l'ordre de +12,69%. Ceci indique qu'une page populaire (ayant un nombre de liens entrants important) est plus probable d'être pertinente qu'une page moins populaire.
- Deuxièmement, l'introduction d'une caractéristique de site (nombre de liens entrants) contenant une page dans le calcul de la probabilité a priori de cette page, permet d'améliorer le résultat obtenu avec le modèle considérant uniquement la caractéristique de la page (nombre de liens entrants), et cela avec les différentes versions proposées. Ceci indique que le calcul de la probabilité a priori d'une page est mieux estimé lorsque les caractéristiques de son site sont prises en compte.

- Troisièmement, on remarque que la troisième version de notre modèle donne le meilleur résultat (+16,51%) par rapport aux versions une (1) et deux (2), dans les quelles on a obtenu respectivement des améliorations de +15,43% et +13,44%. On peut donc en déduire qu'une page doit hériter des caractéristiques de son site et cela selon sa similarité sémantique avec son site. autrement dit une page traitant de la même thématique que son site doit hériter plus des caractéristiques de son site qu'une page traitant une thématique différentes de son site.

**Analyse requête-par-requête**

Afin d'avoir une vision plus fine et détaillée des améliorations obtenues par notre modèle, on a effectué une analyse requête-par-requête. Le graphique ci-dessous présente cette analyse.

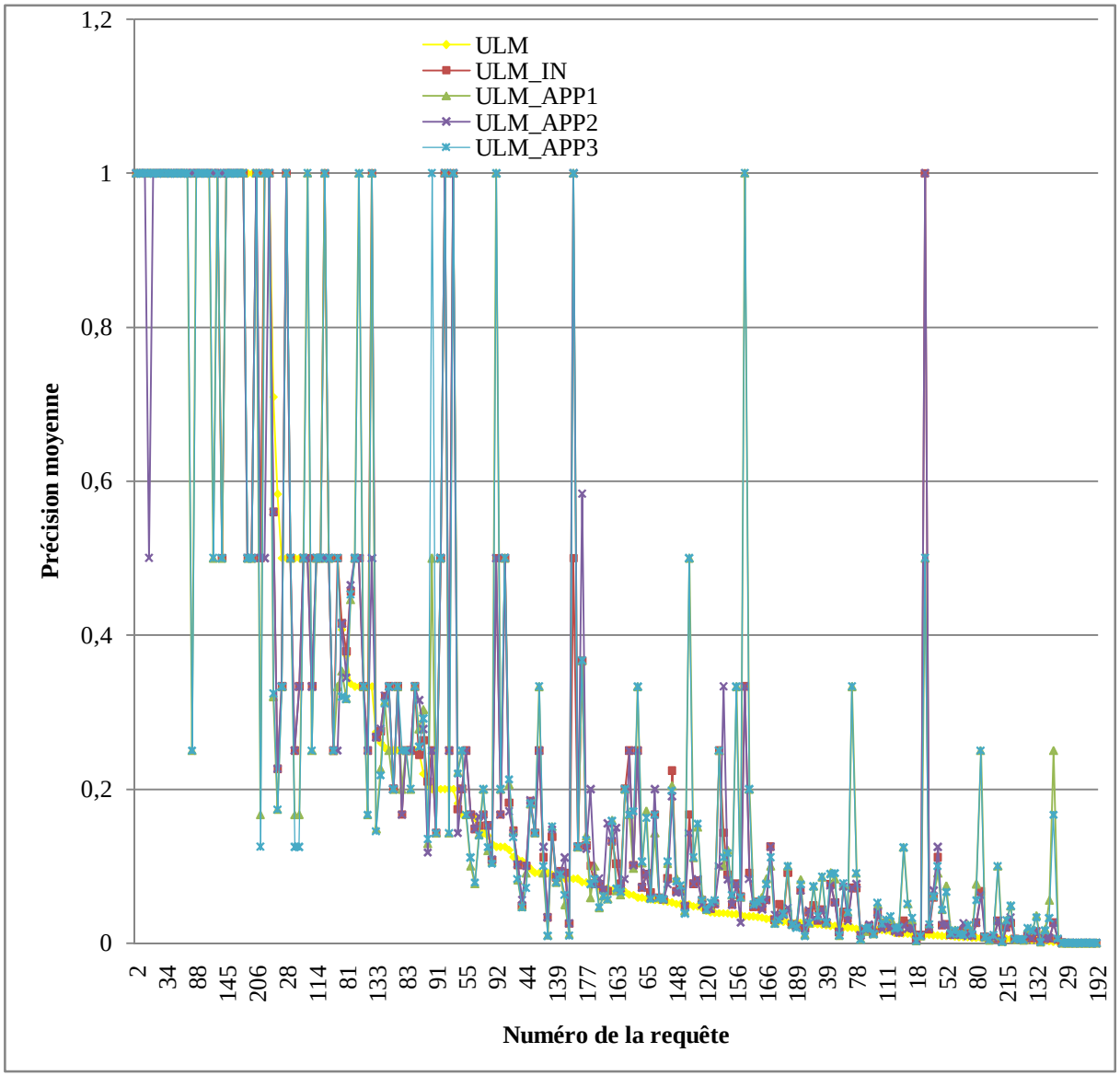



**Figure 6.7 Résultats requête-par-requête avec les différents modèles de recherche.**

À partir des résultats obtenus nous avons noté les remarques suivantes :

- La première version de notre modèle améliore le modèle *ULM_IN* sur 86 requêtes ; ce dernier donne de meilleurs résultats sur 72 requêtes ; les deux modèles présentent des résultats équivalents sur 67 requêtes.

- La seconde version de notre modèle améliore le modèle *ULM_IN* sur 57 requêtes ; ce dernier donne de meilleurs résultats sur 52 requêtes ; les deux modèles présentent des résultats équivalents sur 116 requêtes.

- La troisième version de notre modèle améliore le modèle *ULM_IN* sur 87 requêtes ; ce dernier donne de meilleurs résultats sur 69 requêtes ; les deux modèles présentent des résultats équivalents sur 69 requêtes.

Dans l'optique de comprendre les améliorations constatées, nous avons analysé manuellement quelques requêtes. Ci-dessous, nous illustrons les détails du résultat de la requête numéro WT04-50 (money laundering).

| Document | *ULM* | *ULM_IN* | *ULM_APP1* | *ULM_APP2* | *ULM_APP3* |
|---|---|---|---|---|---|
| *G09-93-3919157* | 4 | 2 | 3 | 3 | 1 |
| *G10-04-2315842* | 14 | 9 | 6 | 7 | 6 |
| *G07-70-0000000* | 87 | 53 | 37 | 47 | 36 |
| *G35-71-1435292* | 117 | 109 | 106 | 111 | 102 |
| *G35-97-3029817* | 126 | 121 | 127 | 127 | 129 |
| *G22-50-2198391* | 252 | 287 | 305 | 304 | 300 |
| *G21-40-2003687* | 396 | 421 | 449 | 435 | 458 |
| *G08-69-0634583* | 458 | 580 | 692 | 677 | 728 |
| *G05-05-1028021* | Non retrouvé | Non retrouvé | Non retrouvé | Non retrouvé | Non retrouvé |
| **MAP** | 0,0622 | 0,1009 | 0,0969 | 0,1009 | 0,1712 |

**Table 6.6 Le rang des documents pertinents avec les différents modèles**

A partir de cette table on peut voir que la troisième version de notre modèle (*ULM_APP3)* permet d'améliorer le rang des documents pertinents comparativement aux modèles *ULM* et *ULM_IN*. Par exemple, le document *G09-93-3919157* a passé du rang 4 avec le modèle *ULM* à la 2ème place avec le modèle *ULM_IN* et à la 1ère place avec notre modèle *ULM_APP3*. Cette amélioration est expliquée principalement par le fait que le document *G09-93-3919157* contient 4 liens entrants, son site contient 589 liens entrants et ce document est similaire avec son site (0,0024 : valeur normalisée).

## 6.4 Conclusion

Nous avons proposé dans ce chapitre une approche permettant d'intégrer l'importance d'un site web dans le calcul de la probabilité a priori de pertinence d'une page web, en assignant plus de confidence aux pages provenant des sites importants (intéressants) que pour celles provenant des sites moins importants.

Les expérimentations effectuées sur la collection *.GOV* ont montré que la prise en compte de l'importance d'un site web dans l'évaluation de la probabilité a priori de pertinence d'une page web améliore significativement le modèle uni-gramme, de plus les différentes versions de notre modèle affichent des améliorations par rapport au modèle considérant uniquement l'importance d'une page dans l'évaluation de sa probabilité a priori. Nous avons constaté également que la troisième version de notre modèle affiche les meilleurs résultats en permettant à une page dont le contenu informationnel est similaire à celui de son site, de bénéficier plus de l'importance du site, que les autres pages.

# CONCLUSION GENERALE

Les travaux présentés dans ce mémoire rentrent dans le cadre de la recherche d'information. Les techniques traditionnelles de la RI représentent le contenu textuel d'un document (requête) par un ensemble de mots-clés ; appelé aussi représentation en sac de mots. Cette représentation est à l'origine des problèmes d'ambigüité et de disparité entre termes. Une représentation des documents et des requêtes, allant au-delà des mots simples est nécessaire pour pallier à ces deux problèmes.

Des évidences autres que le contenu textuel du document peuvent être exploitées, dans le contexte du web, afin d'améliorer les performances de la recherche d'information. Parmi ces évidences, on y trouve la structure des liens. Cependant, la structuration hiérarchique du web (structuration sous forme de site web) est ignorée dans les approches antérieures, qui intègrent cette évidence.

Nous nous sommes intéressé dans le cadre de ce mémoire à proposer des solutions permettant, d'une part, à mieux représenter le contenu sémantique des documents et des requêtes et d'autre part, à combiner le contenu textuel des documents avec une autre source d'évidence à savoir la structure de liens. Cette évidence nous a permis d'estimer la pertinence a priori d'un document web. Plus explicitement nous avons proposés dans ce mémoire trois approches :

1. Une approche permettant une bonne représentation du contenu sémantique des documents, dans le but de réduire le problème d'ambigüité, par l'utilisation des mots composés comme unités de représentation à côté des mots simples. Cette solution est retenue de fait que les mots composés sont moins ambigües que les mots qui les composent. L'extraction des mots composés est basée sur l'utilisation de la relation de cooccurrence entre termes. De plus, nous avons proposé un nouveau schéma de pondération pour ces mots composés, basé sur la notion de dominance entre termes.
2. Une approche permettant une bonne représentation de la requête, dans le but de remédier au problème de disparité des termes, réalisée par l'utilisation d'expansion de la requête en utilisant la technique de pseudo-réinjection de pertinence. Cette nouvelle représentation de la requête permet de mieux caractériser le sujet (thème) de la requête, par ajout de nouveau mots (simples ou composés) apparaissant dans les documents les mieux classés et qui co-occurrent avec les termes de la requête initiale. Cette approche est superposée sur le modèle proposé dans la première approche.

3. Une approche permettant d'intégrer l'importance d'un site web dans le calcul de la probabilité a priori de pertinence d'une page web, en assignant plus de confidence aux pages provenant des sites importants (intéressants) que pour celles provenant des sites moins importants. Une fois ce score apriori calculé, on le combine avec le score obtenu par le contenu textuel du document.

Nous avons formalisé nos trois propositions dans le cadre de modèle de langue. Celui-ci, est caractérisé par sa définition formelle des différentes heuristiques utilisées jusque-là par les modèles classiques de recherche d'information tel que $tf \times idf$, ses résultats très satisfaisants obtenus, et sa capacité à intégrer différentes évidences sur un document. Nous avons évalué nos propositions sur des collections TREC. Les résultats obtenus sont promoteurs.

Plusieurs pistes peuvent être investies à moyen terme pour améliorer les résultats et l'efficacité de nos contributions. Nous pouvons citer entre autres :

Au niveau de la représentation des documents, plus particulièrement à l'étape d'extraction des mots composés nous explorons les points suivants :

- L'impact de l'introduction de la non adjacence entre termes, la directionnalité et l'utilisation de différentes tailles des mots composés.
- L'amélioration de la procédure de filtrage des bi-grammes en utilisant par exemple des règles syntaxiques.
- La détection de l'importance d'un mot composé dans la requête.

En plus, nous prévoyons d'évaluer l'impact de l'utilisation de notre méthode de pondération des mots composés dans d'autres modèles de RI.

Au niveau représentation de la requête nous prévoyons d'explorer différents points :

- En premier lieu, l'utilisation d'autres types de relations autre que la relation de cooccurrence, pour sélectionner les termes d'expansion, telle que l'information mutuelle.
- Deuxièmement, nous examinerons l'efficacité de notre méthode d'expansion de la requête en utilisant la collection comme source de données pour l'expansion de la requête.

- Enfin, nous évaluerons notre méthode sur d'autres collections TREC plus volumineuses.

Au niveau de notre méthode d'exploitation des liens, nous prévoyons d'examiner les points suivants :

- L'évaluation de notre approche sur d'autres collections web.
- L'usage d'autres facteurs de popularité tel que le *PageRank*.
- Enfin, évaluer l'impact de propager la popularité des pages web vers leurs sites. En d'autres termes, une page populaire renforcera la popularité du site qui la contient.

A long terme, nous envisageons de combiner le modèle permettant de capturer la sémantique des documents et de la requête (par l'utilisation des mots composés et l'expansion de la requête, respectivement) avec le modèle exploitant les liens. Nous envisageons aussi, d'explorer l'apport d'autres évidences sur un document, telles que la structure de la page, le texte d'ancre, et des évidences sur la requête, telles que les requêtes passées (queries logs).

# Références bibliographiques


1. Adamson, G., Boreham, J. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Journal of Information Storage and Retrieval* vol. 10, no. 7-8, pp. 253-260, 1974.
2. Adar, E., Teevan, J., Dumais, S.T. & Elsas, J.L. The Web changes everything: Understanding the dynamics of web content. *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pp. 282–291, 2009.
3. Agirre, E., Di Nunzio, G.M.,Mandl, T., and Otegi, A. Clef 2009 ad hoc track overview: Robust–wsd task. *Proceedings of CLEF*. Springer, 2009.
4. Agirre, E., Martinez, D. Knowledge Sources for Word Sense Disambiguation.*Proceedings of the 4th International Conference on Text, Speech and Dialogue*, pp. 1-10, 2001.
5. Agosti, M., Crivellari, F., Melucci, M. The effectiveness of meta-data and other content descriptive data in web information retrieval. *Proceedings of the Third IEEE Meta-Data Conference*, Bethesda MD, pp. 139-149, 1999.
6. Alvarez, C., Langlais, P. Nie, J.Y. Word Pairs in Language Modelling for Information Retrieval. *Rapport interne, RALI*, 2003.
7. Arguello, J., Elsas, J.L., Callan, J., Carbonell, J. G. Document representation and query expansion models for blog recommendation. *Proceedings of the 2nd International Conference on Weblogs and Social Media*. AAAI Press, pp. 10–18, 2008.
8. Aslam, J. A., Montague, M. Models for metasearch. *Proceedings of the 24th annual int. ACM SIGIR conf. on Research and development in information retrieval*, pp. 276–284, 2001.
9. Aussenac-Gilles N., Biébow B., Szulman N., Revisiting Ontology Design: a method based on corpus analysis. *Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management*. pp. 172-188. 2000.
10. Baeza-Yates, R., Ribeiro-Neto, B. A. Modern Information Retrieval. *Pearson Education Ltd.*, Harlow, UK, 2nd edn, 2011.
11. Bai, J., Nie, J.-Y., Cao, G., Bouchard, H. Using query contexts in information retrieval. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, pp. 15–22, 2007.
12. Bai, J., Song, D., Bruza, P., Nie, J., and Cao, G., Query Expansion Using Term Relationships in Language Models for Information Retrieval. *Proceedings of ACM 14th Conference on Information and Knowledge Management*, pp. 688-695, 2005.
13. Banerjee, S., Pedersen, T. The Design, Implementation, and Use of the Ngram Statistic Package. *Proceedings of ICITPCL*, pp. 370-381, 2003.
14. Bartell, B.T., Cottrell, G.W., Belew, R.K. Automatic combination of multiple ranked retrieval systems. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 173 – 181, 1994.
15. Baziz M., Indexation Conceptuelle Guidée Par Ontologie Pour La Recherche d'Information. *Thèse de Doctorat en Informatique de l'Université Paul Sabatier de Toulouse*, 2005.
16. Baziz, M., Aussenac-Gilles, N., Boughanem, M. Désambiguisation et Expansion de Requêtes dans un SRI, Etude de l'apport des liens sémantiques. Dans *Revue des Sciences et Technologies de l'Information (RSTI) série ISI, Hermes, 11, rue Lavoisier*, F-75008 Paris, V. 8, N. 4, pp. 113-136, 2003.
17. Baziz, M., Boughanem, M., Aussenac-Gilles, N. Chrisment, C. Semantic Cores for Representing Documents in IR. *Dans 20th ACM Symposium on Applied Computing*, Santa Fe, New Mexico, USA, ACM Press, New York, NY, USA, pp. 1011 - 1017, 2005.
18. Berger, A. and Lafferty, J. Information retrieval as statistical translation. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222-229, 1999.
19. Berry, M.W., Dumais, S.T., O'Brien, G. W. 1995. Using linear algebra for intelligent information retrieval. *SIAM Rev.* 37(4 ), pp. 573-595, 1995.
20. Bharat, K., Broder, A., Dean, J., Henzinger, M.R. A comparison of techniques to find mirrored hosts on the www. *IEEE Data Engineering Bulletin*, 23(4), pp. 21–26, 2000.
21. Borlund, P. Measures of relative relevance and ranked halflife: performance indicators for interactive ir. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 24–28, 1998.
22. Boubekeur, F. Contribution à la définition de modèles flexibles de recherche d'information basés sur les CP-Nets. *Thèse de doctorat, Université Paul Sabatier*, 2008.

23. Boughanem, M. Outils de validation en recherche d'information. La campagne d'évaluation TREC, 2003. *http://inforsid2003.loria.fr/resumeConfRI.pdf*, 2003.
24. Boughanem, M. Les Systèmes de Recherche d'Information : d'un modèle classique à un modèle connexionniste. *Thèse de Doctorat de l'Université Paul Sabatie*r, 1992.
25. Boughanem, M., Chrisment, C., Soule-Dupuy, C. Query modification based on relevance back-propagation in adhoc environment. *Information Processing and Management*, 35, pp. 121–139, 1999.
26. Boughanem, M., Kraaij, W., Nie, J.-Y. Modèles de langue pour la recherche d'information. Les systèmes de recherche d'informations, pages 163–182. *Hermes-Lavoisier*, 2004.
27. Bourigault D. Lexter, a Natural Language Processing Tool for Terminology Extraction ". *Proceedings of Euralex'96*, Göteborg University, Department of Swedish, pp. 771-779, 1996.
28. Brin, S., Page, L. The anatomy of a large-scale hypertextual web search engine. *Proceedings of WWW7*, Brisbane, Australia, pp. 107-117, May 1998.
29. Brown, P., Della Pietra, S., Della Pietra, V., and Mercer, R. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2), pp. 263-311, 1993.
30. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G. Learning to rank using gradient descent. *Proceedings of the 22nd international conference on Machine learning*, ACM. New York, NY, USA. pp. 89–96, 2005.
31. Cao Z., Qin T., Liu T.Y., Tsai M.F., Li H., « Learning to rank : From pairwise approach to listwise approach ». *Proceedings of the 24th International Conference on Machine Learning*, pp. 129-136, 2007.
32. Cao, G., Gao, J., Nie, J.-Y., Robertson, S.. Selecting good expansion terms for pseudorelevance feedback. *Proceedings of the 31st Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*. ACM Press, pp. 243–250, 2008.
33. Cao, G., Nie, J.Y., Bai, J. Integrating word relationships into language models. *Proceedings of ACMSIGIR Conference on Research and Development in Information Retrieval*, 298–305, 2005.
34. Carmel, D., Amitay, E., Herscovici, M., Maarek, Y., Petruschka, Y., and Soffer, A. Juru at TREC t 0: Experiments with Index Pruning. *Proceedings of the Text Retrieval Conference*, E. M. Yoorhees md D. K. Harman (eds.), NIST Special Publication 500-250, Gaithersburg, MD, November 13-16,2000.
35. Carpineto, C., De Mori, R., Romano, G., Bigi, B. An information theoretic approach to automatic query expansion. *ACM Trans. Info. Syst. 19*, 1, pp. 1–27, 2001.
36. Carpineto, C., Romano, G., «A Survey of Automatic Query Expansion in Information Retrieval». *ACM Computing Surveys*, Vol. 44, No. 1. 2012.
37. Chakrabarti, S., Dom, B. Indyk, P. Enhanced hypertext categorization using hyperlinks. In L. M. Haas and A. Tiwary, editors. *Proceedings ACM SIGMOD International Conference on Management of Data*, pp. 307–318, 1998.
38. Chakrabarti, S., Dom, B., Raghavan, P., Rajagopalan, S., Gibson, D., Kleinberg, J. Automatic resource list compilation by analyzing hyperlink structure and associated text. *Proceedings of the 7th International World Wide Web Conference*, pp. 65-74, 1998.
39. Cho, J., Garcia-Molina, H., Page, L. The anatomy of efficient crawling through url ordering. *Computer Networks and ISDN Systems*, 30(1–7), pp. 161–172, 1998.
40. Church, K. and Hanks, P. Word association norms, mutual information and lexicography. *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pp. 76-83. 1990.
41. Clements, M., de Vries, A. P., and Reinders, M. J. T. The influence of personalization on tag query length in social media search. *Information Processing and Management*, 46(4), pp. 403–412, 2010.
42. Cleverdon, C. Progress in documentation. Evaluation of information retrieval systems. *Journal of Documentation*, 26, pp. 55–67, 1970.
43. Clough, P., Müller, H., Deselaers, T., Grubinger, M., Lehmann, T., Jensen, J., and Hersh, W. The *CLEF 2005 Cross-Language Image Retrieval Track*, 2005.
44. Collins-Thompson, K. Ogilvie, P. Zhang, Y. Callan, J. Information Filtering, Novelty Detection, and Named-Page Finding. *TREC-11 Notebook Proceedings*,2002.
45. Collins-Thompson, K., Callan, J. Query expansion using random walk models. In *Proceedings of the 14th Conference on Information and Knowledge Management*. ACM Press, pp. 704–711, 2005.
46. Cossock, D., Zhang, T. Subset ranking using regression. *Proceedings of the 19th Conference on Learning Theory*, pp. 605-619, 2006.
47. Craswell, N., Hawking, D., Robertson, S. Effective Site Finding using Link Anchor Information, *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 250-257, 2001.
48. Crestani, F., Lee, P.L. Searching the web by constrained spreading activation. *Information Processing and Management*, vol. 36, pp.585–605, 2000.
49. Croft W.B. Editorial. *ACM Trans. Inf. Syst.* 9(3): 185, 1991.

50. Croft, W. B., Turtle, H. R. and Lewis, D. D. The Use of Phrases and Structured Queries in Information Retrieval. *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 32-45, 1991.
51. Croft, W.B. Combining approaches to information retrieval. In Croft, W. B. (Ed.), Advances in Information Retrieval: Recent Research from the Centre for Intelligent Information Retrieval, *Kluwer Academic Publishers*, pp.1-36, 2002.
52. Croft, W.B., Harper, D. J. Using Probabilistic Models of Document Retrieval without Relevance Information. *Journal of Documentation*, 35(4) :pp .285-295, 1979.
53. Cronen-Townsend, S. Croft, W. B. Quantifying query ambiguity. *Proceedings of the 2nd International Conference on Human Language Technology Research*. ACM Press, pp. 104–109, 2002.
54. Cutler, M. Shih, Y. Meng, W. Using the Structure of HTML Documents to Improve Retrieval. *Proceedings of the USENIX Symposium on Internet Technologies and Systems Monterey*, pp. 22-22 1997.
55. Cutts, M. Spotlight keynote. *In Proceedings of Search Engine Strategies*, 2012.
56. Daille, B. Study and implementation of combined techniques for automatic extraction of terminology. The balancing act combining symbolic and statistical approaches to language, *MIT Press*, Cambridge, Massachusetts, pp. 49-66, 1996.
57. Dang, V. and Croft, B. W. Query reformulation using anchor text. *Proceedings of the third ACM international conference on Web search and data mining*, pp. 41–50, 2010.
58. Das, A. & Jain, A. Indexing the World Wide Web: The journey so far. *IGI Global*, chap. 1, 1–28, 2012.
59. David A.E , Zhai, C. Noun-phrase analysis in unrestricted text for information retrieval. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pp. 17-24, 1996.
60. Dean, J., Henzinger, M. R. Finding related pages in the world wide web. *Computer Networks*, 31(11-16): pp. 1467–1479, 1999.
61. Dias, G., Guillore, S., Bassano, J.C, and Lopes, J.G.P. Extraction automatique d'unités complexes: Un enjeu fondamental pour la recherche documentaire. *Traitement Automatique des Langues*, 41(2), pp. 447-472, 2000.
62. Diaz, F., Jones, R. Using temporal profiles of queries for precision prediction. *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pp. 18–24, 2004.
63. Dominich, S. Mathematical Foundations of Information Retrieval. *Kluwer Academic Publishers*, Dordrecht, Boston, London, 2001.
64. Dumais, S. Latent Semantic Indexing (LSI). *Proceeding of TREC-3*, 1994.
65. Eiron, N., McCurley, K.S. Analysis of Anchor Text for Web Search. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 459-460, 2003.
66. Fagan, J. L. Experiments in Automatic Phrase Indexing For Document Retrieval:A Comparison of Syntactic and Non-Syntactic Methods. *Ph.D. thesis, Department of Computer Science, Cornell University*, Ithaca, NY, 1987.
67. Fagan, J.L. The Effectiveness of Non syntactic Approach to Automatic Phrase Indexing for Document Retrieval. *Journal of the american Sociely for Information Science* 40:2, pp.115-132, 1989.
68. Foltz, P. W. Using Latent Semantic Indexing for information filtering. CACM, pp. 40-47, 1990.
69. Fox, C. Lexical analysis and stoplists, Frakes W B, Baeza-Yates R (eds) *Prentice Hall*, New jersey, pp. 102–130. 1992.
70. Fox, E.A., Nunn, G., Lee, W. Coefficients for combining concept classes in a collection. *Proceedings of the 11th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 291–308. 1988.
71. Fox, E.A., Shaw, J.A. Combination of multiple searches. In Harman, D.K. (Ed.), *Proceedings of the 2nd Text Retrieval Conference (TREC-2), NIST Special Publication* 500-215, pp. 243-249, 1994.
72. Fuhr, N. Information Retrieval - From Information Access to Contextual Retrieval. In M. Eibl, C. Wolf, and C. Womser-Hacker, editors, *Designing Information Systems*. Festschrift für Jürgen Krause, pp. 47-57. UVK Verlagsgesellschaft, 2005.
73. Furnas, G.W., Landauer, T.K., Gomez, L.M., Dumais S.T. The Vocabulary Problem in Human-System Communication, *Communications of the ACM* 30, PP. 964-971, 1987.
74. Gao, J. F., Nie, J. Y., Wu, G., Cao, G. Dependence Language Model for Information Retrieval. *Proceedings of the 27th ACM SIGIR Conference on Research and Development in IR*, pp.170-177, 2004.
75. Glover, E.J., Tsiouliklis, K., Lawrence, S., Pennock, D.M., Flake, G.W. Using Web Structure for Classifying and Describing Web Pages. *Proceedings of the 11th international conference on World Wide Web*, pp. 562-569, 2002.
76. Gonzalo, J. Verdejo, F. Chugur, I. Cigarrán, J. Indexing with WordNet synsets can improve text retrieval. *Proceedings the COLING/ACL Workshop on Usage of WordNet for Natural Language Processing*, 1998.

77. Graupmann, J., Cai, J., Schenkel, R. Automatic query refinement using mined semantic relations. In *Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration (WIRI)*. IEEE Computer Society, pp. 205–213, 2005.
78. Hammache, A., Boughanem, M., Ahmed-Ouamer, R. A new language model combining single and compound terms. *IEEE ACM Web Intelligence Conference*. pp. 67 - 70, 2011
79. Hammache,A., Boughanem,M., Ahmed-Ouamer,R. Prise en compte de l'importance d'un site web dans l'estimation de la probabilité a priori de pertinence d'une page web. *31ème édition d'INFORSID*, 2013.
80. Hammache,A., Boughanem,M., Ahmed-Ouamer,R. Pseudo-réinjection de pertinence basée sur un modèle de langue mixte combinant les termes simples et composes. *Dixième édition de la Conférence en Recherche d'Information et Applications*. CORIA2013.
81. Hammache,A., Boughanem,M., Ahmed-Ouamer,R. Combining compound and single terms under language model framework, *Knowledge and Information Systems An International* Journal. ISSN 0219-1377, DOI 10.1007/s10115-013-0618-x.
82. Hammache,A., Boughanem,M., Ahmed-Ouamer,R. Importance du Site dans le Calcul de la Probabilité A Priori de Pertinence d'une Page Web. *Colloque International sur l'Optimisation et les Systèmes d'Information*, Annaba, Algérie 25-27 mai, pp. 417-427, 2009.
83. Hammache,A., Boughanem,M., Ahmed-Ouamer,R. Introduction de la Sémantique d'un Document sous le Modèle de Langage. *Conférence en Recherche d'Information et Applications*, Presqu'île de Giens, France, pp. 433-444, 2009.
84. Hammache,A., Boughanem,M., Ahmed-Ouamer,R. Introduction de la Sémantique d'une Page Web sous le Modèle de Langage. *Journées Scientifiques sur l'Informatique et ses Applications*, Guelma, Algérie 03-04 mars, pp. 160-166, 2009.
85. Hammache,A., Boughanem,M., Ahmed-Ouamer,R. Introduction of the website importance into the evaluation of the prior probability of relevance of web page. *International Conference on Machine and Web Intelligence*. Digital Object Identifier: 10.1109/ICMWI.2010.5647831 , pp. 77 – 81, 2010.
86. Hammache,A., Boughanem,M., Ahmed-Ouamer,R. Recherche d'Information sur le Web : Introduction de l'Importance d'un Site dans le Calcul de la Probabilité A Priori de Pertinence d'une Page Web sous le Modèle de Langage. *Conférence Internationale des Technologies de l''Information et de la Communication*, Sétif, Algérie, 2009.
87. Hammache,A., Boughanem,M., Ahmed-Ouamer,R. Un modèle de langage mixte combinant les termes composés et les termes simples. *Rencontres sur la Recherche en Informatique*, Tizi-Ouzou 12,13 et 14 Juin 2011.
88. Harman, D. Relevance feedback and other query modification techniques. In Information Retrieval : Data Structures and Algorithms, William B. Frakes and Ricardo Baeza-Yates, editors, *Prentice Hall*, Englewood, Cliffs, NJ, pp. 241–263, 1992.
89. Harter, S. Psychological relevance and information science. *Journal of the American Society for Information Science* (JASIS), 43:602–615, 1992.
90. Hauff, C., Azzopardi, L. Age dependent document priors in link structure analysis. *European Conference in Information Retrieval*, pp 552–554, 2005.
91. Haveliwala, T.H. Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering*, Volume 15(4), pp. 784–796, 2003.
92. Hawking, D., Craswal, N. Overview of the TREC-2001 web track". *Proceeding of TREC-10, NIST publication* #500-250, Gaithersburg, pp. 61-67, 2002.
93. Hawking, D., Craswal, N. Overview of the TREC-2002 web track". *Proceeding of TREC-11, NIST* publication #500-251, Gaithersburg, pp. 86-95, 2003.
94. Hiemstra, D. A linguistically motivated probabilistic model of information retrieval. In Nicolaou, C., and Stephanides, C., editors, *Research and Advanced Technology for Digital Libraries - Second European Conference, ECDL'98, Proceedings*, number 1513 in Lecture Notes in Computer Science. Springer Verlag, pp. 569-584, 1998.
95. Hu, J., Deng, W., Guo, J. Improving retrieval performance by global analysis. *Proceedings of the 18th International Conference on Pattern Recognition*. IEEE Computer Society, pp. 703–706, 2005.
96. Huang, X. and Robertson, S.E. "Comparisons of Probabilistic Compound Unit Weighting Methods. *Proceedings of the 2001 IEEE ICDM Workshop on Text Mining*, San Jose, USA, pp. 1-15, 2001.
97. Jacquemin, C. Spotting and Discovering Terms through Natural Language Processing. *MIT Press*, Cambridge, Mass, 2001.
98. Jacquemin, C., Daille, B., Royanté, J., and Polanco, X. In vitro evaluation of a program for machine-aided indexing. *Inf. Process. Manage*. 38, 6, pp. 765-792. 2002.
99. Jansen, B. J., Spink, A. and Pedersen, J. A temporal comparison of AltaVista web searching. *Journal of the American Society for Information Science and Technology* 56(6), PP. 559–570, 2005.
100. Jelinek, F. Statistical Methods for Speech Recognition. *MIT Press*, Cambridge, MA, 1998.

101. Jiang, M., Jensen, E., Beitzel, S. Effective Use of Phrases in Language Modeling to Improve Information Retrieval, 2004.
102. Jones, S., Paynter, G.W. Human evaluation of Kea, an automatic key phrasing system. *JCDL*: pp.148-156, 2001.
103. Kamps, J., Mishne, G., de Rijke, M. Language Models for Searching in Web Corpora, 2005.
104. Khoo, C., Myaeng, S., and Oddy, R. Using Cause-Effect Relations in Text to Improve Information Retrieval Precision. *Information Processing and Management* 37, pp. 119-145, 2001.
105. Kleinberg, J.M. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM* 46, 5, pp. 604–632, 1999.
106. Kraaij, W., Westerveld, D., Hiemstra, D. The Importance of Prior Probabilities for Entry Page Search. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 27–34, 2002.
107. Kraft, D. H. and BueIl, D. A. Fuzzy sets and generalized Boolean retrieval systems. *International Journal on Man-Machine Studies*, 19: pp. 49-56, 1983.
108. Krovetz, R. Croft, W.B..Lexical Ambiguity and Information Retrieval. *ACM Transactions on Information Systems*, 10(1). 1992.
109. Krovetz, R. Homonymy and polysemy in information retrieval. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 72-79, 1997.
110. Kwok, K.L. A neural network for probabilistic information retrieval. *International ACM SIGIR Conference on Research and Developpement in Information Retrieval*, pp 21-30, 1989.
111. Lafferty, J., Zhai, C. Document language models, query models, and risk minimization for information retrieval. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.). *Proceedings of the 24th annual international ACM-SIGIR conference on research and development in information retrieval*, pp.111-119, 2001.
112. Lancaster, F. Information Retrieval Systems: characteristics, testing, and evaluation, John Wiley, New York, 1979.
113. Lang, H., Wang, B., Metzler, D., Li, J-T. Improved Latent Concept Expansion Using Hierarchical Markov Random Fields. *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 249-258, 2010.
114. Latifur, R. K. Ontology-based Information Selection. *PhD Thesis, Faculty of the Graduate School, University of Southern California*. August 2000.
115. Lauer, M. Corpus statistics meet the noun compound: some empirical results. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 47-54, 1995.
116. Lavrenko, V., & Croft, W. B. Relevance-based language models. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.). *Proceedings of the 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, Louisiana, pp.120-127, 2001.
117. Lechani, L., Boughanem, M. Accès personnalisé à l'information : Approches et techniques. *Rapport interne, IRIT*, 2005.
118. Lee, J.H. Analyse of multiple evidence combination. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 267-176, 1997.
119. Lee, K. S., Croft, W. B., Allan, J. A cluster-based resampling method for pseudo-relevance feedback. *Proceedings of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, pp. 235–242, 2008.
120. Lempel, R., Moran, S. The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect. *Proceedings of the 9th international World Wide Web conference on Computer networks: the international journal of computer and telecommunications networking*, pp. 387-401, 2000.
121. Li, X., Croft, W.B. Time-based language models. *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 469–475, 2003.
122. Liu, S., Liu, F., Yu, C., and Meng, W. An effective approach to document retrieval via utilizing WordNet and recognizing phrases. *Proceedings of the 27th Annual international Conference on Research and Development in information Retrieval*. ACM Press, New York, NY, pp. 266-272, 2004.
123. Liu, T.Y., Learning to Rank for Information Retrieval, *Springer*, 2011.
124. Liu, T.-Y., Xu, J., Qin, T., Xiong, W., Li, H. Letor: Benchmark dataset for research on learning to rank for information retrieval. *LR4IR, in conjunction with SIGIR*, 2007.
125. Liu, X. and Croft, W. B. Cluster-Based Retrieval Using Language Models. *Proceedings of the 27th ACM SIGIR Conference on Research & Development on Information Retrieval*, 186-193, 2004.
126. Luhn, H. P. A Business Intelligence System. *IBM Journal Research and Development* (2:4), pp. 314-319, 1958.

127. Lv, Y., Zhai. C. Positional Relevance Model for Pseudo-Relevance Feedback. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 579-586, 2010.
128. Lv, Y., Zhai. C. A comparative study of methods for estimating query language models with pseudo feedback. *Proceedings of the 18th ACM conference on Information and knowledge manag*ement, pp. 1895-1898, 2009.
129. Lv, Y., Zhai. C. Positional language models for information retrieval. *Proceedings of the 32th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 299-306, 2009.
130. Maarek. Y., Berry, D., and Kaiser, G. An Information Retrieval Approach for Automatically Constructing Software Libraries. *IEEE Transactions on Software Engineering* 17: 8. pp. 800-813, 1991.
131. Macdonald, C., and He, B. Researching and Building IR applications using Terrier; *European Conference on Information Retrieval*, 2008.
132. Manning, D., Raghavan, P. And Schute, H. Introduction to Information Retrieval. *Cambridge University Press*, 2008.
133. Manning, D., Schütze, H. Foundations of Statistical Natural Language Processing. *MIT Press*, 2000.
134. Martin, W. J. R., AI, B. P. F., and van Strenkenburg, P. J. G. On the Processing of Test Corpus: Froni Textual Data to Lexicographical Information. In Lexicography. *Principles and Practice, R. R. K. I-Tartmann (cd.), Academic Press*, London, 1983, pp.77-87.
135. Mcinnes, B.T. Extending the log-likelihood measure to improve collocation identification. *Master thesis, University of Minnesota*, 2004.
136. Metzler, D., Croft, W. B. Latent concept expansion using markov random fields. *Proceedings of the international ACM SIGIR conference on Research and development in information retrieval*, pp. 311–318, 2007.
137. Metzler, D., Croft, W.B. A Markov random field model for term dependencies, in: R.A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, J. Tait (Eds.). *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 472–479, 2005.
138. Miao, J., Huang, J.X.,Ye., Z. Proximity-based Rocchio's Model for Pseudo Relevance Feedback. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 535-544, 2012.
139. Miller, D.R.H., Leek, T., Schwartz, R.M. A hidden markov model information retrieval system. *Hearst et al*., pp. 214–221, 1999.
140. Mitra M., Bucklcy C., Singhal A., and Cardie C., An Analysis of Statistical and Syntactic Phrases. *Proceedings of the Fifth Conference on Computer Assisted Information Retrieval*, Montreal, Canada, June 25-27pp. 200-2 14, 1997.
141. Mittendort E., Mateev, B., and Schauble, P. Using the Co-occurrence of Words for Retrieval Weighting. *Information Retrieval* 3:3, pp. 243-251, 2000.
142. Mizzaro, S. Relevance, the whole (hi) story. *Journal of the American Society for Information Science*, 48, pp. 810–832, 1997.
143. Na, S-H., Kim, J., Kang, I-S., Lee, J-H. Exploiting Proximity Feature in Bigram Language Model for Information Retrieval. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 821-822 2008.
144. Nallapati, R. Discriminative models for information retrieval ». *Proceedings on the 27th annual international SIGIR Conference on Research and Development in Information Retrieval*, pp. 64-71, 2004.
145. Navigli, R. Word sense disambiguation: A survey. *ACM Comput. Surv*. 41, 2, PP. 1–69, 2009.
146. Ogilvie, P., Callan, J. Combining document representations for known-item search. Proceedings of *ACM SIGIR conference on Research and development in information retrieval*, pp. 143–150, 2003.
147. Ogilvie, P., Callan, J. Combining structural information and the use of priors in mixed named-page and homepage finding. *TREC-12 Notebook Proceedings* (Gaithersburg, MD, USA, November 2003), NIST.
148. Oliver, A., McBryan. Genvl and WWWW: Tools for taming the Web. *Proceedings of the First International Conference on the World Wide Web*, Geneva, Switzerland, 1994.
149. Parapar, J., E.Losada, D., Barreiro, A. Compression-Based Document Length Prior for Language Model. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 652-653, 2009.
150. Petrovic S, Snajder J, Dalbelo-Basic B, Kolar M. Comparison of collocation extraction measures for document indexing. *Jornal of Computing and Information Technolgie* 14: pp. 321–327, 2006.
151. Ponte, J.M., Croft, W. B. A 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.

152. Porter, M. An algorithm for suffix stripping. *Program*, Vol. 14(3), pp. 130-137, 1980.
153. Qiu, Y., Frei, H.P. Concept Based Query Expansion. *Proceedings of the 16th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.160-169, 1993.
154. Radecki, T. Fuzzy set theoretical approach to document retrieval. *Information Processing and Management*, 15: pp. 247-259, 1979.
155. Ren, F., Fan, L., Nie, J-Y. SAAK Approach: How to Acquire Knowledge in an Actual Application System. *International Conference on Artificial Intelligence and Soft Computing*, Honolulu, pp.136-140, 1999.
156. Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V., Liu, Y. Statistical machine translation for query expansion in answer retrieval. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 464–471, 2007.
157. Rijsbergen C. J.V. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33: 106-119, 1977.
158. Robertson, S.E. , S. Walker. On relevance weights with little relevance information. *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 16–24, 1997.
159. Robertson, S. E. On term selection for query expansion. *Journal of. Documentation* 46(4), pp. 359–364, 1990.
160. Robertson, S. E. The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4), pp. 294-304, 1977.
161. Robertson, S.E., Sparck Jones, K. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science* 27, pp. 129-146, 1976.
162. Robertson, S.E., Walker. S., Jones, S., Hancock-Beaulieu, M. and Gatford, M. Okapi at trec-3. *TREC*, pp. 109-126, 1994.
163. Rocchio, J.J. Relevance Feedback in Information Retrieval, in The Smart System Experiments in Automatic Document Processing in Automatic Document Processing. *Editor Prentice-Gall*. pp. 313-323, 1971.
164. Sabah, G., et Grau, B. Compréhension automatique de textes, chap. 13, pp. 293-307, *Ingéniérie des langues*, sous la direction de J.M.Pierrel, Hermes, 2000.
165. Salton, G. A comparison between manual and automatic indexing methods. *Journal of American Documentation*, 20(1), pp. 61–71, 1971.
166. Salton, G. The smart Retrieval System: Experiments in Automatic Document Processing. *Prentice-Hall*, 1971.
167. Salton, G., E.A. Fox, H. Wu. Extended Boolean information retrieval system. *CACM* 26(11), pp. 1022-1036, 1983.
168. Salton, G., McGill, M. Introduction to Modern Information Retrieval. *McGraw-Hill Int. Book Co*, 1984.
169. Salton., G, Buckley., C. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science* 41, pp. 288-297, 1990.
170. Sanderson, M. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval* 4, pp. 247–375, 2010.
171. Sanderson, M. Word sense disambiguation and information retrieval. *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp. 142-151, Springer- Verlag, 1994.
172. Saracevic, T. Relevance reconsidered. *Conceptions of Library and Information Science*, pp. 201–218, 1996.
173. Sauvagnat, K. Modèle flexible pour la recherche d'information dans des corpus de documents semi-structurés. *Thèse de doctorat, Université Paul Sabatier, Toulouse, France*, 2005.
174. Savoy, J., Picard, J. Retrieval effectiveness on the web. *Information Processing and Management*, vol. 37, pp. 543–569, 2001.
175. Schmid, H. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*. pp 25-36, 1998.
176. Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas and Richard A. Harshman"Indexing by Latent Semantic Analysis". *In Journal of the American Society of Information Science*, Vol. 41:6, pp. 391-407 , 1990.
177. Sheridan, P., Smeaton, A.F. The Application of Morpho-Syntactic Language Processing to Effective Phrase Matching. *Inf. Process. Manage*. 28(3): 349-370 1992.
178. Shi, L., Nie, J. Y., Integrating Phrase Inseparability in Phrase-Based Model. *Proceedings of the 32th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 708–709, 2009.
179. Si, L., Jin, R., Callan, J. P. and Ogilvie, P. A language modeling framework for resource selection and results merging. *Proceedings of the 11th of Conference on Information and Knowledge Management*, pp. 391–397, 2002.

180. Singhal, A., Salton, G., Mitra, M., Buckley, C. Document length normalization. *Information Processing and Management*, 32(5), pp. 619–633, 1996.
181. Smeaton & I. Quigley. Experiments on Using Semantic Distances between Words in Image Caption Retrieval, *Proceedings of an international ACM SIGIR conference on Research and development in information retrieval*, pp.174-180, 1996.
182. Song, F., Croft, W.B. A General Language Model for Information Retrieval. *Proceedings of international ACM SIGIR conference on Research and development in information retrieval*, pp. 279–280, 1999.
183. Sparck Jones, K., Van Rijsbergen, C.J. Progress in documentation. *Journal of Documentation*, Vol. 32, Num. 1, pp. 59-75, 1976.
184. Spink, A., Wolfram, D., Jansen, M. B. J., Saracevic, T. Searching the web: the public and their queries. *Journal of the American Society for Information Science and Technology* 52(3), 226–234, 2001.
185. Srikanth, M. & Srihari, R. Biterm language models for document retrieval. *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 425–426, Tampere, Finland, 2002.
186. Stein, A., Gulla, J. A., Müller, A., Thiel, U. Conversational interaction for semantic access to multimedia information. In Maybury, M. T., ed., Intelligent Multimedia Information Retrieval. Menlo Park, CA. *AAAI/The MIT Pr*ess, pp. 399–421, 1997.
187. Strzalkowski, T. et al. Natural language information retrieval. TREC 3 report, Harman D. (ed.), Overview of the Third Text REtrieval Confrerence (TREC3). NIST special publication, 1995.
188. Tao, T., Wang, X., Mei, Q., Zhai, C. Language Model Information Retrieval with Document Expansion. Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pp. 407–414, 2006.
189. Tao, T., Zhai, C. An exploration of proximity measures in information retrieval. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages 295–302, 2007.
190. Thanopoulos, A., Fakotakis, N. and Kokkinakis, G. Identification of Multiwords as Preprocessing for Automatic Extraction of Lexical Similarities.In 6th International Conference Text, Speech and Dialogue, pp. 98-105, 2003.
191. Tsai, M.-F., Liu, T.-Y., Qin, T., Chen, H.-H., Ma, W.-Y. Frank: a ranking method with fidelity loss. in 'SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval'. ACM. New York, NY, USA. pp. 383–390, 2007.
192. Tsikrika, T., Lalmas, M. Combining evidence for Web retrieval using the inference network model: an experimental study. *Information Processing and Management*, vol. 40, 2004.
193. Turpin, A., Moffat, A. Efficient Approximate Adaptive Coding. *Data Compression Conference*, pp. 357-366, 1997.
194. Van Rijsbergen, C. J. Information retrieval. London: Butterworth, 1979.
195. Voorhees, E. M. Using WordNet to disambiguate word senses for text retrieval. *International Conference on Research and Development in Information Retrieval*, pp. 171–180, 1993.
196. Voorhees, E.M. & Harman, D.K. TREC: Experiment and Evaluation in Information Retrieval. *Digital Libraries and Electronic Publishing, MIT Press*, 2005.
197. Voorhees, E.M. TREC: Continuing information retrieval's tradition of experimentation. *Communications of the ACM* , 50, pp. 51–54, 2007.
198. Wang, X., Zhai, C. Mining term association patterns from search logs for effective query reformulation. *Proceeding of the 17th ACM conference on Information and knowledge management*, pp. 479–488, 2008.
199. Weerkamp, W., Balog, K., and Meij, E. J. A generative language modeling approach for ranking entities. *Advances in Focused Retrieval*, 2009.
200. Wei, X., Croft, W. B. Investigating Retrieval Performance with Manually-Built Topic Models. *Conference - Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, pp. 333-349, 2007.
201. Wei, X., Croft, W. B. LDA-Based Document Models for Ad-hoc Retrieval. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*, pp. 178-185, 2006.
202. Williams, H., Zobel, J. Compressing Integers for Fast File Access. *Computer Journal* 42, pp. 193-201, 1999.
203. Witten, I., Moffat, A., and Bell, T. Managing Gigabytes: Compressing and Indexing Documents and Images, *Van Nostrand Reinhold*, New York, 1994.
204. Wong, S., Ziarko, W., Wong, P. Generalized vector space model in information retrieval. *Proceedings of the 8th ACM SIGIR Conference on Research and Development in information retrieval*, New-York, USA, pp. 18–25, 1985.
205. Woods, W.A. Conceptual Indexing: A better way to organize knowledge. *Technical Report SMLI TR-97-61, Sun Microsystems Laboratories*, Mountain View, CA, 1997.

206. Xu, J., Li, H. Adarank: a boosting algorithm for information retrieval. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, New York,NY, USA. pp. 391–398, 2007.
207. Xu, Y., Jones, G. J. F., Wang, B.. Query dependent pseudo-relevance feedback based on wikipedia. *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, pp. 59–66, 2009.
208. Yue Y., Finley T., Radlinski F., Joachims T. A support vector method for optimizing average precision. *Proceedings of the 30th annual international SIGIR Conference on Research and Development in Information Retrieval*, pp. 271-278, 2007.
209. Zhai, C., Lafferty, J. A study of smoothing methods for language models applied to ad hoc information retrieval. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.), *Proceedings of the 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp. 334-342, 2001.
210. Zhai, C., Lafferty, J. Model-based feedback in the language modeling approach to information retrieval. *Proceedings of the 10th International Conference on Information and Knowledge Management*. ACM Press, pp. 403–410, 2001.
211. Zhang, J., Dimitroff, A. The impact of metadata implementation on webpage visibility in search engine results (Part II). *Information Processing and Management*, 41(3):pp. 691–715, 2005.
212. Zhao, J., Yun, Y. A proximity language model for information retrieval". *Proceedings of the 32th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 291–298, 2009.
213. Zhu J · Xiangji H· Song, D., Rüger, S. Integrating multiple document features in language models for expert finding. *Knowledge Information System* 23, pp. 29–54, 2010.
214. Zhu, X. L., Gauch, S. Incorporating quality metrics in centralized / distributed information retrieval on the World Wide Web. *Proceedings of the 23th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, pp. 288–295. 2000.