UNIVERSITE JOSEPH FOURIER – GRENOBLE I
U.F.R. EN INFORMATIQUE ET MATHEMATIQUES APPLIQUES

THESE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER – GRENOBLE I
Discipline : Informatique

Présentée et soutenue publiquement

par

HO Bao-Quoc

le 18 novembre 2004

TITRE

Vers une indexation structurée basée sur des syntagmes nominaux

(impact sur un SRI en vietnamien et la RI multilingue)

Composition du jury :

Présidence	: M. Christian BOITET
Rapporteurs	: M. Patrick GALLINARI
	M. Mohand BOUGHANEM
Examinateur	: M. Jean-Pierre GIRAUDIN
Directeurs de thèse	: Mme. Marie-France BRUANDET
	Mme. DONG Thi Bich Thuy
	M. Jean-Pierre CHEVALLET

Thèse préparée au sein du laboratoire CLIPS-IMAG
(Communication Langagière et Interaction Personne Système)
Université Joseph Fourier – Grenoble I

Remerciements

Bien des personnes ont contribué, de près ou de loin, à ce travail, que ce soit par leurs conseils, leur contribution ou leurs encouragements. Je leur exprime ma profonde gratitude. Il me serait difficile de toutes les remercier sur cette page. J'espère qu'elles comprendront.

Je tiens à remercier :

M. Christian BOITET, Professeur à l'Université Joseph Fourier, qui m'a fait l'honneur de présider le jury, pour ses renseignements et ses critiques constructives. Cela m'a permis d'améliorer la qualité du manuscrit.

M. Patrick GALLINARI, Professeur de l'UPMC, LIP6, et M. Mohand BOUGHANEM, Professeur à l'Université Paul Sabatier de Toulouse, pour avoir accepté d'être rapporteurs et pour leurs remarques, et pour l'intérêt qu'ils ont manifesté pour ce travail.

M. Jean-Pierre GIRAUDIN, Professeur à l'Université Pierre Mendès France, pour son amiable participation à ce jury.

Mme Marie-France BRUANDET, Professeur à l'Université Joseph Fourier, et M. Jean-Pierre CHEVALLET, Maître de Conférence à l'Université Pierre Mendès-France, qui ont dirigé ce travail, pour le temps qu'ils m'ont consacré durant toutes ces années, pour leurs remarques attentives, leur gentillesse, leur patience et leur écoute pour la correction de mes fautes de français, ainsi que pour toute leur aide durant mes séjours en France.

Mme Thi Bich Thuy DONG, Professeur à l'Université Nationale du Vietnam à Ho Chi Minh Ville, co-directrice de ma thèse, pour son soutien et ses encouragements pendant toutes ces années.

Je remercie tous les membres du laboratoire CLIPS, en particulier Lizbeth, Razan et Jean qui ont partagé le même bureau que moi, pour leur disponibilité et la gentillesse avec laquelle ils m'ont beaucoup aidé dans la vie quotidienne.

Pour terminer, je tiens à remercier ma famille qui a eu confiance en moi, ma grande mère, mes parents qui ont toujours été là quand j'ai eu besoin d'eux, et qui m'ont appris à porter un regard ouvert sur le monde. Je remercie enfin ma femme, mes beaux parents et mes deux petits « Anh » qui ont fait bien des sacrifices pour moi.

Table de matière

Partie I : Introduction

Chapitre 1 Introduction générale

Dans ce chapitre, nous présentons premièrement les motivations qui nous amènent à effectuer cette thèse (section 1.1), puis l'objectif de notre travail (section 1.2) ainsi que, les problématiques dans le domaine de notre recherche (section 1.3). Nous présentons ensuite des travaux autour de notre travail dans la section 1.4. La dernière section résume nos principales contributions (1.5) et précise la structure du reste de cette thèse (1.6).

1.1 Motivations

L'informatique, la science du traitement automatisé de l'information au moyen d'ordinateurs, a pour vocation première le stockage, le traitement et l'accès aux informations. Dans la grande variété des domaines de recherche en informatique, la Recherche d'Information (RI) s'intéresse plus particulièrement à l'accès de manière efficace à la somme gigantesque des informations disponibles. De nos jours, l'information stockée numériquement prend le pas sur tout autre moyen de stockage. Cet état de fait motive donc la recherche en RI afin de fournir des systèmes de recherche d'information (SRI) qui permettent à l'utilisateur d'accéder plus efficacement non seulement en terme de temps, mais également en terme de qualité, aux informations numériques désormais disponibles en très grande quantité. Nous nous intéressons dans cette thèse uniquement aux informations disponibles numériquement en langue naturelle. Nous laissons donc volontairement de côté les informations structurées de type base de données, même si leur développement est indéniable avec l'introduction du langage commun de codage (XML), également les informations d'autres type de média (images, vidéo, son). L'information codée en langue naturelle est d'une part un vecteur de diffusion très efficace et très utilisé, et d'autre part soulève encore de gros problèmes dans le domaine de recherche de la RI, dont le problème du multilinguisme.

Actuellement, les statistiques montrent qu'environ 70% des sites web sont en langue anglaise. Cependant, les sites dans des langues non anglaises ne cessent de croître. A terme, il est inévitable (même probablement souhaitable), que toutes les langues vivantes soient représentées sur le Web, c'est à dire que les internautes trouveront des informations correspondant à leurs besoins dans leur langue maternelle. D'autre part, être capable de lire et comprendre une autre langue que sa langue maternelle ne permet pas forcément de formuler

une requête efficace. La difficulté de construire une "bonne" requête pour un Système de Recherche d'Information (SRI) augmente avec l'usage d'une langue étrangère. Il est par exemple bien difficile de trouver les termes précis ou tournures idiomatiques pour former une requête précise, même si on est capable de comprendre ces termes dans le contexte d'un document. Avec l'information majoritairement disponible uniquement en langue anglaise, les internautes ont donc vraiment besoin d'un système de recherche d'information multilingue, c'est à dire qui leur facilite l'accès aux informations du web par une aide à la construction de requête, ou par la formulation de requêtes dans leur langue maternelle, tout en étant capable de retrouver des documents dans n'importe quelle autre langue. En ce qui concerne cette thèse, nous limiterons nos ambitions aux langues anglaise, française et vietnamienne.

Le besoin de recherche d'information multilingue ne se limite pas à l'internaute privé, il provient également des exigences des organisations, et des entreprises internationales qui veulent avoir un outil de recherche d'information multilingue pour s'adapter à tous leurs employés provenant de différents pays, et ayant donc des langues différentes.

En partant des idées précédentes, un scénario optimiste peut être envisagé : un utilisateur quelconque donne une requête dans sa propre langue maternelle et le système RI multilingue va retrouver tous les documents pertinents dans des langues différentes qui existent dans une base de documents (Internet ou une base de documents d'une organisation). Ensuite, le système affiche le résultat de la recherche en traduisant tous les documents trouvés dans des langues différentes vers la langue de l'utilisateur. Bien sûr, les systèmes de traduction ne sont pas actuellement d'un niveau suffisant pour une traduction de qualité, mais leur usage ponctuel peut permettre une compréhension minimale, suffisante pour juger de leur intérêt après la recherche et demander ensuite éventuellement une traduction plus précise. Ce scénario est utile dans le cas ou l'internaute ne maîtrise pas la lecture ou le déchiffrage grossier d'une lange étrangère.

1.2 Objectif de la thèse

Dans ce travail de thèse, nous nous focalisons sur la représentation précise des informations par l'utilisation d'un contexte local. En effet, notre point de vue est qu'une recherche d'information multilingue doit franchir les barrières des langues. La principale difficulté de ce franchissement est la sélection (traduction) des équivalents terminologiques les plus corrects d'une langue à une autre. En effet, les termes des langues différentes ne recouvrent quasiment

jamais le même champ sémantique, et la dérive du sens d'une requête est inévitable dans une traduction. Pour limiter cette dérive, notre thèse propose l'étude du paradigme du terme composé comme terme d'indexation, qui par l'usage d'informations supplémentaires, réduit et donc précise le champ sémantique et doit permettre de limiter la divergence sémantique lors de la traduction. Les systèmes de SRI actuels adoptent généralement l'attitude inverse, à savoir la réduction des termes à leur plus courte représentation : le mot, le lemme, ou une racine tronquée. C'est globalement le contexte des autres mots du document qui permet de réduire une inévitable ambiguïté. Sanderson [Sanderson 94] a montré que cette approche est correcte dans un contexte monolingue. La thèse que nous soutenons est que cette approche n'est pas applicable à un contexte multilingue.

Notre travail a donc pour but de définir un modèle de ***recherche d'information bilingue*** (français, vietnamien) ***orientée précision*** et basée sur l'utilisation de ***termes d'indexation structurés***.

1.3 Problématique

Afin d'atteindre le but proposé, il faut résoudre un certain nombre de problèmes présentés ci-dessous : le choix de la nature des termes d'indexation, la structuration d'un terme d'indexation, la correspondance entre requête et document, et le passage de la barrière des langues.

1.3.1 Choix de la nature des termes d'indexation

Concernant la nature des termes d'indexation, nous envisageons de mettre en compétition plusieurs solutions possibles : mot simple, mot composé, syntagme ou phrase. Nous supposons que des termes complexes représentent mieux le contenu du document. Nous avons l'intention d'utiliser des syntagmes nominaux comme termes d'indexation. Cette idée se base sur l'hypothèse que le syntagme nominal est plus précis que le mot simple et le mot composé.

1.3.2 Structuration du terme d'indexation

Après avoir choisi le type de terme d'indexation, nous pensons qu'il est important de reconnaître, et de représenter dans le terme d'indexation, la structure du terme composé, pour bien représenter les relations entre les mots. En effet, nous faisons l'hypothèse que la connaissance de cette structure par le SRI participera à sa qualité du point de vue précision de recherche. Par exemple, une caractéristique des termes composés est leur variation

syntaxique : il existe des formulations différentes mais ayant des sens voisins comme : "logiciel de retouche d'image", "logiciels de traitement d'images", "logiciel de traitement des photos numériques", "programme de manipulation des photos numériques", "programme de retouche de photos numériques", etc. L'identification entre "logiciel de traitement", "logiciel", "programme", "programme de retouche", "programme de manipulation", permet d'augmenter la couverture (rappel) des documents retrouvés, tout en maintenant la précision et les chances de traduction correcte si le système possède des associations partielles correctes entre des parties de termes composés. La connaissance de la structure du terme permet aussi de pondérer l'importance relative des mots dans la structure du terme. Dans l'exemple précédent, il est clair que les mots "logiciel" et "programme" sont plus importants que les termes "traitement" et "manipulation" qui leur sont associés. Autrement dit, les mots "logiciel" et "programme" sont des têtes de syntagmes alors que "traitement" et "retouche" en sont des modifieurs. Le même constat peut être fait pour le terme "photo numérique".

Alors que les systèmes à indexation basés sur les mots isolés, sont incapables d'établir cette distinction, notre proposition dans cette thèse est que l'introduction explicite de cette information, hiérarchisant les mots d'un terme composé, est justement celle qui permettra d'atteindre un bon niveau de précision dans un SRI qui l'exploitera efficacement, et que la dérive sémantique liée à la traduction s'en trouvera également limitée.

1.3.3 Fonction de correspondance

Si les termes d'indexation sont des termes composés structurés, il faut alors redéfinir la fonction de correspondance entre la représentation de la requête et celle du document, afin de bien prendre en compte la structure des termes d'indexation et de bénéficier de l'augmentation de la précision d'un système de recherche d'information que nous cherchons à établir dans cette thèse.

1.3.4 Passage de la barrière des langues

Concernant l'aspect multilingue, nous devons choisir une solution pour un « franchissement » pertinent de la barrière linguistique. Le critère de qualité à retenir est celui de la qualité du SRI multilingue obtenu en final, et pas forcément la qualité intrinsèque de la traduction, car cette traduction n'a pas vocation à être utilisée par un acteur humain. Il est cependant fort probable que la richesse et la précision des ressources linguistiques participent à la qualité du SRI. Une solution possible peut être la traduction des termes composés de la requête dans la

langue cible des documents ou la définition d'un langage artificiel comme langage pivot. Cela nous conduit à poser les questions suivantes. Quelles sont les ressources nécessaires au bon fonctionnement d'un tel système : dictionnaire bilingue, corpus parallèle ou comparable ou bien la combinaison des deux, et quelle est leur implication dans le processus de RI et leur importance par rapport au reste des éléments du modèle, en particulier les aspects de pondération. Cet aspect de pondération est un point important en RI car il ne s'agit nullement de trouver la meilleure traduction, mais de satisfaire l'utilisateur par rapport à un besoin d'information souvent maladroitement exprimé. La pondération apparaît comme mesure de l'incertitude de l'expression de ce besoin dans la requête, et de l'expression du thème dans le document, et est également un moyen pour relativiser l'importance des termes des requêtes et du document par rapport au thème principal. Par exemple, les mots outils de la langue ont une importance proche de zéro car ils ne véhiculent que peu d'information.

1.4 Recherches connexes

Nous pouvons situer notre travail selon trois axes : le premier est le type des termes d'indexation, le deuxième est le nombre de langues : monolingue, bilingue ou multilingue, et le troisième est la ressource utilisée pour la « traduction », comme dans la figure suivante :

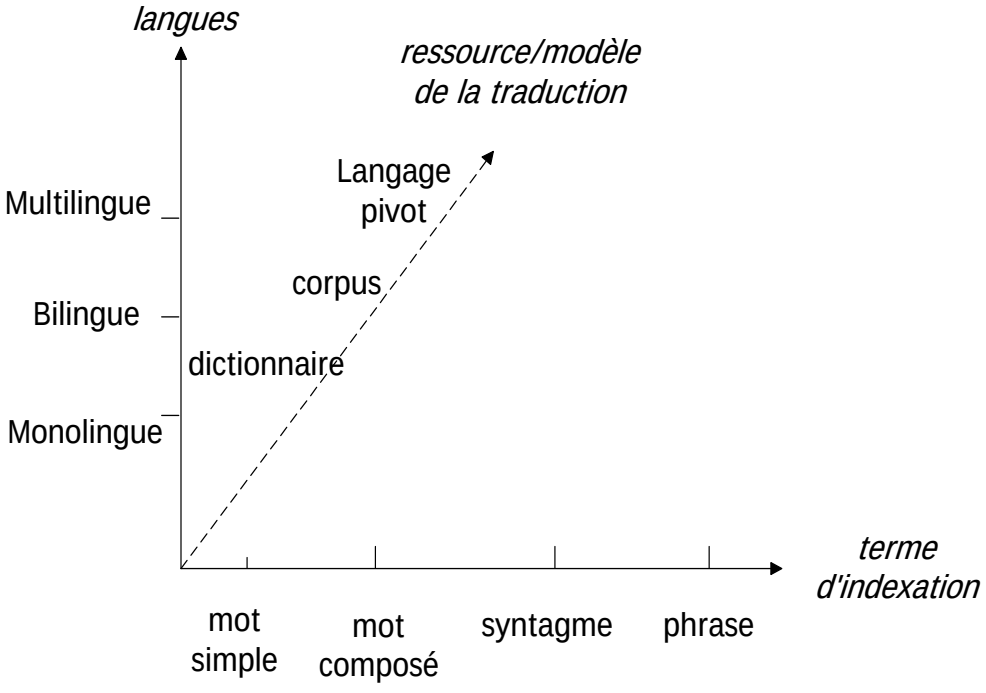


Figure 1.4-1 Axes de travail

Concernant l'axe « terme d'indexation », nous avons étudié des travaux relatifs à une indexation structurée comme le travail de Arampatzis dédié à l'indexation par les syntagmes (« *phrase indexing* » en anglais) [Arampatzis 2000], ceux de Jean Pierre Chevallet et Hadad Hatem, relatifs à un modèle d'indexation syntagmatique [Chevallet et al. 2001], et ceux de Peter Bruza [Bruza et al 93] relatifs a un modèle de pondération d'indexation par syntagmes. Concernant la fonction de correspondance, nous avons été inspirés par les travaux relatifs au modèle d'inférence proposé par Turtle et Croft [Turtle 91] et développé dans les travaux plus récents comme : le modèle d'inférence probabiliste de Wong et Yao [Wong et al. 95], le modèle de réseau de croyance (Belief Network) de Ribeiro [Riberio 96], et le modèle de Luis M. de Campos [Campos 2000].

Pour l'axe multilingue, nous nous sommes orienté vers la définition d'un modèle unifié pour la recherche d'information multilingue, comme dans le travail proposé par [Nie 2002b], et avons utilisé des travaux concernant les modèles de langage [Hiemstra 2001], [Federico et al. 2002].

1.5 Contribution de la thèse

Cette thèse propose une nouvelle technique de recherche d'information pour augmenter la performance d'un système de recherche d'information multilingue. Les contributions sont :

1. La proposition d'un modèle de recherche d'information structurée basé sur des syntagmes nominaux. Ces syntagmes sont organisés sous forme de réseaux bayésiens. L'ensemble des réseaux sert de base d'inférence probabiliste pour la fonction de correspondance. Plus précisément, nous avons proposé :

    a. des règles de décomposition d'un syntagme nominal en des sous-syntagmes. Ces règles permettent de construire un réseau de syntagmes nominaux. Ces réseaux servent de base pour la phase d'interrogation.

    b. de considérer le rôle différent que jouent le mot tête et les modifieurs d'un syntagme nominal pour pondérer les termes.

    c. un moyen d'évaluation de la pertinence entre la requête et les documents en nous basant sur le calcul des probabilités et la propagation de celles-ci le long d'un réseau bayésien.

2. L'étude des spécificités de la langue vietnamienne pour la recherche d'information et des expérimentations d'évaluation pour des types différents de termes d'indexation pour les documents en vietnamien. Ces expérimentations ont permis de vérifier les hypothèses de construction des index et de calcul de la pondération sur cette langue particulière.

3. Le développement d'une méthode de traduction de la requête, syntagme par syntagme, s'appuyant sur la structure des syntagmes.

1.6 Organisation de la thèse

Cette thèse est organisée en neuf chapitres. Après le premier chapitre d'introduction au contexte de la recherche, le chapitre 2 présente les notions de base de la recherche d'information. Une bibliographie concernant la recherche d'information multilingue est présentée dans le chapitre 3.

Dans le chapitre 4, nous nous intéressons plus particulièrement au modèle bayésien de recherche d'information, qui servira de base à notre proposition.

Les chapitres 5 et 6 sont relatifs à la description de notre modèle de RI : le chapitre 5 présente le modèle de RI en version monolingue basé sur les syntagmes nominaux, et le chapitre 6 propose une extension de ce modèle à la RI multilingue.

Une des langues qui nous intéressent plus particulièrement est le vietnamien. Le chapitre 7 est consacré à son étude, en vue de réaliser un SRI vietnamien.

Des expérimentations dédiées à la solution multilingue sont présentées dans le chapitre 8. Enfin, le chapitre 9 est consacré à la conclusion de notre travail ainsi qu'aux perspectives de recherches futures.

Chapitre 2 Système de recherche d'information

2.1 Définition d'un système de recherche d'information

Un système de recherche d'information (SRI) est un logiciel qui assiste un utilisateur dans une tâche de recherche d'information, c'est à dire dans la sélection de documents où peuvent se trouver les éléments de réponse à une question qu'il se pose. La notion de "question" est à comprendre au sens large : cela va d'une question vague et générale, comme " où en est le développement économique au Vietnam", sous-entendu "qu'y a t'il comme informations concernant le développement économique du Vietnam ?", à des question qui appellent une réponse plus précise comme "pourquoi le ciel est-il bleu ?", ou même des question directes comme "quelle est la capitale du Vietnam ?". Selon le degré de précision attendu dans la réponse, on parle de système de recherche documentaire, ou de système de question/réponse. Dans le cas où une réponse très précise et concise est souhaitée, ce qui différencie un SRI d'une base de connaissance, c'est son rôle non pas "porteur de" l'information, mais médiateur vers l'information, quelle qu'en soit la nature. [Salton et al. 83]

La problématique de la recherche d'information peut être alors être décrite comme la satisfaction d'un besoin en information d'un utilisateur, qui est exprimé par une *requête*, sur un ensemble de documents appelé *collection* ou *corpus*. La notion de document reste relativement imprécise : il s'agit simplement d'une entité physique ou électronique répertoriée comme un tout cohérent, et contenant une "information". Pour une discussion plus générale sur la notion de document, on peut se référer au travail collectif de R. Pedauque [Pédauque 04].

Un SRI se décompose classiquement en trois parties : la représentation du contenu des documents, la représentation du besoin de l'utilisateur, et la correspondances entre ces deux représentations. Ces processus sont illustrés dans la Figure 2.1-1 [Croft 93].

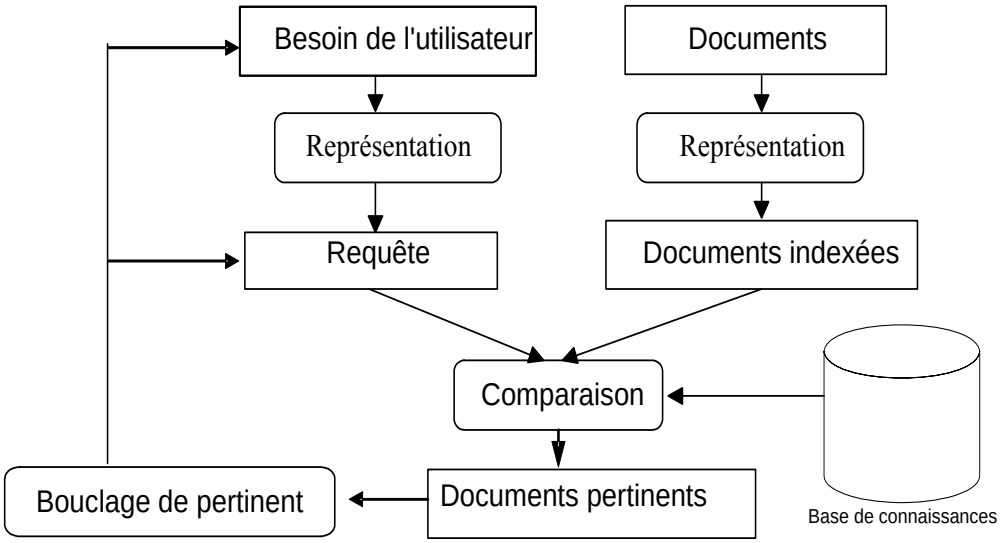


Figure 2.1-1 Système de recherche d'information

La représentation du contenu des documents est appelée phase d'indexation. Cette phase est réalisée « hors ligne », ce qui signifie que l'utilisateur d'un SRI n'est pas directement concerné : c'est un processus uniquement géré par le SRI. Elle a pour but d'extraire les termes d'indexation qui représentent au mieux le contenu sémantique des documents. Un terme d'indexation peut être un mot simple, un mot composé, un syntagme ou une expression plus complexe. La forme des termes d'indexation est précisée par *le langage d'indexation*.

Le besoin d'information d'un utilisateur est exprimé par une requête. La requête peut prendre plusieurs expressions en langue naturelle, ou être une expression booléenne de termes (comme dans les moteurs de recherche sur le Web). La requête d'un utilisateur est représentée sous une forme interne compréhensible par le système. Elle peut être étendue, soit automatiquement par le système, soit manuellement via l'interaction avec l'utilisateur. L'*expansion de requête* a pour objectif d'améliorer les performances du système quant à la qualité des documents retrouvés.

La comparaison entre une requête et un document est basée sur une *fonction de correspondance*. Les résultats sont les documents pertinents pour la requête. Ils peuvent être triés selon une *mesure de pertinence*.

2.2 Modèle de recherche d'information

On utilise la notion de *modèle de SRI* pour décrire le comportement d'un SRI. Un modèle de SRI décrit comment représenter des termes d'indexation (par la définition d'un langage d'indexation) et comment comparer une requête et un document (en utilisant la fonction de correspondance). Par exemple, dans le modèle booléen, on représente un document et une requête comme des expressions logiques d, q et la fonction de correspondance est la vérification de l'implication logique d => q. Dans le modèle vectoriel, un document et une requête sont représentés respectivement par des vecteurs de termes d'indexation $\vec{d}$ = (d_1, d_2, …, d_n), $\vec{q}$ = (q_1, q_2, …, q_n) et la fonction de correspondance est le cosinus des deux vecteurs $\vec{d}$, $\vec{q}$, $\cos(\vec{d},\vec{q}) = \frac{\sum_{i=1}^{n} d_i \times q_i}{\sqrt{\sum_{i=1}^{n} (d_i)^2} \times \sqrt{\sum_{i=1}^{n} (q_i)^2}}$

Afin de représenter l'importance relative d'un terme d'indexation dans un document ou dans une requête, on affecte à chaque terme un poids. Le processus qui calcule et affecte ce poids pour chacun des termes d'indexation, s'appelle *le processus de pondération.* C'est un processus crucial pour un SRI et il joue donc un rôle important dans la qualité de la recherche.

Les éléments caractérisant un modèle de SRI sont :

1. La nature des termes d'indexation : leur taille, leur structure ;
2. La structure de la requête ;
3. La méthode de pondération ;
4. La fonction de correspondance ;
5. Le bouclage de pertinence.

La nature des termes d'indexation. On peut utiliser des mots simples, qui sont faciles à extraire des documents, mais ne représentent pas précisément le contenu des documents. L'utilisation de mots composés ou de syntagmes comme termes d'indexation peut augmenter la précision du système, mais extraire cette sorte de termes d'indexation est plus complexe car cela demande plus de connaissances sur la langue des documents.

La structure de la requête. La requête est représentée soit comme un « sac de mots », soit comme une structure représentant les relations entre des termes.

La méthode de pondération. Il s'agit de choisir ou de proposer une méthode de pondération qui représente bien la mesure de l'importance de chaque terme.

La fonction de correspondance. Elle doit être adaptée à la nature des termes d'indexation et à leur structure pour fournir une mesure de pertinence correcte.

Le bouclage de pertinence. C'est un moyen d'expansion de la requête, qui peut être réalisé automatiquement ou avec l'interaction de l'utilisateur.

Dans notre travail, nous proposerons un modèle de SRI dont nous construisons les éléments ci-dessus pour l'adapter au mieux à la recherche d'information multilingue.

2.3 Evaluation d'un système de recherche d'information

Afin d'évaluer la performance d'un système de recherche d'information, on a besoin d'une collection test qui consiste en un ensemble de documents, un ensemble de requêtes représentant un ensemble de thèmes (topics) et une estimation de pertinence entre ces requêtes et ces documents. Certaines organisations, comme TREC (Text REtrieval Conference) ou CLEF (Cross-Language Evaluation Forum) sont spécialisées dans l'évaluation des systèmes de recherche d'information. Deux mesures principales sont utilisées : *la précision* et le *rappel*.

Soient B : l'ensemble des documents trouvés par le système

S : l'ensemble des documents pertinents pour l'utilisateur dans le corpus

$P = B \cap S$: l'ensemble des documents trouvés par le système pertinents pour l'utilisateur

$$\text{Précision} = \frac{\text{nombre de documents trouvés pertinents pour l'utilisateur (P)}}{\text{nombre des documents trouvés par le système (B)}} = \frac{|P|}{|B|}$$

$$\text{Rappel} = \frac{\text{nombre de documents trouvés pertinents pour l'utilisateur (P)}}{\text{nombre de documents pertinents pour l'utilisateur (S)}} = \frac{|P|}{|S|}$$

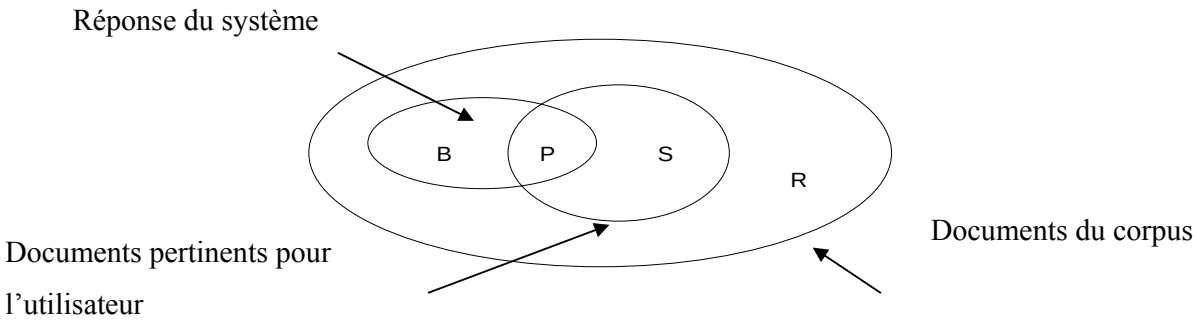


Figure 2.3-1 Evaluation d'un système recherche d'information

La précision représente la qualité des réponses, ou encore la capacité du système à ne retrouver *que* des documents pertinents. Le rappel représente la capacité du système à retrouver *tous* les documents pertinents.

2.4 Traitement automatique des langues pour la recherche d'information

En recherche d'information, afin de bien représenter le contenu d'un document ainsi que le besoin d'information de l'utilisateur, on utilise des techniques de traitement automatique de la langue naturelle (TALN). Une étude sur les techniques de traitement des langues naturelles pour la RI est présentée dans travail de Jacquemin et al. [Jacquemin 2000]. Ces techniques permettent de mieux analyser le contenu du document et de la requête, par exemple par l'extraction de termes complexes, susceptibles de mieux représenter le contenu. Dans cette section, nous aborderons les techniques du TALN utilisées en RI.

2.4.1 Techniques morphologiques

Les techniques morphologiques sont relatives à la construction des mots de la langue. La composition des mots se fait à partir de plus petites entités appelées morphèmes. Le morphème est la plus petite unité lexicale ayant un sens spécifique, c'est-à-dire que chaque morphème est indivisible tout en ayant un sens particulier. La réduction des mots en leurs morphèmes constitutifs est un exemple de traitement utile à la RI. Nous présentons ici les étapes dans l'ordre chronologique des traitements appliqués à partir du texte original.

2.4.1.1 Segmentation en unités linguistiques

En recherche d'information, la segmentation en unités linguistiques est l'étape de base de l'indexation. Les unités linguistiques peuvent être des mots simples comme « maison » ou des mots composés comme « pomme de terre ». La pertinence des unités choisies pour l'indexation influence directement la pertinence du résultat de la recherche. Pour beaucoup de

langues asiatiques, la phase de segmentation en mots est plus complexe que pour l'anglais ou le français car la plupart des unités sont composées, et il n'y a souvent pas de séparateurs de mots. En vietnamien, par exemple, les chaînes délimitées par des blancs sont des syllabes, alors que les mots sont plus souvent polysyllabiques. En japonais, en chinois, en thaï, en lao, en khmer, etc., il n'y a tout simplement pas de blanc, et parfois même pas de séparateur de phrases.

2.4.1.2 Le filtrage des mots vides

Dans un contexte de recherche d'information, tout mot, à faible contenu informatif, peut être considéré comme mot vide. Les déterminants comme le, la …sont un exemple de type de mots vides. La plupart des systèmes de recherche d'information utilisent une liste de mots vides pour éliminer les termes non significatifs dans la phase d'indexation.

2.4.1.3 Racinisation

La « racinisation » est une procédure plus ou moins linguistiquement fondée qui vise à regrouper les mots sémantiquement proches à partir de ressemblances des formes lexicales des mots. En théorie, elle est censée extraire les morphèmes d'un mot, aussi appelés morphèmes lexicaux, les bases, racines, préfixes, infixes, affixes, et produire un de ces éléments (base, racine) comme caractéristique d'une telle famille, ou encore une forme particulière (le lemme, ou le lexème), ou encore un identificateur d'une famille dérivationnelle complète. En pratique, les algorithmes utilisés simplifient souvent ce processus : on parle alors plus généralement de troncature. En recherche d'information, la racinisation des documents et des requêtes vise à améliorer le rappel par regroupement des variations morphologiques. La difficulté de l'opération provient de la complexité et de l'irrégularité plus ou moins grande de la morphologie de la langue étudiée.

En fonction de la précision de la racinisation, il existe des algorithmes de normalisation des mots qui vont d'une *lemmatisation* grammaticalement et morphologiquement correcte jusqu'à des algorithmes de *troncature* plus "radicaux" et moins fondés grammaticalement. La lemmatisation produit en général un seul lemme représentant les formes fléchies d'une même famille, comme les formes conjuguées *produira*, *produisent*…pour le verbe « produire ». Le traitement couvre à la fois la suppression des morphèmes grammaticaux flexionnels (une seule catégorie grammaticale) et dérivatifs (qui couvre plusieurs catégories grammaticales). Par exemple, les variantes de l'adjectif *lent*, le substantif *lenteur*, ou l'adverbe *lentement*. Les algorithmes utilisés ne garantissent jamais d'avoir trouvé un lemme correct, et ne distinguent

pas la morphologie dérivationnelle de la morphologie flexionnelle. Par exemple, l'algorithme de Lovins et l'algorithme de Porter sont deux algorithmes de racinisation qui donnent des résultats satisfaisants en RI pour l'anglais [Lovins 68][Porter 80].

2.4.2 Variation lexicale, synonymie

La variation lexicale est la situation où un concept est exprimé par plusieurs mots différents, par exemple : automobile, voiture, car, véhicule… Cette situation pose un problème à la RI quand l'utilisateur soumet au système une requête contenant le mot « véhicule », mais que le système ne trouve pas les documents qui contiennent le mot « voiture ». Une solution pour cette situation est la technique d'expansion de requête basée sur un thésaurus. Le thésaurus peut être une base terminologique générale (comme WordNet), une base dédiée à un domaine d'application, ou simplement des ensembles de termes jugés similaires.

2.4.3 Variation syntaxique

L'indexation par les syntagmes (« *phrase indexing* » en anglais) est une direction qui a attiré beaucoup de chercheurs en RI durant ces dernières années. Le but de l'indexation par des syntagmes est d'augmenter la précision d'un SRI en diminuant l'ambiguïté des termes d'indexation.

Il y a deux techniques utilisées pour extraire des syntagmes du contenu des documents : la technique syntaxique et la technique statistique.

La technique syntaxique procède en deux phases : la première phase effectue l'étiquetage des mots par leur catégorie syntaxique. Dans cette phase, on associe à chaque mot, en contexte, une « étiquette » syntaxique. Cette étiquette correspond à la catégorie syntaxique du mot. La deuxième est la phase d'extraction des syntagmes, le plus souvent basée sur des patrons d'étiquettes donnant les formes syntaxiques permettant de reconnaître des syntagmes.

Dans la technique statistique, on se base sur les cooccurrences des termes pour trouver des mots qui apparaissent ensemble fréquemment dans une fenêtre qui peut être un paragraphe, une partie de document, un document ou un ensemble de documents.

Une fois les syntagmes extraits, on utilise des techniques qui sont capables de regrouper les variantes d'un syntagme. Il y a plusieurs types de variation syntaxique comme :

- **La variation d'insertion** qui concerne l'ajout d'un modifieur à l'intérieur de certains groupes nominaux. Par exemple : « *lait de brebis* » et « *lait cru de brebis* »

- **La variation de coordination** qui concerne toutes les formes coordonnées de mots (adjectifs ou noms) à l'intérieur du groupe nominal. Par exemple : « *alimentation humaine* » et « *alimentation animale et humaine* »

- **La variation de permutation** qui concerne tous les mots ou les groupes de mots pouvant permuter autour d'un élément pivot (prépositions ou séquences verbales). Par exemple : « *birth day* » et « *day of birth* »

2.4.4 Variation sémantique

A l'inverse de la synonymie, le même mot peut avoir plusieurs sens dans des contextes différents (la polysémie), par exemple : le mot « table » a deux sens différents dans les deux expressions « table de logarithmes » et « table de nuit ». Cette situation diminue la précision du système. Afin de traiter cette variation, on utilise aussi la technique d'étiquetage pour associer à chaque mot une étiquette sémantique qui précise le sens du mot ou le rôle sémantique du mot comme : agent, objet, …

Dans notre approche, nous utilisons un analyseur syntaxique pour analyser les documents et la requête, et extraire des syntagmes nominaux pour représenter le contenu du document.

Partie II : Etat de l’art

Chapitre 3 Recherche d'information multilingue

Dans ce chapitre, nous présenterons un point de vue général sur la recherche d'information multilingue (SRI multilingue). Nous commençons par des notions de base d'un SRI multilingue, les problématiques d'un SRI multilingue, comment évaluer un SRI multilingue, enfin, nous décrirons des approches existantes dans ce domaine.

3.1 Définition

Un système de recherche d'information multilingue est un système de recherche d'information qui permet à un utilisateur d'exprimer une requête dans une langue et de retrouver des documents pertinents dans une autre langue.

3.2 Problématique

La recherche d'information multilingue est une intersection entre la recherche d'information et la traduction automatique. La plupart des recherches sur ce domaine examinent séparément deux tâches : la traduction et la recherche. Le problème de la traduction peut être précisé par les questions suivantes :

1. Quelle est la méthode de traduction utilisée ?

2. Comment choisir une traduction correcte parmi les traductions possibles ?

La première question amène à en poser d'autres : faut-il traduire les documents ou les requêtes ? Quelles sont les ressources utilisées pour la traduction : un logiciel de traduction automatique, un dictionnaire bilingue, ou des informations extraites d'un corpus ? Bien que la recherche d'information multilingue hérite des problèmes de la traduction automatique, ils sont plus faciles à régler en RI car on n'a pas besoin d'une traduction exacte pour chaque terme ni d'une traduction lisible et compréhensible par l'utilisateur, donc syntaxiquement correcte. En RI multilingue, la traduction a seulement une contrainte, celle de garder le thème de la requête d'origine. La deuxième question est donc beaucoup moins difficile à résoudre en RI : la notion de « correction » étant beaucoup moins exigeante, et, dans le doute, on peut éventuellement produire plusieurs traductions.

Les processus d'un SRI multilingue consistent en : la traduction de la requête vers les langues du corpus, la recherche monolingue sur des corpus différents et la combinaison des résultats. Il est clair que ce type de SRI multilingue par recherche monolingue ne profite qu'à un utilisateur qui possède suffisamment de connaissances pour déchiffrer les documents obtenus dans la langue cible. Il est courant d'être capable de lire une langue étrangère mais il est plus difficile de trouver les justes termes pour un requête précise : c'est sur ce point que ce type de système de RI multilingue peut apporter une aide précieuse à l'utilisateur. Il est aussi toujours envisageable de compléter cette phase de recherche multilingue par une traduction même approximative des documents trouvés, dans le but de donner à l'utilisateur une idée sur la pertinence réelle des documents proposés. Il sera alors à même de sélectionner ceux qui l'intéressent le plus et qui méritent une traduction soignée.. Selon Nie [Nie 2002b], la séparation des deux tâches, la traduction et la recherche, dans un système de recherche d'information multilingue pose les problèmes suivants :

1. La sélection des termes traduits est indépendante de la distribution des termes d'origines. Cela veut dire qu'un terme traduit est choisi, mais qu'il n'a pas la même valeur de discrimination pour les documents.

2. La mesure d'incertitude de la traduction n'est pas intégrée dans le processus de la recherche : la phase de traduction est un tâche incertaine mais la mesure d'incertitude de cette phase n'est souvent pas conservée et prise en compte lors de la phase de recherche.

3. La phase de combinaison des résultats obtenus dans des langues différentes est délicate car les critères de pondération et la distribution des documents pertinents dans une collection multilingue n'est pas homogène. Ce point rejoint la notion de fusion de recherches menées en parallèle par des moteurs de recherche différents.

Ces dernières années, certains travaux ont proposé un modèle pour la RI multilingue dans lequel les deux tâche de traduction et de recherche sont intégrées [Hiemstra 2001], [Lavrenko et al. 2001], [Lafferty et al. 2001],[Nie 2002a]. Selon ce point de vue, il reste à résoudre l'intégration de ces deux tâches dans un modèle formel.

Dans le cadre de notre travail, nous nous posons la question suivante : est-ce que l'on peut résoudre ces problèmes par un modèle de recherche d'information qui permette de mieux représenter le contenu du document et de réaliser un calcul de pertinence entre la requête et

les documents dans des langues différentes ? Avant d'y répondre, nous allons examiner les différents systèmes et les approches proposés dans la littérature.

3.3 Evaluation

Pour évaluer la performance d'un SRI multilingue, une méthode souvent utilisée est de comparer la précision moyenne sur 11-point (cf.[Salton et al 83]) du système avec celle d'un SRI monolingue correspondant. Par exemple, l'évaluation de la performance d'un SRI multilingue anglais-français peut être réalisée comme dans la figure 3.3-1 suivante. On comparera la précision obtenue avec un SRI monolingue et celle obtenue avec un SRI multilingue.

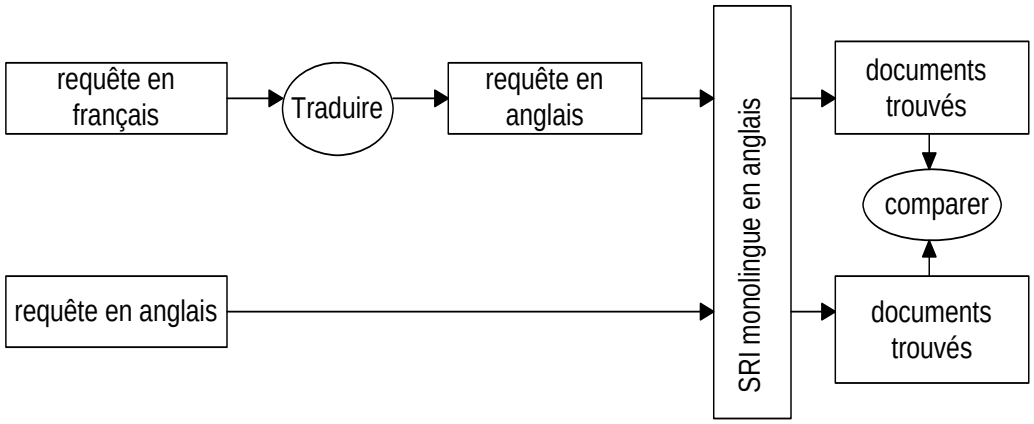


Figure 3.3-1 Evaluation d'un SRI multilingue

3.4 Classification des approches

Les approches pour construire un SRI multilingue sont divisées en deux grands types : les approches utilisant un *vocabulaire contrôlé* comme langage d'indexation prédéfini, et celles qui extraient des termes d'indexation du contenu de document (*texte libre*). Pour avoir une vue générale de la recherche d'information multilingue, voir l'étude de Douglas W. Oard et Bonnie J. Dorr [Oard 96].

Dans les approches basées sur un vocabulaire contrôlé, on utilise une liste prédéfinie de termes d'indexation multilingues pour indexer automatiquement ou manuellement les documents.

La première expérience d'utilisation d'un thésaurus multilingue prédéfini est due à Salton en 1970 [Salton 70]. Dans son expérimentation, il a utilisé une liste de concepts multilingues

anglais – allemand construite manuellement à partir d'une traduction de l'anglais vers l'allemand. L'expérimentation a été réalisée sur un corpus contenant 468 résumés en allemand et 1095 résumés en anglais. La précision moyenne a été d'environ 95% par rapport à celle d'un système monolingue. De cette expérimentation, Salton a déduit que « *cross-language processing…is nearly as effective as processing within a single languag*e ».

Il est très rare de disposer, pour indexer les documents, d'un thésaurus multilingue qui couvre plusieurs domaines. Si l'on en dispose, la mise à jour d'un tel thésaurus est coûteuse en temps et délicate pour conserver une cohérence aux termes choisis. C'est pour cette raison que la plupart des travaux en RI multilingue se sont orientés vers l'indexation automatique du contenu de document (texte libre).

Les approches 'texte libre' en RI multilingue sont divisées en trois grands axes. Le premier axe est appelé « *traduction des documents* ». Cette approche utilise un logiciel de traduction automatique ou une traduction manuelle de tous les documents d'une langue vers l'autre. Le deuxième axe propose des méthodes de « *traduction de la requête* ». Ce type d'approche a attiré beaucoup de chercheurs car il est a priori plus facile à réaliser. Le troisième axe utilise un « *langage artificiel* » ou « *langage pivot* » pour représenter les documents et les requêtes qui sont écrits dans des langues différentes.

Les approches pour construire un SRI multilingue peuvent être représentées par la figure suivante (3.4-1) :

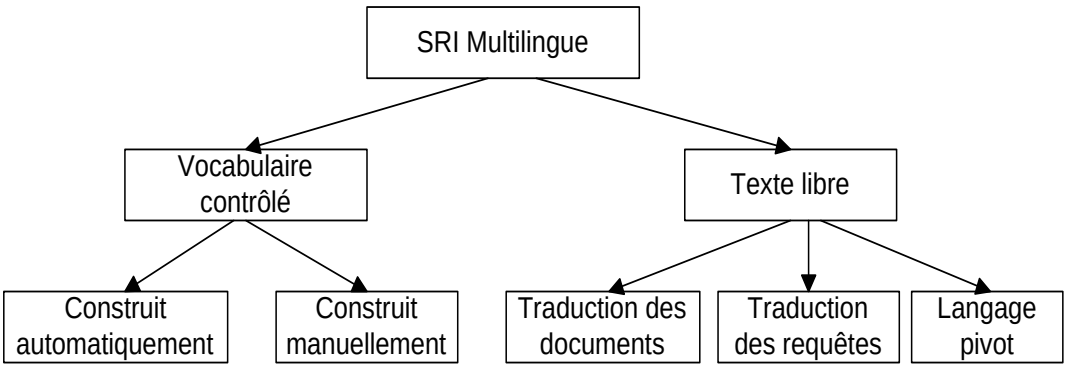


Figure 3.4-1 Classification des approches d'un SRI multilingue

La plupart des travaux sur la recherche d'information multilingue sont des approches « texte libre » et notre travail s'inscrit aussi dans cette direction. Dans la suite, nous présentons plus en détail quelques approches « texte libre » pour donner un vue générale de cette direction en RI multilingue.

3.5 Traduction de document

La première approche consiste à traduire tous les documents d'une langue vers une autre. Une traduction manuelle représente un énorme travail et prend beaucoup de temps. Accessoirement on doit multiplier l'espace de stockage puisque chaque document est représenté en autant d'exemplaires qu'il y a de langues cibles. Par contre, l'utilisation d'un logiciel de traduction automatique produit une traduction de qualité très souvent insuffisante. C'est pourquoi la plupart des approches se sont orientées vers la traduction de la requête. Par exemple, Dougles W. Oard et al. [Oard 98] ont utilisé le système Logos, un système de traduction automatique, pour traduire les documents en allemand de la collection SDA/NZZ de TREC-6 vers l'anglais. Ils ont obtenu une précision moyenne plus élévée que si l'on traduit les requêtes en utilisant le même logiciel Logos (0.2171 par rapport à 0.1561). On en conclut que la quantité d'information traduite peut compenser dans le cadre de la RI, la faible qualité des traductions. La traduction seule de la requête doit alors conduire à de plus mauvais résultats, et le manque de contexte pour choisir la bonne traduction des termes, doit être compensé par d'autres sources d'information.

3.6 Traduction de la requête

En général, les requêtes sont composées de mots simples, donc la traduction de la requête est facile à réaliser. Mais la difficulté que l'on rencontre pour traduire une requête est le manque de contexte ce qui conduit à interprétations erronées. La traduction de la requête est réalisée de l'une des manières suivantes :

1. L'utilisation d'un logiciel de traduction automatique ;
2. L'utilisation d'un dictionnaire bilingue ;
3. L'utilisation d'un corpus parallèle ou comparable.

Avant de donner des approches ou techniques dans cette direction, nous représentons précisément les notions de corpus parallèle et de corpus comparable qui sont utilisés dans les approches présentées.

- Un *corpus parallèle* est un corpus qui contient deux ensembles de documents, dans deux langues différentes, où chaque document d'une langue possède son équivalent traduit dans l'autre langue.

- Un *corpus comparable* est un corpus qui contient également deux ensembles de document, chacun dans une langue différente. Cependant, les couples de documents ne sont pas des traductions exactes, mais ils parlent tous les deux du même sujet.

3.6.1 Utilisation d'un logiciel de traduction automatique

L'utilisation d'un logiciel de traduction automatique est l'approche la plus directe. Il y a deux facteurs qui influencent le résultat de la traduction. L'un est que la requête est souvent courte, elle manque donc de contexte pour la traduction. Et l'autre est la qualité de logiciel de traduction. La traduction peut transformer le sens de la requête d'origine. De plus, l'utilisation d'un logiciel de traduction automatique ne permet pas d'intervenir lors de la phase de traduction. Donc, la qualité de la requête traduite (du point de vue de la RI) est dépendante du logiciel utilisé.

Les problèmes de cette approche sont [Nie 2004]:

1. Choix incorrect de la traduction du mot ou terme :

   Par exemple : « organic food » est traduite par « nourriture organique » alors que la bonne traduction est « nourriture biologique ».

2. Syntaxe incorrecte

   Par exemple : « human-assisted machine translation » est traduite par « traduction automatique humain-aidée » alors qu'une bonne traduction possible peut être « traduction automatique assistée manuellement »

3. Traduction des mots inconnus

   Par exemple : le nom propre Bérégovoy peut être traduit par Bérégovoy ou Beregovoy. Les noms propres en particulier et les entités nommées en général doivent subir un traitement particulier. Il faut qu'ils soient identifiés surtout s'ils ont un sens possible comme "Bill Gates" ("Les porte de la facture") ou s'ils doivent trouver un équivalent phonétique dans la langue cible comme c'est le cas par exemple pour le chinois vers l'anglais.

3.6.2 Utilisation d'un dictionnaire bilingue

Les approches utilisant un dictionnaire bilingue peut être divisées en deux types :

1. Traduction de chaque mot de la requête séparément :

    a. Choix du premier terme correspondant dans le dictionnaire comme bonne traduction;

    b. Choix de tous les termes possible en traduction

2. Traduction de tous les mots de la requête globalement :

    a. Choix des termes de la traduction qui produisent une meilleure cohésion entre eux.

Une étude sur des approches se basant sur un dictionnaire bilingue a été réalisée par Pirkola et al. [Pirkola et al. 2001]. Les problèmes principaux associés à cette approche sont :

1. la couverture de dictionnaire (i.e des termes ne peuvent pas être traduits parce qu'ils n'existent pas dans le dictionnaire) ou autrement dit, comment traiter des termes non traduit parce que ce sont des termes composés nouveaux, des noms propres, des termes d'un domaine spécialisé ou des variantes de morphologie du terme.

2. l'ambiguïté de la traduction : il faut choisir la traduction la plus correcte dans le contexte où est utilisé le terme.

Nous commençons cette partie par la description d'une expérimentation de Lisa Ballesteros et Croft en 1996 pour montrer l'impact de ces problèmes sur la qualité du SRI. Dans leur travail, Lisa Ballesteros et Croft ont utilisé un dictionnaire bilingue anglais – espagnol pour traduire les requêtes. Etant donnée une requête $q = \{s_1,s_2,\ldots,s_n\}$ dans une langue source S, pour chaque terme s_i de la requête, ils choisissent à partir du dictionnaire un ou plusieurs termes qui ont les sens correspondant à s_i dans le dictionnaire. S'il n'existe pas de terme correspondant, ils conservent s_i dans la requête traduite. En RI multilingue, la manière la plus simple pour traiter des mots intraduisibles est alors de les conserver tel quels dans la requête de la langue cible. Le résultat montre une baisse (-50, - 60%) par rapport au SRI monolingue équivalent [Ballesteros et al. 96]. Ce résultat peut s'expliquer par le fait qu'aucun des deux problèmes évoqués ci-dessus n'est véritablement pris en compte. Cependant, cette expérimentation peut être utilisée comme une base pour comparer les différentes autres solutions.

Ballesteros et Croft, 1996 [Ballesteros et al 96] ont réalisé également une expérimentation dans laquelle ils utilisent un dictionnaire bilingue pour traduire la requête, ainsi que deux corpus d'apprentissage dans les deux langues. Ces corpus ne sont pas alignés. Ils ont utilisé une technique appelée « bouclage de pertinence » pour désambiguïser les termes candidats. Le bouclage de pertinence est une technique d'expansion de requête utilisant les termes trouvés dans des documents pertinents pour la requête. La requête initiale Q_1 dans la langue L1 est étendue via un bouclage de pertinence en la soumettant à la partie de documents dans la langue L1 du corpus d'apprentissage, la requête étendue Q_1' obtenue est traduite par un dictionnaire bilingue (chaque terme dans la requête en langue L1 donne tous les termes correspondants dans la langue L2). La requête traduite Q_2' est encore une fois étendue par les documents pertinents dans la langue L2 dans le corpus d'apprentissage pour obtenir la requête traduite Q_2.

Cette technique est illustrée par la figure suivante :

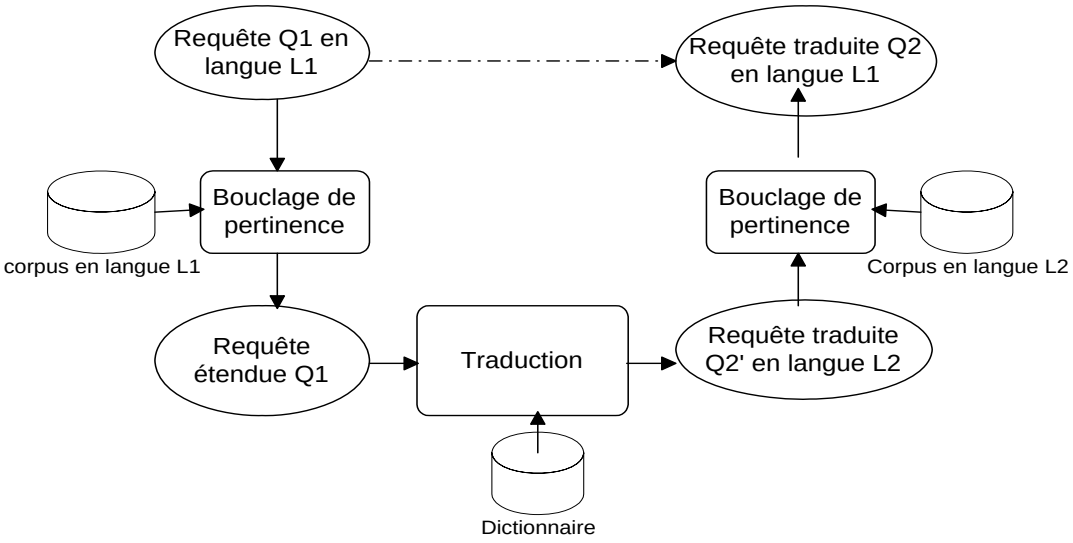


Figure 3.6-1 Technique de bouclage de pertinence

La précision moyenne est 0,1242, soit 51% de la précision moyenne de l'approche monolingue. Le résultat est donc meilleur grâce à l'utilisation de deux bouclages de pertinence.

Des méthodes de traduction des groupes de mots ou des syntagmes de la requête sont utilisé pour éviter les problèmes de traduction inhérent à une traduction mot à mot.. Les groupes de mots examinés pour la traduction peuvent être des mots adjacents [Ballesteros et al. 98] ou deux mots ayant une relation syntaxique entre eux [Gao et al. 20002]. Il est bien sur évident

que la conservation des syntagmes pour une traduction, et surtout dans le cas de mots composés, évite une dérive de sens. Il faut cependant disposer, d'une part d'un dictionnaire qui contienne des syntagmes, et d'autre part, il faut un minimum d'analyse de la requête pour repérer correctement ces syntagmes. Le cas des mots composés, est assez facile à régler du fait de leur stabilité (ils sont très souvent présents dans les dictionnaires de traduction). Par contre dans un cadre plus général, les termes exprimés par des syntagmes sont plus fastidieux à énumérer, car d'une part un terme est la désignation d'une notion précise dans un domaine, et donc la création de nouveaux termes est permanente notamment dans les domaines scientifiques; et d'autre part, ils sont souvent sujets à variation. Par exemple, les mots composés "pomme de terre", "garde barrière" sont stables, mis à part l'usage du tiret "-" qui tend à disparaître, par contre, les termes techniques comme "charge admissible" existe aussi de manière plus précise comme "charge admissible maximale", ou "charge admissible par roue". Ce dernier exemples est issu du dictionnaire terminologique du Québec qui recense plus de 90 variations de termes (avec leur traduction en anglai) pour le seul terme "charge".

Les syntagmes sont extraits de la requête et traduits en utilisant un dictionnaire qui contient des syntagmes [Grefensteette et al 96]. Si un syntagme ne se trouve pas dans le dictionnaire, il est traduit en se basant sur une mesure de cooccurrence [Ballesteros et al 98] ou sur des corpus d'apprentissage alignés [Davis 97b].

Dans la suite, nous présentons quelques techniques de désambiguïsation.

3.6.2.1 Désambiguïsation en utilisant l'étiquette syntaxique du mot

Davis et al. 1997, ont réalisé une expérimentation sur un SRI multilingue anglais-espagnol en utilisant un dictionnaire bilingue avec des étiquettes syntaxiques pour chaque mot. Pour chaque mot de la requête en langue source, ils n'ont choisi dans le dictionnaire que les termes correspondants ayant la même étiquette syntaxique que celle du mot source. La précision moyenne est 0,19, par rapport à celle de 0,14 obtenue par traduction sans désambiguïsation. On observe donc une légère augmentation de la précision [Davis et al. 97a].

3.6.2.2 Désambiguïsation basée sur le calcul de cooccurrences

Plusieurs expérimentations utilisent le calcul de cooccurrences pour désambiguïser la traduction. Ces expérimentations sont différentes au niveau du calcul de cooccurrences.

Nous présentons la méthode de calcul de cooccurrences utilisée par Lisa Ballastières, étant donnés deux termes t_1, t_2 dans la langue source étiquetés syntaxiquement. Pour chaque terme t_i, on a dans le dictionnaire une collection de termes correspondants dans la langue cible. Ces termes ont la même étiquette syntaxique que t_i. Supposons que A soit l'ensemble des termes qui correspondent à t_1 et B ceux à t_2. On calcule la mesure de cooccurrence de toutes les paires *(a,b)* où *a* appartient à A et *b* appartient à B par la formule :

$$em(a,b) = \max(\frac{n_{ab} - En(a,b)}{n_a + n_b},0)$$

où n_a, n_b sont les nombres d'occurrences de a et b dans le corpus, n_{ab} est le nombre de fois où a,b sont en cooccurrence dans le document. $En(a,b) = \frac{n_a n_b}{N}$ où N est le nombre des fenêtres dans le corpus. Les paires (a,b) sont rangées selon le score em(a,b) et celle qui a le score le plus élevé est choisie comme traduction de la paire (t_1,t_2). S'il y a plus d'une paire qui ont le score le plus haut, elles sont toutes choisies comme traduction de la paire (t_1,t_2).

Ballesteros et al. 1998 [Ballesteros et al. 98] ont réalisé une expérimentation qui combine les trois techniques de désambiguïsation : le bouclage de pertinence, l'utilisation des étiquettes syntaxiques du mot et la mesure de cooccurrences. Le résultat est amélioré de manière appréciable. En effet, la précision moyenne est 91% de précision moyenne en monolingue.

Cette approche peut être vue comme une approche pour traduire des groupes de mots en remplacement de la méthode mot à mot.

3.6.2.3 Désambiguïsation basée sur la mesure de similarité entre des termes

Mirna Adriani et C.J Rijsbergen, 2000 [Adriani et al. 2000] ont proposé une technique d'indentification des syntagmes potentiels dans l'ensemble des mots traduits obtenus à partir des mots de la requête. Ils utilisent la mesure de similarité entre les termes appelée « *Dice similarity coefficient* » [van Rijsbergen 79]. Cette mesure de similarité est calculée comme suit :

$$sim(x,y) = 2\sum_{i=1}^{N}(w'(x,i) \times w'(y,i)) / (\sum_{i=1}^{N} w^2(x,i) + \sum_{i=1}^{N} w^2(y,i))$$

avec

w(x,i) = la pondération du terme x dans le document i

w(y,i) = la pondération du terme y dans le document i

w'(x,i) = w(x,i) si le terme y apparaît aussi dans le document i, sinon 0

w'(y,i) = w(y,i) si le terme x apparaît aussi dans le document i, sinon 0

N = nombre de documents dans le corpus

La pondération w(x,i) du terme x est calculée en utilisant la mesure tf*idf comme dans le modèle vectoriel [Salton et al 83].

En se basant sur cette mesure de similarité, ils recherchent des syntagmes potentiels dans la langue cible qui seront choisis comme la traduction des termes dans langue source. Le processus est le suivant : étant donnée une requête dans la langue source ayant *m* termes $q=\{t_1,t_2,\ldots t_m\}$, cette technique produit *p* syntagmes potentiels $\{p_1,p_2\ldots.p_m\}$. Chaque p_i contient un terme $t_{i,k}$' appartenant à S_i, l'ensemble des traductions possibles de t_i obtenues à l'aide d'un dictionnaire, et un autre terme $t_{j,l}$' appartenant à un autre ensemble S_j ($j \neq i$), telle que la mesure similarité sim($t_{i,k}$', $t_{j,l}$') est la plus grande parmi les couples concernant les autres termes de S_i . L'algorithme pour construire les p_i est le suivant :

1 Pour chaque t_i (i=1 à m) appartenant à q, retrouver l'ensemble S_i des termes dans le dictionnaire qui correspondent à t_i

2 Pour chaque S_i (i=1 à m)

2.1 Pour chaque $t_{i,k}$' (k =1 à | S_i |) appartenant à S_i

2.1.1 Pour chaque S_j (j=1 à m $\wedge$ j $\neq$ i), rechercher $t_{j,l}$'$\in$ S_j où sim($t_{i,k}$',$t_{j,l}$') est maximum

2.1.2 Chercher $t_{j,max}$' parmi les $t_{j,l}$' obtenus à l'étape 2.1.1 tel que :

$$t_{j,max}' = \text{argmax}_{tj,l'} \{sim(t_{i,k}',t_{j,l}') \mid j=1 \text{ à } m \wedge j \neq i\}$$

et former un syntagme $p_{i,j}$' contenant $t_{i,k}$' , $t_{j,max}$'

Choisir p_i dans l'ensemble des $p_{i,j}$' (j=1 à $|S_i|$) tel que la similarité entre ces deux termes soit maximum

Ensuite, les syntagmes potentiels dans P sont utilisés pour former la requête traduite ou pour étendre de la requête traduite. Les auteurs ont utilisé le corpus de test en anglais de TREC AP avec 24 requêtes en espagnol. Le résultat est remarquable. Il est de 119,41% par rapport à la

traduction simple mot à mot. Et la précision moyenne par rapport à celle obtenue en monolingue est de 64%.

3.6.2.4 Constat

Comme nous l'avons dit ci-dessus, les approches basées sur un dictionnaire bilingue pour traduire la requête peuvent arriver à une précision de 91% par rapport à un SRI monolingue. Cette approche est limitée par le problème de la disponibilité d'un dictionnaire de traduction, l'incomplétude du dictionnaire de traduction, ainsi que la désambiguïsation de la traduction. Nous observons que la traduction des groupes de mots ou des syntagmes est plus efficace que la traduction mot à mot. Les techniques de désambiguïsation qui se basent sur le calcul des cooccurrences entre des termes ont donné des résultats acceptables.

3.6.3 Utilisation de corpus parallèles ou comparables

Afin d'éviter le problème de la couverture du dictionnaire, on peut construire automatiquement un thésaurus multilingue à partir de corpus comparables ou de corpus parallèles. L'idée principale de cette approche est de se baser sur un corpus d'apprentissage qui est aligné, pour construire automatiquement un thésaurus multilingue. La différence entre ces approches est la mesure de similarité utilisée : la similarité entre deux vecteurs [Sheridan et al. 96] ou la similarité probabiliste [Nie et al. 98].

Les problèmes que rencontrent ce type d'approche sont:

1. La disponibilité de corpus parallèles : il n'est pas facile de trouver des corpus parallèles pour des paires de langues minoritaires. Il y a des approches qui exploitent Internet pour construire des corpus parallèles [Nie et al. 99]

2. L'alignement des unités choisies : afin de calculer la mesure de similarité, on a besoin d'un processus d'alignement des unités des documents dans les deux langues du corpus parallèle. Une unité d'alignement peut être un document, un paragraphe, une phrase, un syntagme ou un mot. Plus l'unité est petite, plus l'algorithme doit être sophistiqué.

Dans la suite, nous présentons deux approches qui utilisent deux moyens différents de calcul de la mesure de similarité.

3.6.3.1 Construction d'un thésaurus de similarité multilingue

Paraic Sheridan et al. 1996 ont proposé une méthode appelée 'Thésaurus de similarité' (similarity thesaurus). Cette approche est basée sur un thésaurus de similarité multilingue construit automatiquement pour traduire la requête. Afin de construire automatiquement ce thésaurus, on a besoin d'un corpus parallèle aligné document par document. Les documents alignés sont concaténés afin d'obtenir un seul document qui est donc multilingue, l'ensemble des documents multilingues obtenus constitue le corpus d'apprentissage.

Le corpus d'apprentissage est indexé et l'on obtient l'ensemble des termes d'indexation multilingues. Chaque terme d'indexation t_i est représenté par un vecteur dans l'espace des documents du corpus d'apprentissage, $t_i=(w_{i1},\ldots,w_{iN})$ où w_{ij} est la pondération de terme t_i dans le document j, et N est le nombre de document du corpus d'apprentissage. La valeur de w_{ij} est calculée par tf*idf comme dans le modèle vectoriel [Salton et al. 83]). On observe que deux termes s_i dans la langue L_1 et t_k dans la langue L_2, l'un étant la traduction de l'autre, ont une même distribution dans le corpus et que les deux vecteurs associés à s_i et t_k sont proches.

Les auteurs construisent alors une matrice de similarité $S[s_{ij}]$ entre les termes des deux langues du corpus d'apprentissage. L'élément s_{ij} est une mesure de similarité entre le terme i dans la langue L_1 et le terme j dans la langue L_2 (le cosinus de l'angle entre les deux vecteurs). Avec cette matrice, on peut « traduire » la requête q_s en langue source en replaçant chaque termes s_i de q_s par les terme t_j les plus similaires à s_i.

Les auteurs ont réalisé des expérimentations en allemand – français. Le thésaurus de similarité est construit sur un corpus parallèle allemand/français du droit suisse, qui contient 91,310 termes en allemand et 43,066 termes en français. La précision moyenne, dans le cas où ils ont choisi les 5 termes dans langue cible les plus similaires à un terme de la langue source est 15% plus élevée que celle obtenue en monolingue (0.3297 par rapport à 0.2863) [Sheridan et al. 96].

3.6.3.2 Un modèle de traduction probabiliste basé sur des documents du Web

Jian-Yun Nie et al. 1998 [Nie et al. 1998] ont proposé un modèle probabiliste de traduction. Le modèle probabiliste de traduction est basé sur une distribution probabiliste $Pr(p_t|p_s)$ où p_s est une phrase dans la langue source et p_t est une phrase dans la langue cible (target). Les probabilités sont estimées en se basant sur un corpus parallèle aligné phrase à phrase de la façon suivante :

Supposons que les phrases p_s, p_t sont constituées deux listes de termes respectivement $s_1,\ldots,s_n$ et $t_1,\ldots,t_m$ et que p_s et p_t sont alignés par un alignement a, alors a_i indice q'une position particulière de s qui est alignée avec la position j dans t (par exemple a_2=4 exprime que t_2 est aligné avec s_4) et s_{aj} désigne un mot de p_s à la position a_j

La probabilité $Pr(p_t|p_s)$ est décomposée comme la somme sur tous les alignements possibles :

$$\Pr(p_t \mid p_s) = \sum_{a \in A} \Pr(p_t, a \mid p_s)$$

où A est l'ensemble de tous les alignements possibles

La probabilité conditionnelle $Pr(p_t,a|p_s)$ sous l'alignement a est analysée comme :

$$Pr(p_t,a|p_s) = Pr(p_t|a,p_s) \times Pr(a|p_s) = K_{s,t}\ Pr(p_t|a,p_s)$$

$Pr(a|p_s)$ est estimée par la constante $K_{s,t}$ parce que tous les alignements sont considérés équiprobables. Cela veut dire que $K_{s,t} = \Pr(a \mid p_s) = \frac{1}{|A|}$

Le cœur du modèle est t(t|s), la probabilité lexicale que quelques termes s soit traduits comme terme t. Cette probabilité est estimée en utilisant l'algorithme « Expectation Maximization » [Brown et al. 93].

La valeur $Pr(p_t|a,p_s)$ dépend du produit de probabilités lexicales de chaque couple de termes connecté par l'alignement :

$$\Pr(p_t \mid a, p_s) = C_{t,s} \prod_{j=1,m} t(t_j \mid s_{aj})$$

où

$C_{t,s}$ est une constante qui représente certaines dépendances entre les longueurs m et n des phrases p_t et p_s.

Cette méthode demande un calcul complexe de probabilités et d'alignement des documents phrase à phrase. Mais elle donne un résultat intéressant, car la précision moyenne est 94% de celle obtenue en monolingue [Nie et al. 98]

3.6.3.3 Constat

Nous avons constaté que les approches se basant sur un corpus parallèle ou comparable augmentent de façon importante la qualité des résultats. La précision arrive à 94% de celle d'un SRI monolingue dans le travail de Nie. Ces résultats permettent de penser que les approches basées sur des corpus parallèles ou comparables sont plus efficaces que celles basés sur un dictionnaire bilingue. Cependant, il reste difficile de conclure que la méthode est effectivement meilleure, car les expérimentations ont été réalisées sur des corpus différents. De plus, la faible disponibilité de corpus parallèles ou comparables est un obstacle à la mise en œuvre pratique de ces méthodes. Il faut également noter que la phase d'alignement pour un corpus parallèle ou comparable est complexe. Finalement, si le domaine du corpus parallèle/comparable n'est pas assez proche de celui du corpus de travail, le résultat de la « traduction » en sera fortement affecté.

3.6.4 Utilisation d'un pseudo langage pivot

Certaines méthodes proposent de passer la barrière de la langue en utilisant un langage pivot. Ce langage est alors utilisé pour représenter le document et la requête indépendamment des langues sources et cibles. Tout le problème est alors la définition de ce langage pivot pour la RI multilingue et la conversion et de l'enconversion entre des langues naturelles avec ce langage. La technique Latent Semantic Indexing (LSI) appliquée à la RI multilingue peut être vue comme l'introduction d'un langage pivot par changement de l'espace d'expression des vecteurs d'index sur de nouvelles dimensions concrétisant ce pivot : le document et la requête sont représentés dans un espace commun indépendant de la langue. Cette approche est basée sur le modèle vectoriel. Tout le problème réside dans la définition de l'espace vectoriel.

3.6.4.1 Technique de LSI en contexte monolingue

Le modèle LSI est une extension du modèle vectoriel. Dans le modèle vectoriel, on représente les documents et les requêtes par des vecteurs dans l'espace des termes d'indexation. Cet espace est généralement de grande dimension. La plupart des méthodes de recherche d'information sont basées sur une correspondance entre les termes de la requête et les termes des documents. C'est pour cela que le système ne peut pas retrouver des documents pertinents pour une requête s'ils ne contiennent pas de mots communs avec la requête. La technique LSI cherche à traiter automatiquement la dépendance entre des termes afin que des termes « similaires » soient représentés par un représentant commun. C'est pour cela que l'on peut

retrouver des documents qui ne contiennent pas exactement les termes de la requête, mais seulement des termes similaires.

La technique LSI examine les similarités des contextes où les termes apparaissent pour créer un espace des traits (features) dans lequel deux termes apparaissant dans des contextes similaires sont proches l'un de l'autre. Le modèle LSI utilise une méthode de l'algèbre linéaire, appelé, « *la décomposition en valeurs singulières* » *(SVD – Singular Value Decomposition en anglais)* pour découvrir les relations importantes entre des termes [Deerwester et al. 90].La figure 3.6-2 illustre l'efficacité du modèle LSI. La méthode vectorielle traditionnelle représente des documents comme une combinaison linéaire de m termes supposés orthogonaux. Au contraire, le modèle LSI représente les documents et les termes dans un espace sémantique à k dimension (k<m). Deux termes utilisés dans des contextes similaires sont représentés par des vecteurs proches dans l'espace réduit.

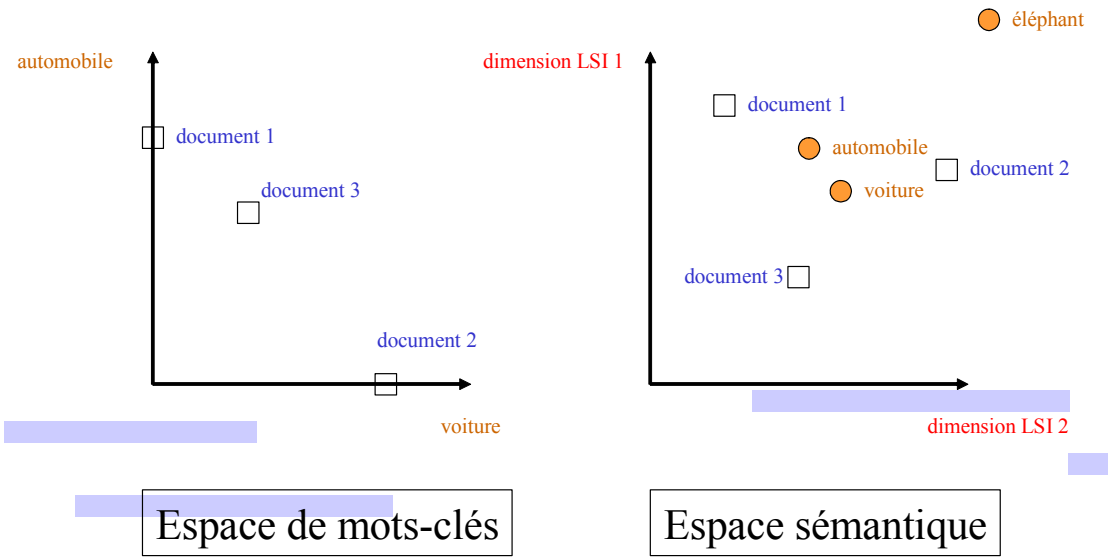


Figure 3.6-2 Deux espaces de représentation des documents

Le résultat de la décomposition en valeurs propres est l'ensemble des vecteurs représentant la localisation de chaque terme et document dans l'espace réduit à k dimensions. Le processus de l'interrogation consiste à utiliser les termes d'une requête pour localiser un point dans l'espace réduit. La requête est construite par la somme des vecteurs pondérations des termes appartenant à la requête. Les documents sont ordonnés selon leurs valeurs de similarité avec la requête.

Tout d'abord nous allons expliquer quelques notions relatives à la décomposition en vecteurs propres. Soit une matrice A d'ordre (m×n), les valeurs propres de A sont égales à la racine carrée positive des valeurs propres du produit matriciel de la matrice A par sa transposée A^t. Les valeurs propres d'une matrice sont calculées par la méthode appelée « La décomposition en valeurs propres ».

La décomposition en valeurs propres décompose une matrice A d'ordre (m x n) en un produit de trois matrices. $A = U . S . V^t$

Où U et V sont des matrices orthogonales et S une matrice diagonale.

- La matrice U d'ordre (m x m) est la matrice des vecteurs propres de $A . A^t$, avec $U^t . U = I$
- La matrice V d'ordre (n x n) est la matrice des vecteurs propres de $A^t . A$, avec $V^t . V = I$
- La matrice S d'ordre (m x n) est une matrice diagonale dont les termes $s_{ii} \geq 0$ sont les valeurs propres de A, les valeurs s_{ii} sont ordonnées de façon décroissante.

En se basant sur cette décomposition, la technique LSI s'applique à la recherche d'information de la façon suivante. Soient un corpus contenant *n* documents indexé par *m* termes d'indexation et la matrice A_{mxn} terme-document, la décomposition en valeurs propres, transforme A en un produit de trois matrices :

$$A_{m\,x\,n} = U_{m\,x\,m}\, S_{m\,x\,n} V^t{}_{n\,x\,n}$$

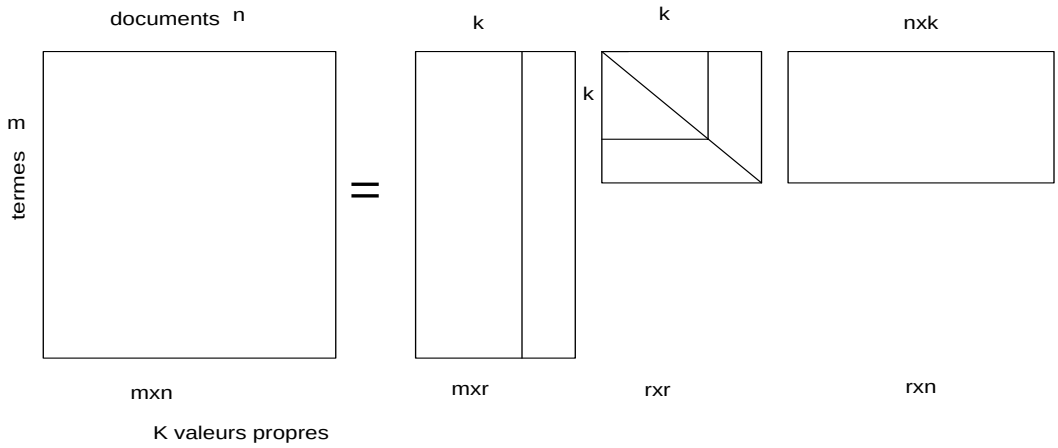


Figure 3.6-3 Décomposition en valeurs propres

Seules les valeurs propres non nulles nous intéressent et, de plus, on peut choisir de ne garder que les k plus grandes. La matrice peut alors être transformée de la façon suivante :$A_{m\,x\,n} \cong U_{m\,x\,k} \times S_{k\,x\,k} \times V^{t}{}_{k\,x\,n}$

Où

$U_{m\,x\,k}$ est la matrice des vecteurs de termes représentés dans l'espace des k plus grandes valeurs singulières, elle permet de représenter les termes dans ce nouvel espace :$V^{t}{}_{k\,x\,n}$ est la matrice qui permet de représenter les documents dans l'espace des k plus grandes valeurs propres ;

S est la matrice diagonale des k valeurs propres.

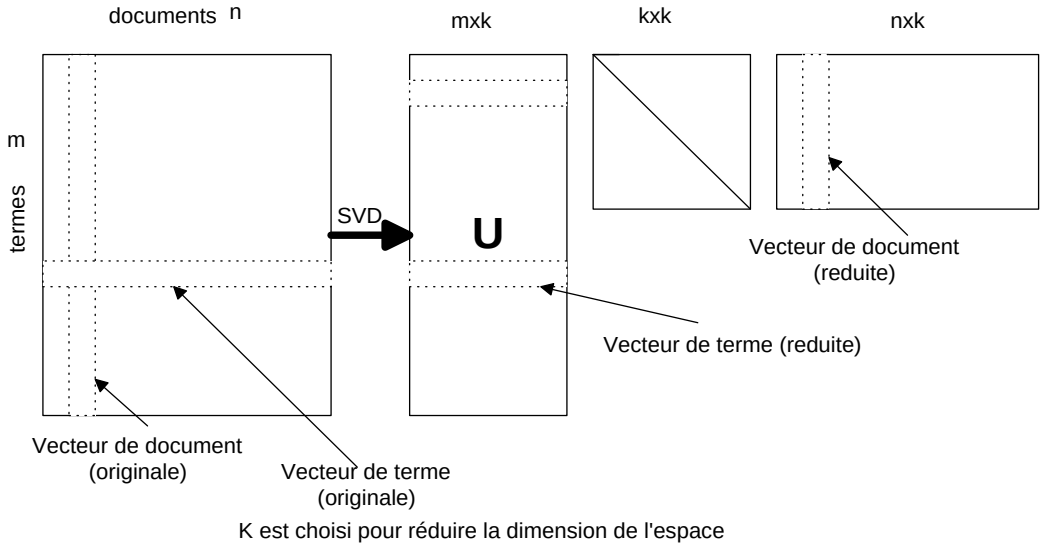


Figure 3.6-4 Technique LSI pour la RI monolingue

Soient la requête q représentée par une matrice colonne Q (m x1) et le document d représenté également par une matrice-colonne D (m x1), les représentations de q et d dans le nouvel espace des k valeurs propres sont obtenues à l'aide de la matrice U. Q se transforme en $Q' = U^t .Q$ et D en $D' = U^t .D$. On peut ensuite calculer la similarité entre les deux vecteurs (représentés par les matrices-colonnes Q' et D') comme le cosinus de l'angle des deux vecteurs.

$$sim(\vec{q},\vec{d}) = \cos({}^tU\,\vec{q},{}^tU\,\vec{d})$$

3.6.4.2 La technique LSI pour un SRI multilingue

Dans un SRI multilingue, il faut utiliser un corpus parallèle comme corpus d'apprentissage, on combine alors les deux documents alignés du corpus d'apprentissage en un seul document bilingue, et l'on construit ensuite une matrice « *termes x documents* » A du corpus des documents bilingues. On utilise la technique de « décomposition en valeurs propres » pour décomposer A, et U est utilisé comme un espace commun de représentation indépendant des langues.

Susan T. Dumais et al., 1996 [Dumais et al. 96] ont appliqué la méthode LSI (Latent Semantic Indexing) à un SRI multilingue. Ils construisent les deux matrices A de taille m1×n et B de taille m2×n où A est la matrice termes×documents de la partie de documents dans la langue source du corpus d'apprentissage et B est la matrice termes×documents de la partie de documents dans la langue cible, et les colonnes se correspondant dans A et B sont des documents traduits l'un à l'autre. On construit une matrice multilingue *termes×documents* C de taille (m1+m2)×n en « concaténant » les colonnes de A et de B. On applique la technique de « *la décomposition en valeurs propres »* sur C pour obtenir un matrice U comme présenté dans la partie précédente. Dans ce cas, U peut se voir comme une combinaison de deux matrice U_A (m1×k) et U_B (m2×k)

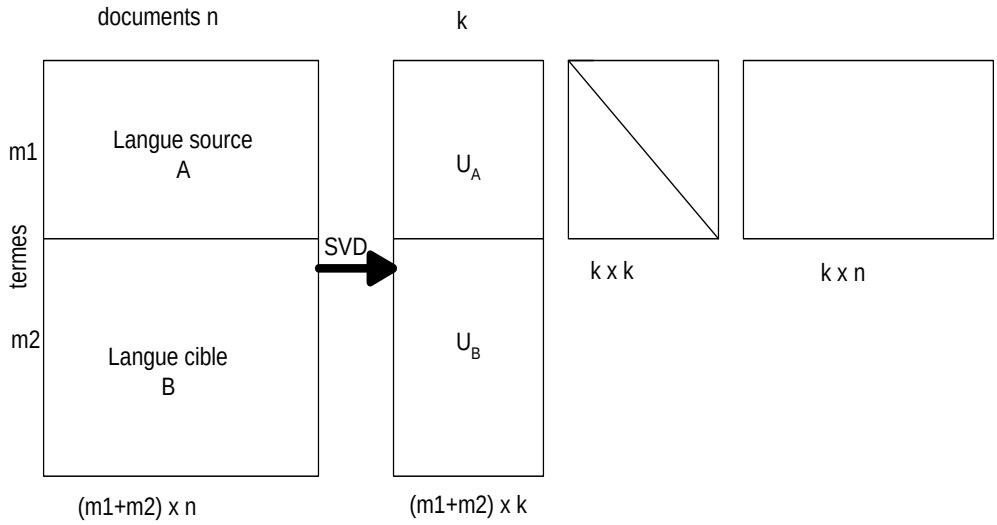


Figure 3.6-5 Technique LSI pour la RI multilingue

On utilise U_A pour la transformation de la requête et U_B pour la transformation du document. La similarité entre une requête Q et un document D est calculée comme le cosinus de l'angle des deux vecteurs associés aux matrices colonnes

$Q' = U^t_{A\,.\,Q}$ et $D' = U^t_B\,.\,D$

$$sim(\vec{q},\vec{d}) = \cos({}^tU_A\vec{q}, {}^tU_B\vec{d})$$

Dans leur expérimentation, les auteurs ont obtenu une précision 99% par rapport à celle obtenue en monolingue. Cette technique semble donc très efficace sur les corpus testés. Cependant, le calcul de la décomposition en valeurs propre est coûteux en temps et en place mémoire, le choix de la valeur k n'est pas facile. De plus, cette méthode nécessite également un corpus aligné de taille importante et couvrant le domaine des textes à indexer.

3.7 Résumé

La RI multilingue peut être vue comme un « mariage » entre la recherche d'information et la traduction automatique. En général, la performance d'un SRI multilingue est plus basse que celle d'un SRI monolingue. Elle varie de 50% à 90% d'un SRI monolingue. Chaque approche soulève des problèmes particuliers que nous récapitulons dans le tableau suivant :

L'approche	**Les problèmes**
Traduction automatique	La qualité de la traduction Le coût
Dictionnaire bilingue	La disponibilité du dictionnaire La couverture thématique du dictionnaire La désambiguïsation de la traduction
Corpus parallèle	La disponibilité de corpus parallèles La complexité de la phase d'alignement Le domaine du corpus parallèle par rapport au domaine du corpus de travail

Toutes les approches présentées ci-dessus traitent séparément les deux tâches de traduction et de recherche. Cependant, elles n'abordent pas du tout l'aspect pondération des termes traduits ni celui de l'intégration des mesures d'incertitude de la traduction. Nous pensons au contraire que ces deux aspects doivent être étudiés de manière explicite, et qu'ils doivent être partie prenante d'un modèle plus complet de RI multilingue.

Dans la partie suivante, nous présenterons des approches qui proposent un modèle général pour un SRI multilingue. Avec de tels modèles, la fonction de correspondance permet de calculer directement la pertinence entre une requête dans une langue et un document dans une autre langue, et la phase de traduction est alors une phase intégrée au calcul de la pertinence.

3.8 Modèle pour la SRI multilingue

3.8.1 Modèle de langue (language modelling) pour la recherche d'information

Nous présentons l'interprétation de Greiff [Greiff et al. 2003] du modèle de langue dans le domaine de recherche d'information comme suit.

Un modèle de langage est une distribution de probabilités sur un ensemble de chaînes de caractères. Plus formellement, soit un vocabulaire V, l'ensemble V* des chaînes sur V est défini par :

$$V^* = \{ \sigma \mid (\exists\, n \in N)\ (\exists\, v_1,\ldots,v_n \in V\ )[\ \sigma = \langle v_1, \ldots, v_n \rangle]\}$$

Une distribution de probabilités sur V*, est une fonction Pr définiée par :

$$\Pr : V^* \rightarrow [0,1] \text{ telle que } \sum_{\sigma \in V^*} \Pr(\sigma) = 1$$

La plupart des approches basées sur un modèle de langue traitent les documents eux-mêmes comme des modèles de langue et une requête comme une chaîne de caractères obtenue de façon aléatoire à partir de ces modèles. On mesure la pertinence d'un document avec une requête par la probabilité que la requête soit observée en sachant le document D trouvé : Pr(Q|D). Autrement dit, Pr(Q|D) est la probabilité de Q connaissant le modèle de langue de D [Lavrenko et al. 2003].

Par exemple, dans les travaux de Miller et al. [Miller et al. 99], Song et Croft [Song et al. 98] et Hiemstra [Hiemstra 2001], la requête est traitée comme une séquence de mots indépendants et la probabilité Pr(Q|D) est estimée par la formule suivante :

$$\Pr(Q \mid D) = \prod_{w \in Q} \Pr(w \mid D) \times \prod_{w \notin Q} (1 - \Pr(w \mid D)) \qquad (3.8.1)$$

où w est un mot de la requête.

En RI, un corpus contient N documents. Ce corpus est représenté par un l'ensemble de termes d'indexation $\{t_1,\ldots,t_m\}$. On peut avoir N+1 modèles de langage : un modèle global pour le corpus et N modèles locaux pour les N documents [Jones et al. 2003]. La probabilité Pr(Q|D) est estimée en combinant les modèles locaux de document avec le modèle global de corpus par la formule suivante :

$$\Pr(Q \mid D) = \prod_{w \in Q} ((1 - \lambda) \Pr(w) + \lambda \Pr(w \mid D)) \qquad (3.8.2)$$

où Pr(w) est une distribution de probabilités dans le modèle global de corpus

Pr(w|D) est une distribution de probabilités dans le modèle local du document D.

λ est un paramètre de lissage (smoothing).

3.8.2 Vers un modèle unifié pour la recherche d'information multilingue

Pour la recherche d'information multilingue, la requête Q et les documents D sont dans des langues différentes, on ne peut pas estimer directement Pr(Q|D) comme présenté ci-dessus, car Q ne peut pas être traité comme un échantillon du modèle de langue de D. On peut estimer la probabilité que D implique Q indirectement via des traductions possibles Q' de Q vers la langue du document D. On obtient alors :

$$\Pr(Q \mid D) = \sum_{Q' \in T(Q)} [\Pr(Q \mid Q') \times \Pr(Q' \mid D)] \qquad (3.8.3)$$

où T(Q) est l'ensemble des traductions de Q

Dans cette formule, la facteur Pr(Q|Q') est la probabilité que Q' soit une traduction de Q.

Il y a plusieurs approches qui suivent cette idée, comme celles de Hiemstra et Jong [Hiemstra et Al. 99], Xu et al. [Xu et al. 2001], Berger et Lafferty [Berger et al. 99] et Federico et al. 2002.

Par exemple, Federico et al. 2002 [Federico et al. 2002] ont proposé une approche « N-best query translation » pour la recherche d'information multilingue, dans laquelle le modèle de traduction est un modèle de Markov caché [Rabiner 90], la probabilité jointe d'une paire (Q,Q') est décomposée comme suit, et nous permet d'avoir le modèle de traduction.

$$\Pr(Q = w_1, ..., w_n, Q' = w_1', ..., w_n') = \Pr(w_1') \Pr(w1 \mid w_1') \prod_{k=2}^{n} \Pr(e_k \mid e_{k-1}) \times \Pr(w_k \mid w_k')$$

Etant donnés un modèle de traduction et une requête Q, les meilleures traductions peuvent être trouvées par un algorithme de recherche comme celui de Viterbi [Rabiner 90]. La probabilité Pr(Q'|D) est calculée en se basant sur la formule (3.8.2) du modèle de langue.

Les auteurs ont réalisé des expérimentations sur le corpus de CLEF 2001 en utilisant trois méthodes de traduction : le logiciel Babefish (Systran), 1-Best, et la traduction manuelle. Les mesures de précision moyenne sont présentées dans la table suivante :

Type de traduction	Précision moyenne	%
Babelfish	0.4034	77,54%
1-best	0.4350	83,62%
Manuelle	0.5202	100%

Tableau 1 Les résultats de méthodes de traduction différentes

3.9 Conclusion

Dans ce chapitre, nous avons présenté le contexte de la recherche en recherche d'information multilingue. Nous observons que la plupart des approches se concentrent sur la traduction de la requête par un dictionnaire bilingue, ou par un corpus parallèle ou comparable. Certaines approches s'orientent vers une représentation indépendante de la langue. Les plus récentes cherchent un modèle unifié pour la recherche d'information multilingue, comme le modèle de langage.

Dans notre travail, nous nous sommes orienté vers un modèle général de recherche d'information multilingue utilisant la richesse et la structure des termes d'indexation complexes. Pour atteindre cet objectif, nous avons dans une première étape, étudié un modèle d'indexation structuré monolingue. Ce modèle est ensuite étendu pour prendre en compte le multilinguisme.

Dans le chapitre suivant, nous présentons le modèle d'indexation structuré qui va servir de base à notre proposition.

Chapitre 4 Modèle de recherche d'information basé sur un réseau bayésien des expressions d'indexation

Bruza et al. [Bruza et al 93a], [Bruza et al 93b] [Bruza et al 94] ont proposé un modèle de recherche d'information basé sur l'inférence probabiliste en utilisant un réseau bayésien des expressions d'indexation. Dans cette partie, nous présenterons les idées principales de ce modèle.

Tout d'abord, nous présentons les notions concernant le modèle logique pour la recherche d'information, le réseau bayésien, et ensuite l'approche proposée par Bruza qui est la base de notre proposition dans le chapitre suivant.

4.1 Un modèle logique pour la recherche d'information

4.1.1 Introduction

L'idée principale du "modèle logique" pour la recherche d'information, est de considérer qu'un document D est pertinent pour une requête Q si après application d'un certain nombre de règles d'inférence, on peut montrer que la requête peut être dérivée du document D. Lorsque D et Q sont représentés par des formules logiques, cette dérivation se ramène à vérifier que D implique Q, notée $D \rightarrow Q$. Cependant en RI, on ne peut utiliser la logique classique car il est très rare qu'une telle implication logique soit directement vérifiée. Par convention, nous disons que la pertinence d'un document D pour une requête Q est mesurée par une mesure d'incertitude associée aux dérivations nécessaires pour montrer qu'à partir de D on peut dériver la requête Q. Cette mesure de pertinence est notée $P(D \rightarrow Q)$ [Lamas 98], [Nie 92]. Le problème est alors de déterminer comment évaluer cette incertitude. Nous allons présenter dans la suite le modèle de Bruza et al.

4.1.2 Modèle et système de recherche d'information : définitions

Nous proposons les définitions suivantes pour pouvoir ensuite décrire la notion de correspondance basée sur la dérivation.Un modèle de recherche d'information est défini par un triplet $\mathcal{M} = <\mathcal{O}, \mathcal{C}, \mathcal{R}>$ où :

- $\mathcal{O}$ est un ensemble d'objets d'information (documents) ;
- $\mathcal{C}$ est un langage de descripteurs complexes pouvant être, par exemple, des termes simples, des syntagmes nominaux, des termes structurés, etc. C'est le langage qui permet d'indexer les documents;
- $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{C}$ est une relation d'indexation qui relie un objet d'information $\mathcal{O}$ à des descripteurs.

Un système de recherche d'information est défini par un triplet $\Delta = <\mathcal{M}, \mathcal{S}, \mathcal{P}>$ où :

- $\mathcal{M}$ est un modèle de recherche d'information ;
- $\mathcal{S}$ est un ensemble de règles d'inférence certaines ;
- $\mathcal{P}$ est un ensemble de règles d'inférence incertaines.

4.1.3 Notion de dérivation

Nous pouvons alors proposer une notion de dérivation pour modéliser la correspondance entre une requête et un document. Soit $\Delta = <\mathcal{M}, \mathcal{S}, \mathcal{P}>$ un système de recherche d'information et $\mathcal{C}$ est un langage de description de $\mathcal{M}$. Pour un ensemble de descripteurs $d \subseteq \mathcal{C}$, un descripteur $x \in \mathcal{C}$, et une règle $s \in \mathcal{S}$, on utilise $d \vdash_s x$ pour exprimer que x peut être déduit de d par l'application de la règle d'inférence certaine. En général, $d \vdash_\Delta x$ désigne une séquence d'une ou de plusieurs étapes de dérivation concernant des règles certaines. De même, on utilise $d \rightsquigarrow_p x$ pour exprimer que x est déduit de d via une règle incertaine $p \in \mathcal{P}$; $d \rightsquigarrow_\Delta x$ désigne une séquence d'une ou de plusieurs étapes de dérivation concernant des règles certaines avec au moins une utilisation d'une règle incertaine.

Afin de simplifier la notation, on peut enlever l'indice Δ dans $\vdash_\Delta$ et $\rightsquigarrow_\Delta$. Dans la suite du chapitre, nous n'indiquerons plus l'indice Δ.

4.1.4 Inférence et pertinence

Soient D un document appartenant à l'ensemble des objets d'information $\mathcal{O}$, $\mathcal{X}(D)$ l'ensemble des expression d'indexation (ou descripteurs) représentant le contenu de D, $\mathcal{X}(D) \subseteq \mathcal{C}$, et une

requête constituée d'un seul descripteur $q \in \mathcal{C}$. Le processus est alors d'utiliser des règles d'inférence afin de déduire la requête q à partir de $\mathcal{X}(D)$ que l'on note ainsi :

$$\mathcal{X}(D) \vdash q \Rightarrow D \rightarrow q$$

Bruza et al. ont [Bruza et al. 93a] proposé un moyen de calculer la mesure de pertinence $P(D \rightarrow Q)$ par une probabilité. Ce calcul utilise un langage d'indexation et une structure des termes d'indexation sous forme d'un réseau bayésien qui permet de calculer des probabilités entre des termes d'indexation. Ces probabilités sont utilisées comme une mesure incertaine pour le processus de dérivation. Nous présentons tout d'abord les notions de base du réseau bayésien, et son utilisation en recherche d'information.

4.2 Réseau bayésien

Dans cette partie, nous présentons les notions de base du réseau bayésien [Patrick 91], [Neapolitan 2004],

4.2.1 Notions de base d'un graphe acyclique orienté

4.2.1.1 Définition (graphe orienté)

Un graphe orienté est un couple G = (V,A). V est l'ensemble des nœuds et $A \subseteq VxV$ est l'ensemble des arcs. Un arc orienté de x vers y est noté (x,y).

4.2.1.2 Définition Parents(N)

Les parents du nœud N, désigné par Parents(N), sont l'ensemble des nœuds du graphe reliés au nœud N par des arcs orientés vers N. C'est l'ensemble des *prédécesseurs* directs de N.

4.2.2 Définition d'un réseau bayésien

4.2.2.1 Définition

Un réseau bayésien, aussi appelé graphe causal probabiliste, est défini par un triplet (V,R,PC) dont les éléments sont :

- L'ensemble V des événements $\{V_i\}$ d'un domaine examiné. Ces événements sont associés à chaque nœud du graphe. Chaque événement V_i est associé à deux statuts possibles :V_i est réalisé, noté V_i et V_i n'est pas réalisé, noté $\neg V_i$

- L'ensemble R de toutes les relations entre les événements. Ces relations sont des arcs orientés où la direction du lien indique la causalité : {Vpères → Vfils}

- L'ensemble PC de toutes les probabilités conditionnelles d'un fils connaissant ses pères $Pr(V_{fils} | V_{père1},\ldots,V_{pèren})$. Pour les nœuds V_j n'ayant pas de père, la probabilité $Pr(V_j|\varnothing)$ se réduit à la probabiliste à priori $Pr(V_j)$.

4.2.2.2 Matrice de liaison

Les probabilités d'un nœud sont calculées en considérant toutes les combinaisons possibles des valeurs de ses parents. Ces probabilités conditionnelles sont enregistrées dans une matrice appelée *matrice de liaison (link matrix)*. Chaque ligne de cette matrice correspond à une valeur possible du nœud fils, et chaque colonne à une combinaison possible des valeurs des nœuds parents. Puisque chaque nœud prend une valeur dans l'ensemble {vrai, faux}, la matrice d'un nœud fils ayant n nœuds parents aura la dimension $2x2^n$.

Par exemple : On a un réseau ayant trois nœuds A, B, C. Supposons que A est vrai si et seulement si B et C, sont tous les deux vrais, on a une matrice de liaison du nœud A comme dans le tableau suivant :

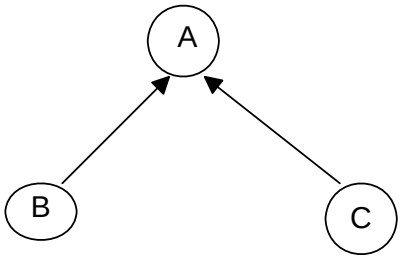


Figure 4.2-1 Un exemple de réseau bayésien

La matrice de liaison du nœud A, qui contient les probabilités conditionnelles de A est

	BC	¬BC	B¬C	¬B¬C
A	1	0	0	0
¬A	0	1	1	1

Les probabilités du nœud A sont calculées en se basant sur la réalisation ou la non réalisation des évènement B et C.

4.2.2.3 Calcul de la probabilité $Pr(V_{fils} | V_{père1}, \ldots, V_{pèren})$.

Soit un événement A dépendant de n combinaisons possibles des valeurs de ses pères C_i (i=1,n). Les évènements C_i sont des évènements différents et incompatibles deux à deux (c'est à dire qu'on ne peut avoir deux causes réalisées simultanément).

La probabilité Pr(A) est calculée par

$$\Pr(A) = \sum_{i=1}^{n} \Pr(A, C_i) \qquad (4.2.1)$$

On a

$$\Pr(A, C_i) = \Pr(A \mid C_i) \times \Pr(C_i) \qquad (4.2.2)$$

Remplaçant 4.2.2 dans 4.1.1, on a

$$\Pr(A) = \sum_{i=n}^{n} \Pr(A \mid C_i) \times \Pr(C_i) \qquad (4.2.3)$$

Cette équation fournit une base pour calculer la crédibilité d'un événement A qui est réalisé comme la somme des crédibilités de tous les cas possibles pour lesquels A peut être réalisé.

Par exemple : Un pommier perd ses feuilles pour deux causes possibles : l'automne et la maladie. Cette connaissance peut être représentée par un graphe bayésien.

V = {Automne, Maladie, Perte de feuilles}. A chaque Vi est associé un ensemble {vrai, faux}

Le réseau R est le graphe suivant

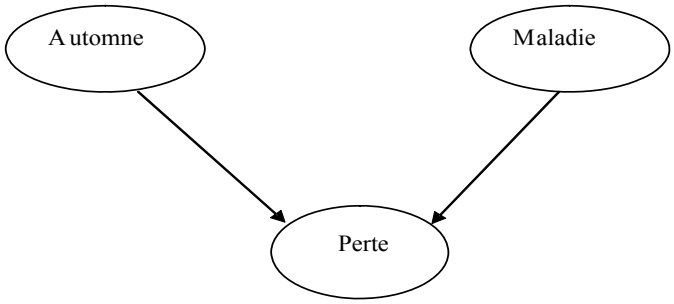


Figure 4.2-2 Un exemple de réseau Bayésienne

L'événement « perte de feuille » noté « Perte » a 4 combinaisons possibles des deux causes qui peuvent se réaliser ou ne pas se réaliser :

« Maladie∧Automne », « ¬Maladie∧Automne », Maladie∧¬Automne »,
« ¬Maladie∧¬Automne ». Donc, la *probabilité a priori* de l'événement « perte de feuille » est calculée comme suit :

$$\begin{aligned}\Pr(\text{Perte}) = \; & \Pr(\text{Perte}|\text{Maladie}\wedge\text{Automne})\times\Pr(\text{Maladie})\times\Pr(\text{Automne})+ \\ & \Pr(\text{Perte}|\text{Maladie}\wedge\neg\text{Automne})\times\Pr(\text{Maladie})\times\Pr(\neg\text{Automne})+ \\ & \Pr(\text{Perte}|\neg\text{Maladie}\wedge\text{Automne})\times\Pr(\neg\text{Maladie})\times\Pr(\text{Automne})+ \\ & \Pr(\text{Perte}|\neg\text{Maladie}\wedge\neg\text{Automne})\times\Pr(\neg\text{Maladie})\times\Pr(\neg\text{Automne})\end{aligned}$$

On suppose que les probabilités a priori des nœuds feuilles et les probabilités conditionnelles sont stockées dans des matrices :

Matrice de liaison de « Maladie » :

Maladie = vrai	Maladie = faux
0,1	0,9

Matrice de liaison de « Automne »

Automne = vrai	Automne = faux
0,25	0,75

Matrice de liaison de « Perte »

	Automne = vrai		Automne = faux	
	Maladie = vrai	Maladie=faux	Maladie=vrai	Maladie=faux
Perte= vrai	1	1	0,9	0,05
Perte=faux	0	0	0,1	0,95

En se basant sur ces probabilités, on peut calculer la probabilité d'un événement donné si on connaît l'information sur la probabilité des événements causes de ce dernier

La probabilité a priori pour le nœud « Perte » peut être calculée en utilisant la formule ci-dessus :

$$\begin{aligned} Pr(Perte) = \quad & = 1x0,1x0,25 + 0,9x0,1x0,75+1x0,9x0,25+0,05x0,75x0,9 \\ & = 0.10375 \end{aligned}$$

Supposons que l'on ait observé que l'événement « maladie » est réalisé, c'est-à-dire que maladie est « vrai » alors Pr(Maladie) = 1 et Pr(¬Maladie)=0. Dans ce cas la probabilité de réalisation de l'événement « Perte » doit être mise à jour. Il n'y a que deux causes possibles « Maladie ∧ Automne », « Maladie∧¬Automne ». La probabilité de « perte » devient donc :

$$\begin{aligned} Pr(Perte) = \quad & Pr(Perte|Maladie\wedge Automne)\times Pr(Maladie)\times Pr(Automne)+ \\ & Pr(Perte|Maladie\wedge\neg Automne\ )\times Pr(Maladie)\times Pr(\neg Automne) \\ Pr(Perte) = \quad & 1x1x0,25+0,9x1x0,75 = 0,925 \end{aligned}$$

Le processus de mise à jour est réalisé pour tous les nœuds concernés du graphe. On appelle cela le processus de propagation (ou la propagation). En effet, lorsqu'on observe la réalisation d'un événement, cette réalisation modifie la probabilité des éléments du réseau qui lui sont liés.

4.3 Inférence probabiliste sur un réseau bayésien

4.3.1 Langage d'indexation

Soit un ensemble $\mathcal{D}$ de documents. Chaque document $D\in\mathcal{D}$ est représenté par un ensemble de descripteurs $\mathcal{X}(D)$ appelés *« expressions d'indexation »*. Celles-ci peuvent être des termes simples, des termes composés ou des structures complexes comme des syntagmes, des locutions… Dans l'approche de Bruza, [Bruza et al 93a], une expression d'indexation est une chaîne de termes séparés par des connecteurs qui représentent la relation entre les termes. Un terme prend une valeur dans l'ensemble T et un connecteur prend une valeur dans l'ensemble C. T est l'ensemble des substantifs, des substantifs qualifiés par des adjectifs ou mots composés. C est un ensemble de prépositions et un connecteur spécifique appelé « connecteur

nul », désigné par °, c'est le cas lorsqu'il n'y a pas de préposition. Une expression d'indexation peut être décrite par une grammaire sous forme BNF de la façon suivante :

Expr → ε | Nexpr

Nexpr → Termes {Connecteur Expr}*

Termes → t, t ∈ T

Connecteur → c, c ∈ C

où ε désigne une expression vide

Cette grammaire définit le langage d'indexation $\mathcal{C}=\mathcal{L}(T,C)$ et on peut observer que $\mathcal{L}(T,C)$ contient le langage d'indexation limité aux termes simples comme $\mathcal{L}(T,\varnothing)$

Une expression d'indexation, à la différence d'un terme simple, possède une structure syntaxique. Il y a plusieurs manières d'interpréter une expression d'indexation, donc de la structurer. Bruza et al. ont défini des priorités sur les connecteurs afin de structurer une expression d'indexation.

Par exemple : Soit l'expression d'indexation « *simulation of behavior of stockprices by computer program*» parenthésée comme *[simulation of [[behavior of [stockprices]] by [computer [program]]*, elle peut se représenter par un arbre appelé *arbre associé* ou *graphe associé.*

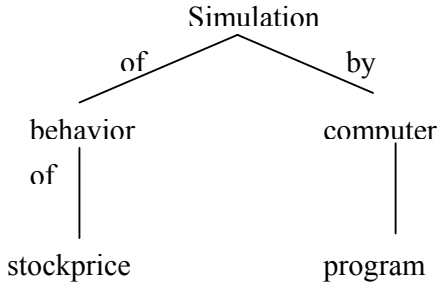


Figure 4.3-1 Graphe de dépendances

4.3.2 Treillis des termes d'indexation

4.3.2.1 Le treillis d'une expression d'indexation

Les expressions d'indexation sont plus intéressantes pour représenter le contenu sémantique d'un document que les termes simples ou les termes composés car elles permettent de représenter également des relations entre des termes, ce qui précise l'interprétation sémantique que l'on peut en faire.

Définition 1 : (sous-expression) :

Une *sous-expression* d'une expression d'indexation donnée $i \in \mathcal{L}(T,C)$ est une expression associée à un sous-graphe du graphe associé à i.

Définition 2 (ensemble des sous-expressions)

Etant donnée une expression d'indexation $i \in \mathcal{L}(T,C)$, l'ensemble des variations de i est l'ensemble de toutes les sous-expressions de i. On les dénote par $\wp(i)$.

$$\wp(i) = \{ j \mid j << i \}$$

Où la relation << est la relation « *est une sous-expression de*»

Mode de construction du treillis à partir du graphe de dépendance

Dans [Bruza et al 93b], les auteurs donne la méthode suivante pour construire le treillis des sous-expressions :

1. À partir d'un graphe de dépendances G ayant n nœuds, on détermine les plus grands sous-graphes de G (i.e. les sous-graphes ayant n-1 nœuds)
2. Pour chaque sous-graphe obtenu à l'étape précédente, on répète l'étape 1 jusqu'à ce qu'on obtienne des graphes n'ayant qu'un seul nœud.

Par exemple, à partir du graphe de dépendance (figure 4.3-1) de l'expression d'indexation précédente *[simulation of [[behavior of [stockprices]] by [computer [program]]*. Cette expression de longueur 5 peut être décomposée en deux sous-expressions de longueur 4 qui correspondent à tous les sous-graphes de longueur 4 qui peuvent être extraits de ce graphe, ce qui donne :

Exp1= [simulation of [[behavior of [stockprices]] by[computer]]]

Exp2= [simulation of [[behavior] by [computer [program]]

De même l'expression *Exp1* peut être décomposée en deux sous -expressions de longueur 3 :

[simulation of [behavior of [stockprices]]]et [simulation of [[behavior by[computer]]

et ainsi de suite.

Le treillis de toutes les sous-expressions est représenté dans la figure suivante :

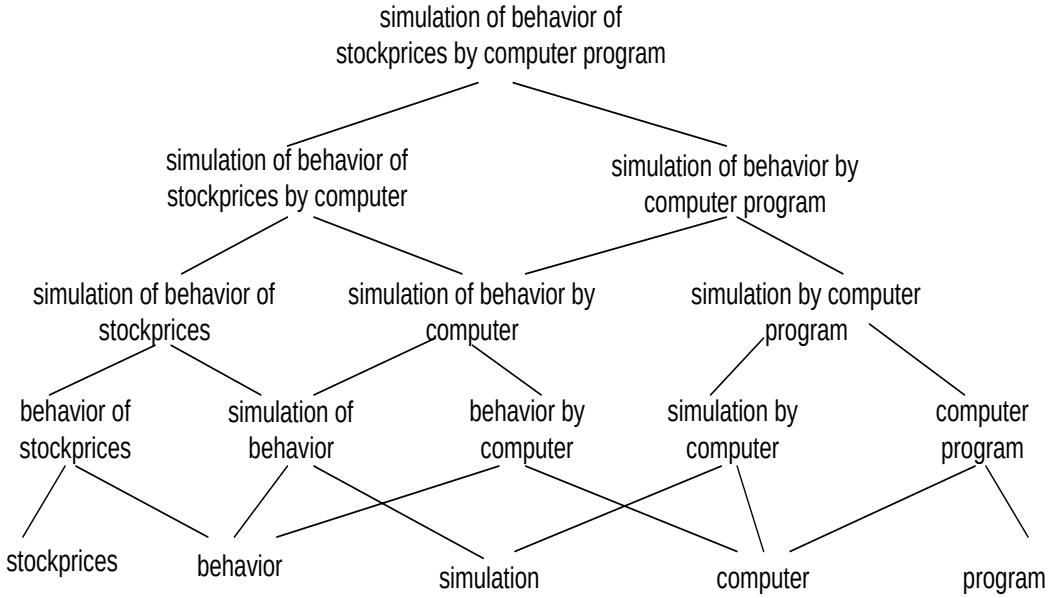


Figure 4.3-2 Exemple d'un treillis de syntagmes nominaux

Les auteurs ont utilisé cette structure pour construire un réseau bayésien qui sert de base aux inférences probabilistes.

4.3.2.2 Treillis des termes d'indexation

Soit l'ensemble $\mathcal{D}$ des documents d'un corpus, I est un ensemble d'expressions d'indexation extraites de tous les documents de $\mathcal{D}$, $I \subseteq \mathcal{L}(T,C)$. L'ensemble des termes d'indexation de $\mathcal{D}$ est donc l'ensemble de toutes les sous-expressions construites à partir des expressions de I, soit $\mathfrak{I}$ cet ensemble $\mathfrak{I}=\bigcup_{i\in I}\wp(i)$.

4.3.3 Règle d'inférence

4.3.3.1 Règle d'inférence certaine

Définition : Soient i et j des expressions d'indexation appartenant à $\mathcal{L}$(T,C) et << la relation « *est une sous-expression de* » sur $\mathcal{L}$(T,C). Alors, si j est une sous-expression de i, j peut être déduit de i ou :

$$j << i \Rightarrow i \vdash j$$

4.3.3.2 Règle d'inférence incertaine

Définition : Soit $i_1,i_2,\ldots,i_n$, $n \geq 1$, et k des expressions d'indexation dans un langage $\mathcal{L}$(T,C). Alors

$$Pr(k|i_1 \wedge i_2 \wedge \ldots \wedge i_n) > 0 \Rightarrow i_1,i_2,\ldots i_n \rightsquigarrow k$$

Afin de réaliser des inférences incertaines, Bruza et al. [Bruza et al 93a] ont construit un réseau bayésien des expressions d'indexation.

La section suivante décrit la façon dont sont calculées les probabilités.

4.3.4 Réseau bayésien des termes d'indexation

4.3.4.1 Structure du réseau

Dans la section 4.3.2, nous avons présenté un treillis des termes d'indexation. Bruza et al. utilisent ce treillis pour construire un réseau bayésien qui servira de base au mécanisme d'inférence probabiliste.

Tout d'abord, nous présentons l'ensemble V des nœuds du réseau. Chaque nœud représente un événement associé à une expression d'indexation. Cet événement est réalisé si l'expression d'indexation est observée. Pour distinguer le nom d'événement et l'expression d'indexation elle-même, on utilise des majuscules.

L'ensemble A des arcs du réseau est défini en se basant sur la relation « *est une sous-expression de* ». Si i est une sous-expression de j (i.e. i <<j), on a un arc du nœud associé à i vers le nœud associé à j.

Par exemple : Le treillis dans la Figure 4.3-2 forme le réseau suivant :

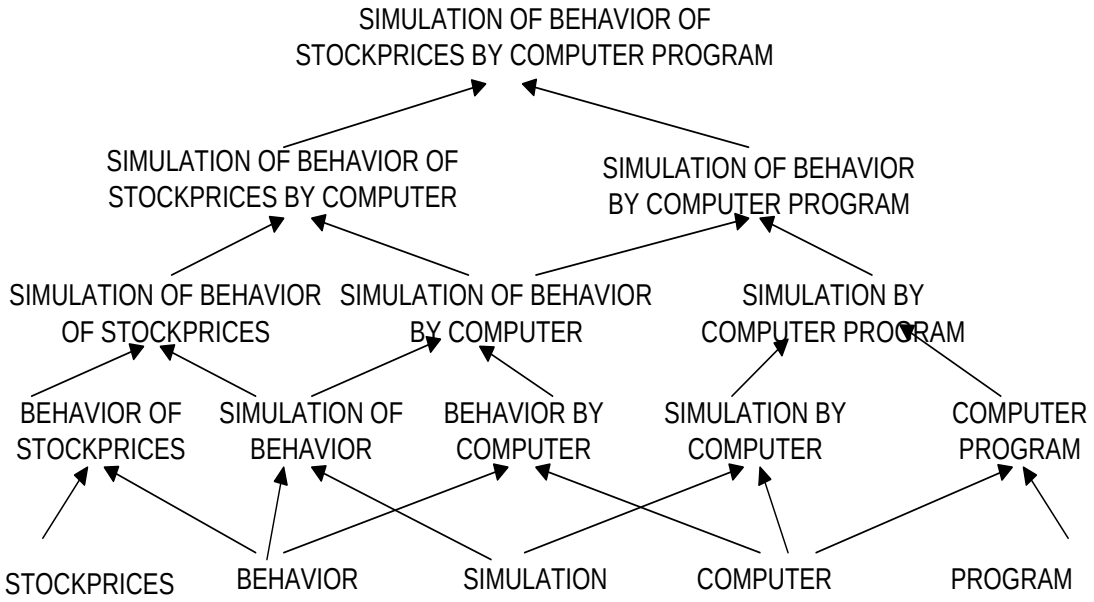


Figure 4.3-3 Exemple d'un réseau bayésien de syntagmes nominaux

4.3.4.2 Calcul de la probabilité

Bruza et al. ont proposé la méthode de calcul des *probabilités a priori* selon les caractéristiques des nœuds :

1. Si le nœud est un terme simple, c'est-à-dire un terme qui ne peut pas être décomposé (i.e. le terme n'a pas de parents).

$\forall t_i \in \mathcal{X}(D) \cap T$

$$\Pr(t_i) = \lambda tf(t_i) \text{ et } \Pr(\neg t_i) = 1 - \Pr(t_i)$$

où λ est un facteur de normalisation pour obtenir $\Pr(t_i) \in [0,1]$, et $tf(t_i)$ est la fréquence du terme t_i dans le corpus.

2. Si le nœud N est une expression de longueur de 2. On définit la longueur de N, notée l(N), par le nombres de termes ($t \in T$) contenus dans N. Cela signifie que ses deux parents sont des terms simples.

soit $N_1 \in T$, $N_2 \in T$ les parents de N

La probabilité a priori de N est donnée par la formule (4.2.3)

$$\begin{aligned}\Pr(N) = \quad & \Pr(N|N_1 \wedge N_2) \times \Pr(N_1) \times \Pr(N_2) + \\ & \Pr(N|N_1 \wedge \neg N_2) \times \Pr(N_1) \times \Pr(\neg N_2) + \\ & \Pr(N|\neg N_1 \wedge N_2) \times \Pr(\neg N_1) \times \Pr(N_2) + \\ & \Pr(N|\neg N_1 \wedge \neg N_2) \times \Pr(\neg N_1) \times \Pr(\neg N_2)\end{aligned}$$

puisque N est une combinaison de N_1, N_2. On ne peut pas observer que N est réalisé si N_1 ou N_2 n'est pas réalisé. N ne peut être réalisé que si N_1 et N_2 sont réalisés, d'où :

$$\Pr(N|N_1 \wedge \neg N_2) = 0$$
$$\Pr(N|\neg N_1 \wedge N_2) = 0$$
$$\Pr(N|\neg N_1 \wedge \neg N_2) = 0$$

Il n'y a qu'un seul cas possible où l'on peut observer N, et la formule précédente devient

$$\Pr(N)=\Pr(N|N_1 \wedge N_2) \times \Pr(N_1) \times \Pr(N_2).$$

On ne peut observer N que si N_1 et N_2 sont réalisés, le nombre de cas possibles de combinaisons de N_1 et N_2 est le nombre de connecteurs. Pour cette raison, Bruza a proposé un moyen d'estimer la probabilité de $\Pr(N|N_1 \wedge N_2)$ basée sur la probabilité d'obtenir un connecteur entre N_1 et N_2. c'est-à-dire

$$\Pr(N) = \Pr(N|N_1 \wedge N_2) = \Pr(\text{connecteur entre } N_1 \text{ et } N_2).$$

3. Si le nœud est une expression d'indexation K de longueur supérieure ou égale à 3. Le processus de construction du treillis est tel qu'une expression ayant n termes ($n \geq 3$) n'est composé que de deux sous expressions I et J ayant n-1 termes. Ces deux sous-expressions ont n-2 termes communs et leur combinaison pour obtenir l'expression de longueur n est unique. Donc la probabilité de ce nœud sachant ses parents est égale à 1.

Soient trois expressions d'indexation I, J, K qui n'appartiennent pas à T et (I,K),(J,K) qui appartiennent à l'ensemble des arcs du réseau, alors

$$\Pr(K|I \wedge J) = 1$$

4.3.5 Calcul de la pertinence

L'hypothèse faite dans ces travaux de Bruza et al est que la requête ne contient qu'une seule expression d'indexation.

4.3.5.1 Définition :

Soient $\mathcal{L}(T,C)$ un langage d'expressions d'indexation, D un document, et $\mathcal{X}(D)$ l'ensemble des expressions d'indexation représentant le contenu de D. $\mathcal{X}(D) \subseteq \mathcal{L}(T,C)$. Soit Q une requête qui ne contient qu'une seule expression d'indexation $q \in \mathcal{L}(T,C)$. La probabilité de pertinence de $D \rightarrow q$, notée $P(D \rightarrow q)$ est définie de la façon suivante :

$$P(D \rightarrow q) = \max \{Pr(q|i) \mid i \in \mathcal{X}(D)\}$$

Soit une requête $q \in \mathcal{L}(T,C)$. Pour chaque document $D \in \mathcal{D}$, pour chaque $i \in \mathcal{X}(D)$, en utilisant le réseau construit, on peut calculer Pr(q|i) de la manière suivante : on soumet i au réseau comme un fait et on propage le calcul de la probabilité de q, en calculant la probabilité des nœuds situés sur les chemins qui mènent de q à i.

En général, le processus de propagation est lourd car le réseau peut contenir de nombreux nœuds. Des études ont été faites pour limiter la propagation, on peut citer le travail du J.J. IJDENS [IJDen et al.95]. Il détermine le plus petit ensemble de nœuds (sous-réseau mininal) dont la probabilité change lorsque l'on donne une expression d'indexation comme un fait.

Nous représentons ci-dessous, l'algorithme proposé par Bruza et J.J IJDENS.

4.3.5.2 Théorème

Soitent i et q deux expressions d'indexation, et Pr une distribution de probabilités sur les nœuds du réseau bayésien de expressions d'indexation.

$$Pr(q|i) = \prod_{y \in Bot(q) - Bot(i)} Pr(y \mid \wedge Parents(y))$$

où $\wedge$Parents(y) désigne la proposition $t_1 \wedge \ldots \wedge t_n$ où $t_i \in$ Parents(y)

$$Bot(i) = \{j << i \mid l(j) \leq 2 \}$$

d'où $P(D \rightarrow q) = \max \{ \prod_{y \in Bot(q) - Bot(i)} Pr(y \mid \wedge Parents(y)) \mid i \in \mathcal{X}(D)\}$

La preuve détaillée de ce théorème se trouve dans [IJdens et al. 95].

4.3.5.3 Algorithme de calcul de la probabilité de pertinence

Supposons que l'on ait une requête q ne comprenant qu'une seule expression d'indexation et et D un document. Bruza et al. ont proposé l'algorithme suivant pour calculer la pertinence, soit P(D→q) :

1. Construire l'ensemble $\wp(q)$ des sous-expressions de q , soit $\wp(q) = \{ j \mid j << q\}$
2. Calculer Bot(q) = $\{ j \in \wp(q) \mid l(j) \leq 2 \}$

   Pour tout i $\in \chi$ (D)

   Construire l'ensemble $\wp(i)$

   Calculer Bot(i)

   Si Bot(q) = Bot(i) alors Pr(q|i) =1 sinon

   $$\mathbf{Pr(q|i) = \prod_{y \in Bot(q) - Bot(i)} Pr(y|\wedge parent(y))}$$

   fin de pour tout i,
3. P(D→q)= max {Pr(q|i) | i ∈ χ (D)}

Exemple : On a un document D qui contient un seul TIS i = « simulation of behavior of stockprices by computer programs". La requête Q aussi consiste un seul TIS q = "computer simulation of earthquakes".

Les réseaux sont construits pour le TIS q et le i sont :

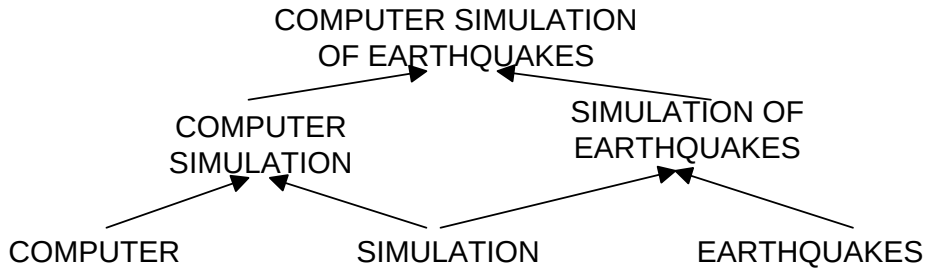


Figure 4.3-4 Réeau du TIS q

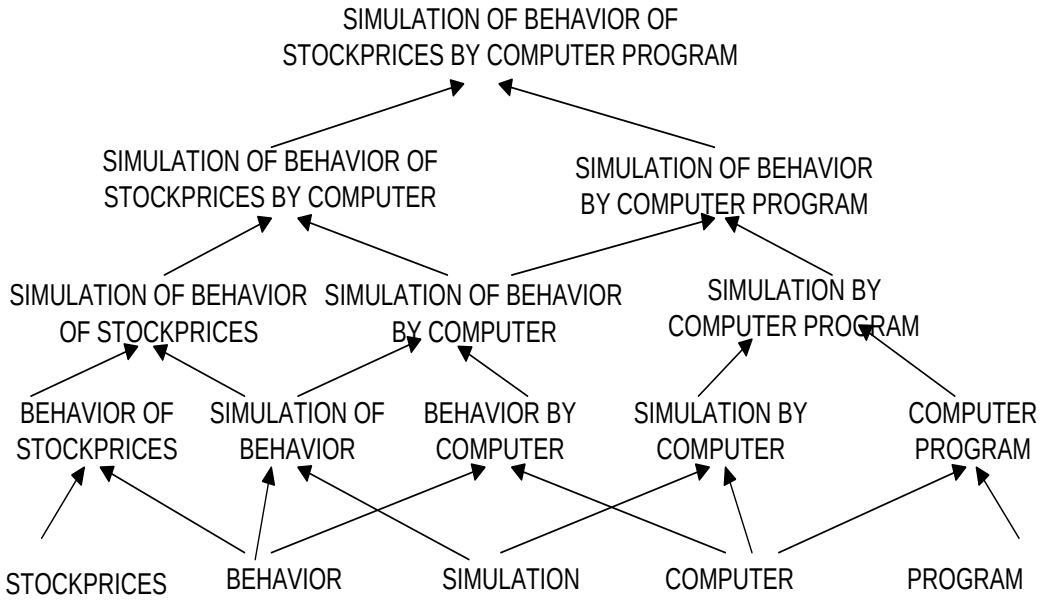


Figure 4.3-5 Réseau du TIS i

Bot(q) = {computer simulation, simulation of earthquakes, computer, simulation, earthquakes}

Bot(i) = { behavior of stockprices, computer program, simulation of behavior, simulation by computer, behavior by computer, simulation, behavior, stockprices, computer, program}

Supposons que les probabilités des termes simples sont les suivantes :

Pr(computer) = 0.25 Pr(simulation)=0.25 Pr(behavior) = 0.125

Pr(stockprices) = 0.125 Pr(program)=0.125 Pr(earthquakes)=0.125

Et supposons que la probabilité de connecteur *nu*l est 0.53 et de connecteur *of* est 0.15

Bot(q) – Bot(i) = {computer simulation, simulation of earthquakes, earthquakes}

Donc,

Pr(q|i) = Pr(computer simulation|computer ∧simulation)×Pr(simulation of earthquakes|simulation∧earthquakes)×Pr(earthquakes)

= 0.53×0.15×0.125 = 0.0103

$P(D \rightarrow q) = Pr(q|i) = 0.0130$

4.4 Conclusion

Cette approche propose un moyen de représenter le contenu du document par une structure et est donc une amélioration par rapport à une représentation par un « sac de mots ». Cette structure sert de base pour construire un réseau bayésien qui permet d'avoir une évaluation probabiliste de la pertinence entre Q et D, notée par Pr(Q|D).

Les points faibles de cette approche sont les suivants. Le premier est la phase de construction du treillis d'une expression. Cette phase est basée sur une liste des priorités de connecteurs pour définir des dépendances entre des termes. Ce choix peut produire des sous-expressions qui n'ont pas de sens. Le deuxième point est la phase de calcul des probabilités des nœuds du réseau. La probabilité des nœuds ayant deux pères est identifiée à la probabilité du connecteur les reliant. Dans ce calcul, on ne tient pas compte du type de relations existant entre deux termes, ni de la probabilité de ces termes.

Partie III : Proposition d'un modèle probabiliste d'indexation par syntagmes

Chapitre 5 Un modèle de recherche d'information

Dans ce chapitre, nous présenterons un modèle de recherche d'information dans lequel les termes d'indexation contiennent des termes simples, des termes composés et des syntagmes que nous appelons termes d'indexation syntagmatique (TIS) qui seront définis précisément dans la suite. Ces TIS sont organisés sous forme d'un réseau bayésien qui représente les relations existant entre eux. Cette représentation sert de base d'inférence pour calculer la pertinence entre la requête et les documents dans la phase d'interrogation.

Pour extraire ces TIS, un analyseur syntaxique est utilisé. Et pour construire le réseau, nous proposons un ensemble de règles de décomposition qui nous permettent de décomposer un TIS en des sous-TIS plus petits. Ces TIS sont utilisés pour construire le réseau de TIS.

Notre approche est inspirée de l' « indexation par un réseau bayésien des expressions d'indexation » de Bruza [Bruza et al. 93] et de « Réseau d'inférence » de Turtle et Croft 1991 [Turtle et al. 91] et de Wong et Yao 1991 [Wong et al. 91].

Nous présentons tout d'abord les notions concernant un terme d'indexation syntagmatique (TIS) dans la section 5.1, ensuite, dans 5.2, nous présentons la structure de l'ensemble des termes d'indexation. Dans 5.3, nous proposons une solution simple pour la recherche d'information basée sur des TIS. Une autre solution plus complexe est présentée dans la section 5.4.

5.1 Terme d'Indexation Syntagmatique (TIS)

5.1.1 Définition (terme d'indexation syntagmatique : TIS)

Dans cette partie, nous reprenons le concept de *terme d'indexation syntagmatique* (TIS) qui est proposé dans le travail du M.H. Haddad et J.P. Chevallet [Chevallet et al. 2001] afin d'obtenir un modèle d'indexation relationnel.

On a un ensemble d'étiquettes, désigné par $L=\{l_1, l_2, \ldots l_n\}$, où les l_i sont *atomes syntagmatiques*. Un atome syntagmatique peut être un mot simple ou un mot composé comme « pomme de terre »

On a aussi un ensemble de relations $R = \{R_1,\ldots,R_m\}$. Chaque R_j est une relation binaire sur L.

Si T désigne la tête du terme, les A_i désignent les arguments appelés « modifieurs »,un terme d'indexation syntagmatique I est structuré de la façon suivante:

$$I = T\ R_1\ [A_1]\ R_2\ [A_2]\ldots R_n\ [A_n]$$

Où

$T \in L$

$R_i \in R$

Soit A_i est un atome syntagmatique ($A_i \in L$) soit A_i est un terme d'indexation syntagmatique structuré

Un syntagme peut être réduit à une seule tête. Les R_i qualifient les relations entre la tête et les modifieurs. Cette qualification de la relation est optionnelle. Elle n'apparaît que si cette relation n'est pas générique. Le modifieur est défini de façon récursive. Cela veut dire qu'il peut être, soit un atome syntagmatique, soit son tour un terme d'indexation syntagmatique.

Un TIS peut être représenté sous forme d'un arbre n-aire dans lequel les nœuds sont des atomes syntagmatiques du TIS. La direction d'un arc est du mot « tête » vers son modifieur par la relation R_i. Par exemple, voici quelques formes d'arbre d'un TIS

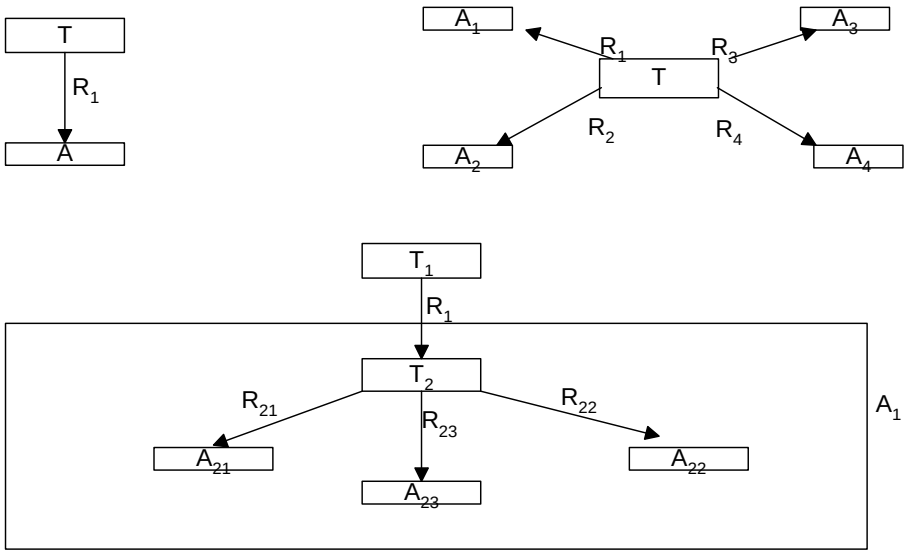


Figure 5.1-1 Un exemple des formes d'un abre

Le syntagme « *simulation of behavior of stockprices by computer program* » pris comme exemple dans le chapitre précédent est un TIS qui peut être structuré de la façon suivante :

Simulation *of [**behavior** of [stockprices]] by [**program** [computer]]*

Dans cet exemple, les mots en gras sont les mots têtes du syntagme initial ou les mots têtes des sous-syntagmes.

Le graphe représentant ce TIS est :

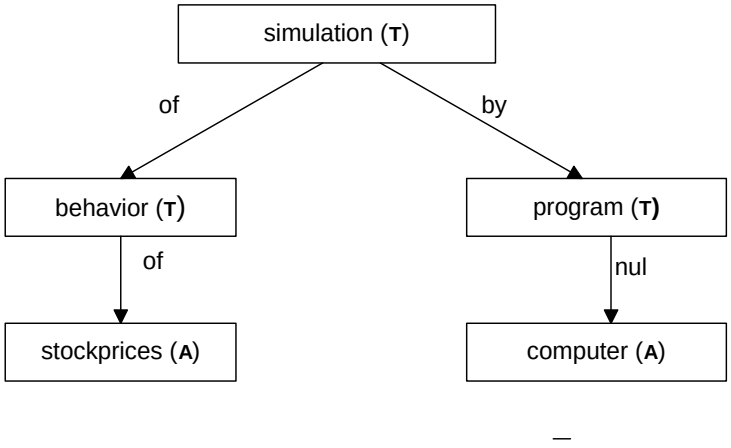


Figure 5.1-2 Un exemple du graphe de dépendance

A partir de ce graphe, nous utilisons des règles, qui seront présentées dans la partie suivante, pour décomposer ce TIS en des sous-TIS de plus en plus petits jusqu'aux atomes. Les sous-TIS obtenus seront utilisés pour construire un réseau bayésien qui met en évidence la relation de « causalité » existant entre eux. Cela veut dire que si l'on observe un ou des sous-TIS fils, on peut calculer la probabilité que l'on a d'observer ses TIS pères.

5.1.2 Règles de décomposition d'un TIS

L'objectif de cette section est de formaliser des règles de transformation de TIS de manière à obtenir un réseau de sous-termes sur lequel sera basée la fonction de correspondance. Le but de cette décomposition est de conserver autant que possible le thème du syntagme initial. Pour ce faire, nous proposons quelques hypothèses sur lesquelles nous construisons des règles de décomposition.

En premier, nous faisons l'hypothèse qu'un syntagme réduit à sa tête et à l'un de ses modifieurs prend un sens plus général que le syntagme original. Nous concrétisons cette hypothèse dans la règle suivante :

Règle 1 : Distribution de la tête

Etant donné un TIS I sous forme I = T R_1 [A_1] R_2 [A_2]…R_n [A_n], nous décomposons I en n sous TIS I_1 = T R_1 [A_1], …. I_n = T R_n [A_n], c'est à dire que nous donnons un rôle prépondérant à la tête T.

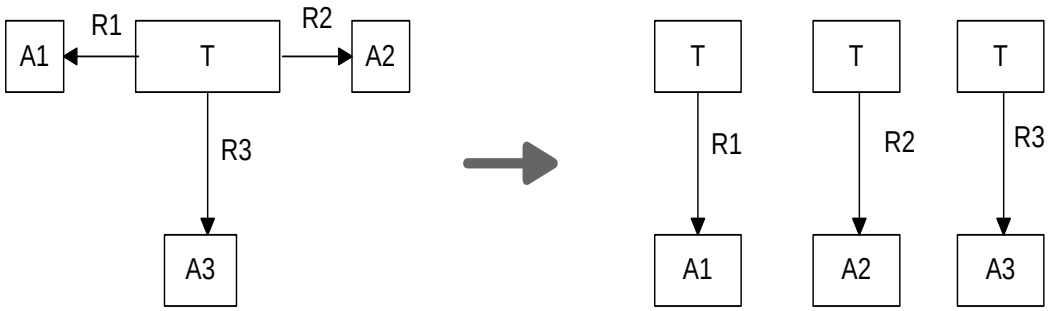


Figure 5.1-3 La règle 1

Avec cette hypothèse, le sens principal d'un TIS, représenté par le mot tête, est gardé dans les sous-TIS obtenus.

Par exemple : le TIS de l'exemple précédent se décompose en deux de la façon suivante :

I_1 = ***Simulation*** *of [**behavior** of [stockprices]]*

I_2 = ***Simulation*** *by [**program** [computer]]*

En seconde hypothèse, nous proposons d'abord qu'un syntagme conserve son sens après suppression de ses modifieurs. Nous proposons ensuite que le modifieur d'un syntagme conserve suffisamment de sens pour être utilisé en relation avec le syntagme original. Cette dernière hypothèse permet de simplifier un syntagme par désemboîtement de sa structure. Du fait de la destruction de la structure du syntagme, cette hypothèse est très probablement moins souvent valide que la précédente. Nous proposons la règle suivante :

Règle 2 : Extraction du sous arbre au premier niveau

Soit un TIS I sous forme T_1 R_1 [T_2 R_2 [A_2]], c'est à dire dont la tête T1 possède un seul modifieur qui n'est pas un atome. Nous décomposons I en deux sous-TIS I_1 = T_1 R_1 [T_2] et I_2 = T_2 R_2 [A_2]. La relation R_1 n'est conservée qu'entre les têtes des syntagmes

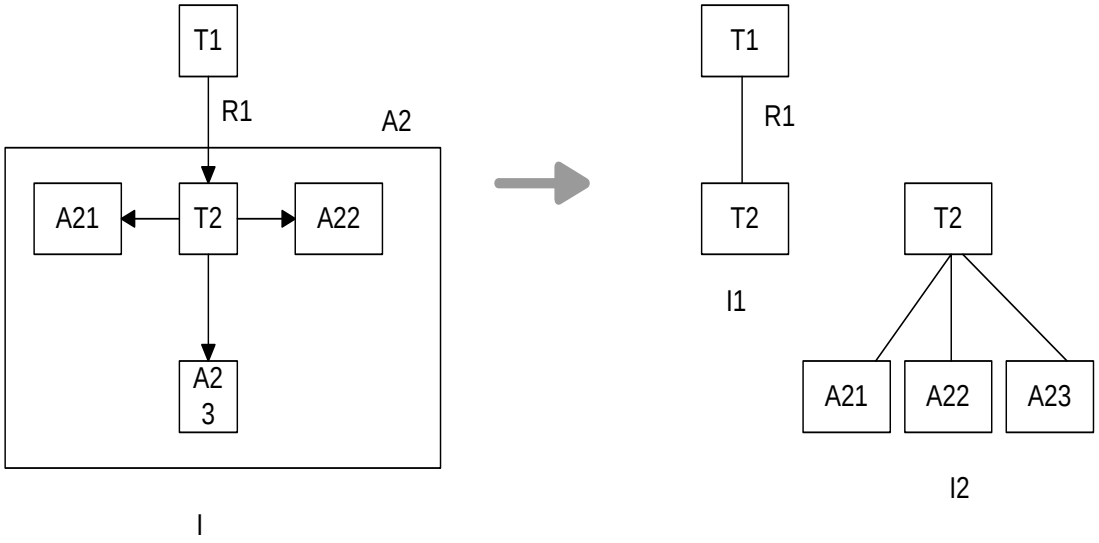


Figure 5.1-4 La règle 2

Par exemple : **Simulation** of [**behavior** of [stockprices]] est décomposé en deux

I_1 = **Simulation** of [behavior]

I_2 = **behavior** of [stockprices]

De même, **simulation** by [**program** [computer]] est décomposé en deux

I_3 = **Simulation** by [program]

I_4 = **program** [computer]

Avec cette règle, nous obtenons un sous-TIS I_1 qui garde le sens principal du TIS d'origine, et I_2 est un morceau d'information du TIS d'origine important, mais ne correspondant pas exactement au sens principal du TIS d'origine. Cette observation nous servira de guide dans la phase de calcul de la probabilité du TIS d'origine lorsque l'on a obtenu certains de ses sous-TIS

Règle 3 : Eclatement des atomes

Etant donné un TIS I sous forme T_1 R_1 $[A_1]$ où A_1 est un atome syntagmatique, nous décomposons I en deux syntagmes atomiques T_1 et A_1

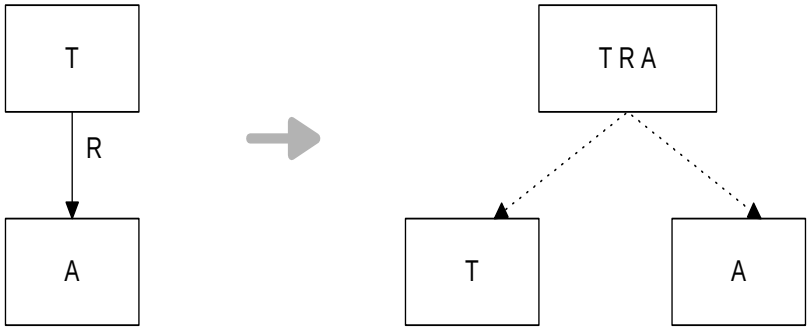


Figure 5.1-5 La règle 3

Comme pour les TIS obtenus par la règle 2, la partie de tête T garde le thème du syntagme T R [A]. Cela veut dire que si l'on a observé le terme T comme une représentant d'un document D, alors la probabilité d'observer le TIS T R [A] est plus grande quand l'on a observé le terme A.

Par exemple : ***Simulation*** *of [behavior]*, est décomposé en deux mots simples : *Simulation*, *behavior*.

Ces règles prises dans le sens inverse, c'est-à-dire, dans le sens de la construction d'un arbre, sont similaires aux opérateurs des graphes conceptuels de Sowa [Sowa 84]. Les règles 1 et 2 sont relatives à l'opération de jointure. La règle 3 n'existe pas dans le modèle de Sowa.

5.1.3 Relation d'ordre de TIS

Exactement comme dans le modèle de Sowa, l'application des règles de décompositions induit un ordre partiel sur les TIS. Etant donné un TIS I, un TIS J est un sous-TIS de I, si J est obtenu à partir de I par l'application des règles précédentes ou si J = I. Autrement dit, entre I et J il y a une relation « *sous TIS* », notée J << I.

La relation est réflexive car il suffit de n'appliquer aucune règle, si à partir d'un TIS I, on obtient J par l'application des règles, et si à partir de J on obtient K, alors évidemment, on peut obtenir K a partir de I par l'application de cette même séquence de règles : cette relation est donc transitive. Les trois règles produisent des arbres ayant un nombre de nœuds strictement inférieur à l'arbre de départ, ce qui assure l'antisymétrie de la relation. Cela veut dire que si I << J et J << I est alors I = J.

Deux atomes syntagmatiques différents ne peuvent pas être produits l'un à partir de l'autre car ils ont le même nombre de nœuds. Ce contre-exemple prouve que cette relation est seulement un ordre partiel.

5.1.4 Réseau des termes d'indexation syntagmatique

Etant donné un TIS, nous appliquons les règles 1 et 2 de façon récursive jusqu'à ce que la règle 3 s'applique pour obtenir des atomes. L'application de ces trois règles sur n'importe quel TIS (de taille finie) décompose tous les nœuds et feuilles du TIS en atomes en un nombre fini d'étapes.

La preuve de cette propriété est la suivante :

On fait un raisonnement par récurrence sur la hauteur de l'arbre. Un arbre de hauteur 2 qui possède n feuilles est éclaté par la règle 1 en n arbres ayant une seule racine et une seule feuille sur laquelle la règle 3 s'applique pour obtenir deux atomes. La propriété est donc vraie pour tout arbre de hauteur 2. Supposons que cette propriété soit vraie pour tous les arbres de hauteur n. Soit un arbre de hauteur n+1, après l'application éventuelle de la règle 1, nous obtenons des arbres de hauteurs n+1 dont la racine n'a qu'un seul fils et sur lequel la règle 2 peut s'appliquer. La règle 2 produit deux arbres un de hauteur 2 et un de hauteur n. Ce qui prouve récursivement cette propriété.

Le réseau est construit selon le processus de décomposition et les arcs sont orientés des sous-TIS obtenus vers le TIS d'origine. Par exemple, le réseau pour le TIS « *simulation of behavior of stockprices by computer program* » est le suivant :

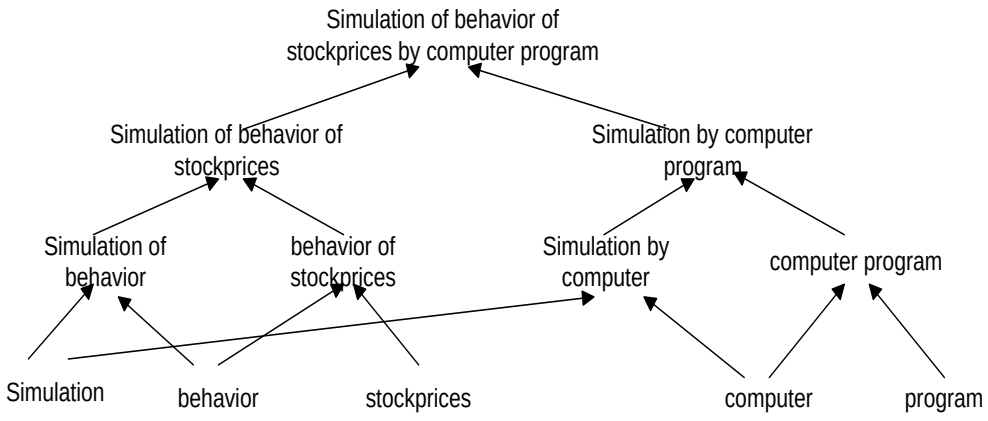


Figure 5.1-6 Un réseau de syntagmes nominaux

Nous observons que le réseau obtenu par ces règles est plus petit que le réseau obtenu par l'approche de Bruza (Figure 4.3-3). Cela vient du fait que la décomposition proposée est limitée par les trois règles.

5.2 Ensemble des termes d'indexation

5.2.1 Réseau bayésien des termes d'indexation

Dans cette partie, nous représentons le processus pour construire l'ensemble des termes d'indexation associés au document D. Cet ensemble est construit en deux phases : la première consiste à extraire des syntagmes nominaux à partir du contenu des documents et la deuxième consiste à les décomposer en des sous-syntagmes en appliquant les règles proposées. Les syntagmes extraits directement du contenu des documents sont utilisés pour représenter le contenu de document et nous supposons que les règles proposées permettent d'obtenir des sous-TIS qui gardent le thème du syntagme d'origine, donc ils sont aussi utilisés pour représenter le contenu du document.

La phase d'indexation est réalisée selon les étapes suivantes en se basant sur un analyseur syntaxique :

1. Etiquetage syntaxique de chaque mot du corpus ;

2. Extraction des syntagmes nominaux les plus grands possibles : nous utilisons pour cela des patrons syntaxiques ou des règles de transformation présentés dans le chapitre 7. Chaque syntagme nominal obtenu est un terme d'indexation syntagmatique (TIS)

3. Structuration des TIS : en fonction de l'analyseur utilisé, ces trois premières étapes peuvent être ou ne pas être confondues. C'est le cas lorsqu'on utilise les résultats d'un analyseur de dépendance. Celui-ci propose directement une structuration arborescente possible de chaque syntagme. Eventuellement, des processus supplémentaires peuvent venir compléter cette structuration. Par exemple, en cas d'ambiguïté, on peut choisir le plus fréquent.

4. Construire les réseaux correspondant aux TIS obtenus à l'étape 3, comme présenté dans la partie précédente.

Dans ce paragraphe, nous discutons plus en détail de l'ensemble des termes d'indexation que nous utiliserons. Pour chaque document D, après l'étape 2, nous obtenons l'ensemble des syntagmes nominaux extraits directement du contenu de document, désigné Extraction(D). Cet ensemble peut être utilisé comme un ensemble de termes d'indexation syntagmatiques associés au document. L'utilisation des syntagmes nominaux les plus grands comme termes d'indexation peut augmenter la mesure de précision du système mais cela va diminuer le rappel. Pour ne pas perdre le rappel, nous réalisons les étapes 3 et 4 pour structurer des syntagmes nominaux et pour les décomposer. Tous les syntagmes nominaux obtenus à la fin de d'étape 4 forment un ensemble, noté Décomposition(D). Donc, pour nous, l'ensemble des termes d'indexation associés au document D est l'union de l'ensemble Extraction(D) et de l'ensemble Décomposition(D), noté $\mathcal{X}$(D). Il faut noter que c'est une approximation car les éléments de Décomposition(D) ne se trouvent pas explicitement dans D.

Les termes d'indexation sont organisés en un réseau bayésien. Le premier niveau du réseau (les feuilles) est constitué des atomes, et les niveaux supérieurs sont des termes d'indexation complexes (TIS) qui sont composés des termes d'indexation du niveau précédent. Cette structure permet de calculer la probabilité d'un terme observé dans un document si on connaît les probabilités de ses sous-TIS.

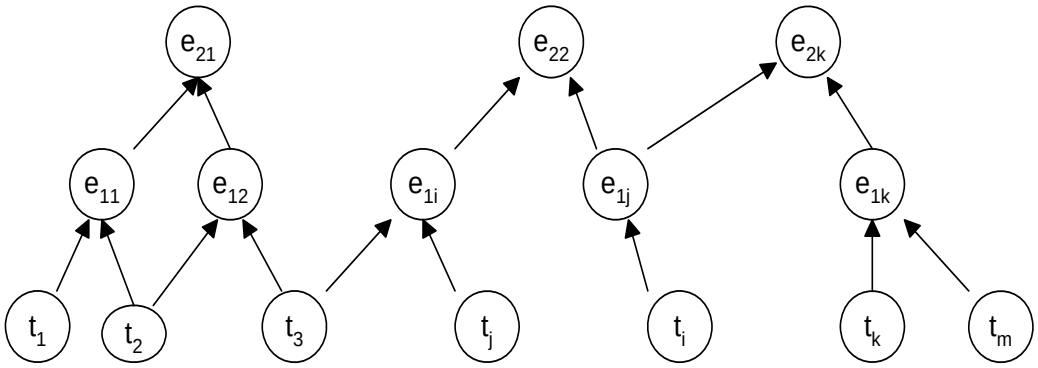


Figure 5.2-1 Réseau des termes d'indexation

Nous appliquons la même décomposition en sous-TIS à la requête. Nous obtenons aussi un ensemble de réseaux pour la requête. Les réseaux de la requête qui représentent la dépendance entre des termes de la requête, sont construits de façon similaire.

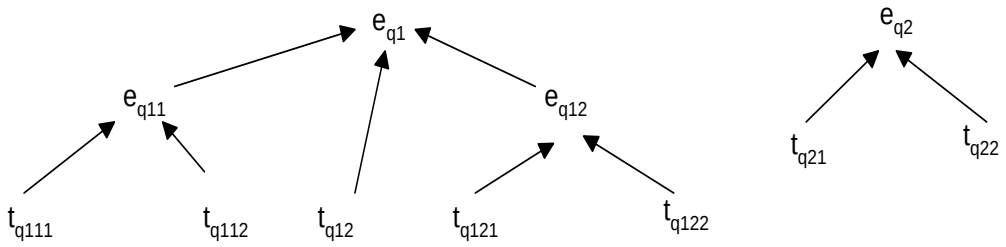


Figure 5.2-2 Réseau des termes d'une requête

Le niveau le plus haut du réseau est l'ensemble des TIS e_{qi} extraits directement de la requête.

En résumé, dans notre modèle, les termes d'indexation combinent les termes simples et les termes d'indexation syntagmatiques.

De manière similaire à la proposition de Bruza, nous proposons d'organiser l'ensemble des termes d'indexation sous la forme d'un ensemble de réseaux bayésiens. Chaque réseau représente la relation « sous-TIS » entre les termes d'indexation. Cette structure représente la relation d'un terme d'indexation avec les autres termes dans le contexte du document. Cela permet de calculer une valeur de pertinence sensible au contexte.

5.2.2 Probabilité d'un nœud dans un réseau bayésien des TIS

Nous rappelons que chaque nœud N dans le réseau associé à un événement concernant un terme d'indexation syntagmatique. Donc, il y a deux valeurs pour chaque nœud N: N = vrai (événement associé à N est réalisé) et N = faux (événement associé à N n'est pas réalisé). La probabilité de N=vrai, noté Pr(N) et la probabilité de N= faux, noté Pr(¬N) dépendent de la réalisation d'événements associés à ses pères, c'est une probabilité conditionnelle. Pour un nœud N n'ayant pas de père, la probabilité Pr(N) ou Pr(¬N) est la probabilité a priori de l'événement associé à N.

5.2.2.1 Nœud ayant N pères (probabilité conditionnelle)

Soit un nœud N ayant n pères, l'événement N observé (ou réalisé) dépend des événements associés à ses pères. Pour n pères, chaque nœud père peut prendre deux valeurs vrai ou faux. Donc, on a 2^n sous ensembles possibles de valeurs d'événement concernant ces n pères, appelé configurations de N. L'ensemble de toutes les configurations possibles de N est noté $\pi(N)$. Chaque configuration est notée $\pi_i(N)$. On observe que les $\pi_i(N)$ sont différents et incompatibles deux à deux d'un point de vue probabiliste. La probabilité que l'événement

associé au nœud N soit réalisé peut se calculer en faisant la somme de toutes les probabilités des 2^n configurations possibles de N par la formule :

$$\Pr(N) = \Pr(N \mid \pi(N)) = \sum_{i=1}^{2^n} \Pr(N \mid \pi_i(N) \times \Pr(\pi_i(N))$$

La probabilité d'une configuration $\pi_i(N)$ est obtenu par le produit des probabilités de chaque variable de la configuration car elle sont supposées indépendantes, donc :

$$\Pr(\pi_i(N)) = \prod_{Ni \in \pi_i(N)} \Pr(N_i) \times \prod_{\neg Nj \in \pi_i(N)} (1 - \Pr(N_j))$$

où $N_i \in \pi_i(N)$ signifie que la variable de N_i est à vrai dans la configuration

$\neg Ni \in \pi_i(N)$ signifie que la variable de Ni est à faux dans la configuration.

Donc, on peut développer la formule précédente comme :

$$\boxed{\Pr(N) = \sum_{i=1}^{2^n} \Pr(N \mid \pi_i(N)) \times \prod_{Ni \in \pi_i(N)} \Pr(N_i) \times \prod_{\neg Nj \in \pi_i(N)} (1 - \Pr(N_j))} \qquad (5.2.1)$$

5.2.2.2 Probabilité conditionnelle Pr(N|π_i(N))

Normalement, les probabilités conditionnelles $\Pr(N|\pi_i(N))$ sont stockées dans une matrice, appelée la matrice de liaison (comme celle présentée dans le chapitre 4). Mais quand le nombre de pères d'un nœud N est grand, le calcul et le stockage de ces matrices posent des problèmes de temps et d'espace de stockage. Nous choisissons d'utiliser des fonctions de probabilité au lieu d'utiliser la matrice de liaison.

Nous calculons la probabilité conditionnelle $\Pr(N|\pi_i(N))$ d'un nœud N en considérant ses pères et selon la règle de décomposition utilisée pour ce nœud.

1. Si un nœud N associé à un TIS I se décompose en n nœuds $N_1,\ldots,N_n$ par la règle 1. Nous supposons que les Ni ont la même mesure d'importance pour le N et que la probabilité conditionnelle de N en sachant $\pi_i(N)$ est dépendante seulement du nombre de pères ayant la valeur « vrai » (i.e. l'événement associé à ce père est réalisé) . Supposons que $\Pr(N_i)=p_i$, $\Pr(N|\pi_i(N))$ est calculée par la formule suivante :

$$\Pr(N \mid \pi_i(N)) = \frac{p_1 + \ldots + p_n}{n} \quad (5.2.2)$$

2. Si un nœud N associé à un TIS I se décompose en utilisant la règle 2 et la règle 3 en deux TIS I1, I2, dont I2 est un TIS qui garde la tête de I. Nous proposons les pondérations suivantes :

   - $\Pr(N| i_1 \wedge i_2) = \alpha$ (i.e. la probabilité d'événement N réalisé, si on a observé que I_1 est réalisé et I_2 est réalisé, est α)
   - $\Pr(N| i_1 \wedge \neg i_2) = \beta$ (i.e. la probabilité d'événement I réalisé si on a observé que I_1 est réalisé et I_2 n'est pas réalisé est β car I1 est le mot tête de I)
   - $\Pr(N|\neg i_1 \wedge i_2) = \lambda$
   - $\Pr(N|\neg i_1 \wedge \neg i_2) = 0$

$$\Pr(I|\pi(I)) = \alpha \times \Pr(i_1) \times \Pr(i_2) + \beta \times \Pr(i_1) \times \Pr(\neg i_2) + \lambda \times \Pr(\neg i_1) \times \Pr(i_2) \quad (5.2.3)$$

Ces paramètres sont déterminés par l'expérimentation avec la contrainte $\alpha > \beta > \lambda$ et $\alpha,\beta,\lambda \in [0,1]$

Par exemple : La requête q = « recherche d'information multimédia » contient un seul TIS « recherche d'information multimédia », représenté par le réseau suivant. Supposons que α = 0,8, β = 0,7 et λ = 0,5

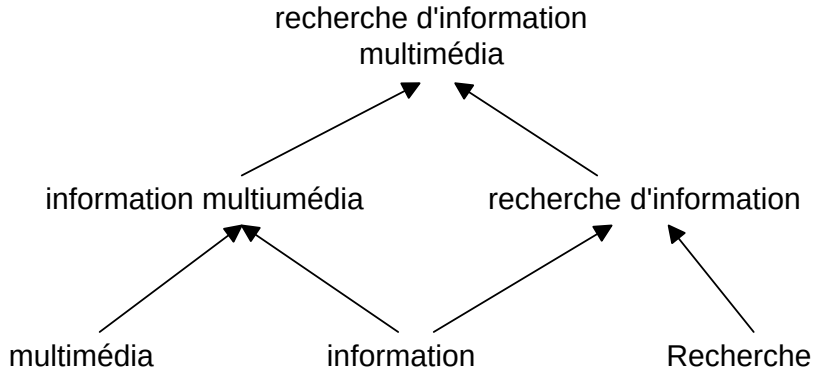


Figure 5.2-3 La requête

Le terme d'indexation « recherche d'information multimédia» se décompose en deux sous-TIS « recherche d'information » et « information multimédia » selon la règle 2. Donc les probabilités conditionnelles sont :

Pr(recherche d'information multimédia|recherche d'information ∧ information multimédia) = 0.8
Pr(recherche d'information multimédia|recherche d'information ∧¬information multimédia) = 0.7
Pr(recherche d'information multimédia|¬recherche d'information ∧information multimédia) = 0.5
Pr(recherche d'information multimédia|¬recherche d'information ∧¬information multimédia) = 0

Dans la suite, nous proposons deux solutions pour calculer la pertinence entre une requête Q et un document D. Dans la première solution, pour évaluer la pertinence, nous utilisons uniquement les réseaux de TIS de la requête. La deuxième solution plus complexe se base sur les deux réseaux.

5.3 Première solution

Pour définir la fonction de correspondance entre une requête Q et un document D, nous avons besoin des définitions suivantes :

Etant donné un TIS I, l'ensemble de tous les sous-TIS de I est désigné par $\wp(I)$.

Etant donné une requête Q, $\mathcal{X}(Q)$ est l'ensemble des termes d'indexation associés à Q. Nous définissons l'ensemble des Racines(Q) qui est l'ensemble de toutes racines de tous les réseau associés à Q.

Dans la suite, un nœud du réseau de TIS est désigné par *t* pour un atome syntagmatique et par *e* pour un TIS (au lieu de N comme désigné dans la partie précédente).

5.3.1 Mesure de pertinence

Rappelons dans le modèle logique pour la recherche d'information, un document D est pertinent pour une requête Q si D implique Q, noté D → Q. Dans notre modèle, la mesure de pertinence P(D→Q) est estimée par la probabilité de Q sachant D, Pr(Q|D). La requête Q est représentée par un ensemble de réseaux dont les TIS e_q sont les racines des réseaux ($e_q \in$ Racines(Q)). Nous supposons que ces TIS e_q sont indépendants. La probabilité Pr(D→Q) est calculée comme suit :

$$\Pr(Q \mid D) = \prod_{e_q \in Racines(Q)} \Pr(e_q \mid D) \qquad (5.3.1)$$

La pertinence entre une requête Q et un document D est le produit des probabilités conditionnelles $\Pr(e_q|D)$ des TIS racines des réseaux associés à Q.

Ces probabilités sont calculées en se basant sur les réseaux de TIS des e_q de la façon suivante. Chaque fois que nous observons un document D, nous supposons que le document est pertinent pour le TIS e_q. L'événement « le document D est pertinent pour e_q » implique que tous les termes d'indexation associés à D sont pertinents pour e_q et que les valeurs de pertinence de ces termes peuvent être estimées par la pondération de ces termes dans le document D. Ces événements changent les probabilités des termes d'indexation qui appartiennent à $\wp(e_q) \cap \mathcal{X}(D)$. Le changement de ces probabilités produit une propagation des changements de probabilité $\Pr(e|D)$ pour tous les nœuds e dans le réseau du TIS e_q.

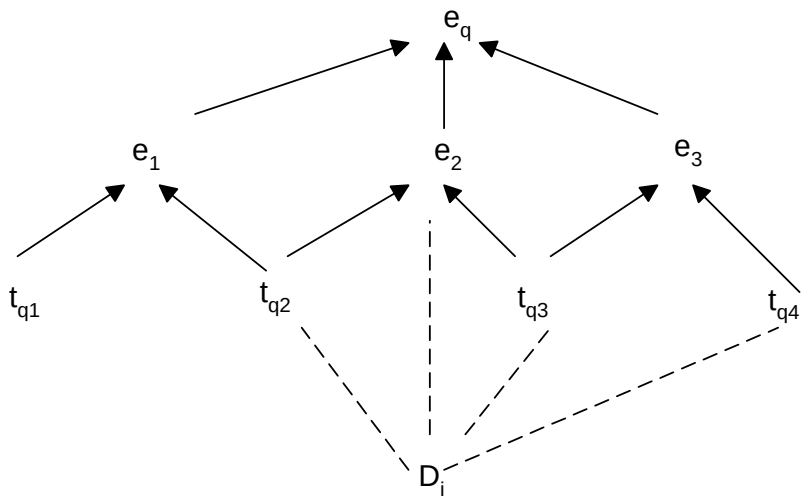


Figure 5.3-1 Réseau de la requête

Le processus de propagation est réalisé en se basant sur le calcul des probabilité $\Pr(e|D)$ des termes *e* appartenant à $\wp(e_q) \cap \mathcal{X}(D)$. Pour les nœuds feuilles qui n'appartiennent pas à $\mathcal{X}(D)$, nous supposons que la probabilité $\Pr(t|D)$ est égale à zéro. Cela signifie que D n'est pas pertinent pour *t*. Ensuite, les probabilités conditionnelles des nœuds $e_i \in e_q$ sont calculées jusqu'au sommet du réseau e_q.

Dans la suite, nous présentons un moyen d'estimer la probabilité a priori et la probabilité conditionnelle des nœuds dans un réseau d'un TIS d'une requête sachant que l'on observe un document D.

5.3.2 Probabilité des nœuds feuilles

La probabilité d'un nœud feuille *t* en sachant qu'un document D est observée et calculée de la façon suivante :

$$PP(t) = \begin{cases} w_t & \text{si t } \in \chi(\text{D}) \\ 0 & \text{sinon} \end{cases} \qquad (5.3.2)$$

où

$w_t \in [0,1]$

PP : Probabilité de Présence du terme t dans le document D.

Nous proposons deux méthodes pour estimer cette probabilité :

1. Méthode binaire : cette probabilité est égale à 1 si le terme t appartient au document D, si non à 0.

2. Méthode tf×idf : nous estimons cette probabilité par la valeur tf×idf normalisée du terme t. Cette méthode permet de distinguer le rôle du terme dans des documents différents.

5.3.3 Probabilité d'un nœud (probabilité conditionnelle)

La probabilité conditionnelle d'un nœud *e* peut être calculé par deux moyens. La première est l'utilisation la probabilité de présence du terme *e* dans le document D. La deuxième est calculée la probabilité de *e* connaissant toutes les probabilité de ses parents. Afin de pondérer le rôle d'importance d'un terme composé, nous avons choisi la valeur maximale des deux valeurs obtenues par deux calculs différents :

$$\Pr(e) = Max(PP(e), \Pr(e \mid \pi(e))) \qquad (5.3.3)$$

où

PP(e) est la probabilité de présence du terme e estimée par la formule 5.3.2.

Pr(e|π(e)) est la probabilité du nœud e connaissant tous les parents de e estimée par la formule (5.2.2) ou la formule (5.2.3).

Dans la suite, nous donnons deux exemples dans lesquels nous utilisons deux solutions pour la pondération d'un terme de la requête qui appartient à l'ensemble des termes d'indexation d'un document. Dans le premier, nous considérons la présence ou non d'un terme de la requête dans un document examiné. C'est pour cela que nous affectons 1 à tous les termes de la requête appartenant à $\mathcal{X}(D)$. Dans le deuxième, nous considérons l'importance d'un terme de la requête dans un document examiné soit la pondération tf×idf normalisée.

5.3.4 Exemple 1

Dans cet exemple, nous utilisons le cas le plus simple de la formule (5.3.2) où $w_t = Pr(t_q|D)$ vaut 1 pour tous les $t_q \in \wp(e_q) \cap \mathcal{X}(D)$ et où $Pr(t_q|D) = 0$ si $t_q \notin \mathcal{X}(D)$. Nous supposons que les paramètres α, β, λ de la formule (5.2.3) soient respectivement 0,8 ;0,7 ;0,5.

Soit une collection $\mathfrak{D} = \{D_1, D_2, D_3\}$

D_1 = {efficacité de la recherche d'information sur l'Internet}

D_2 = {information de commerce}

D_3= {information sur la recherche en biologie}

D_4= {commerce électronique}

Et une requête q = {recherche d'information multimédia}

L'ensemble des termes d'indexation de la collection $\mathfrak{D}$ est :

{efficacité, recherche, information, efficacité de la recherche, recherche d'information, efficacité de la recherche d'information, commerce, information de commerce , biologie, information sur la recherche, recherche en biologie, information sur la recherche en biologie, commerce électronique, électronique}

- calcul de $Pr(q|D_1)$

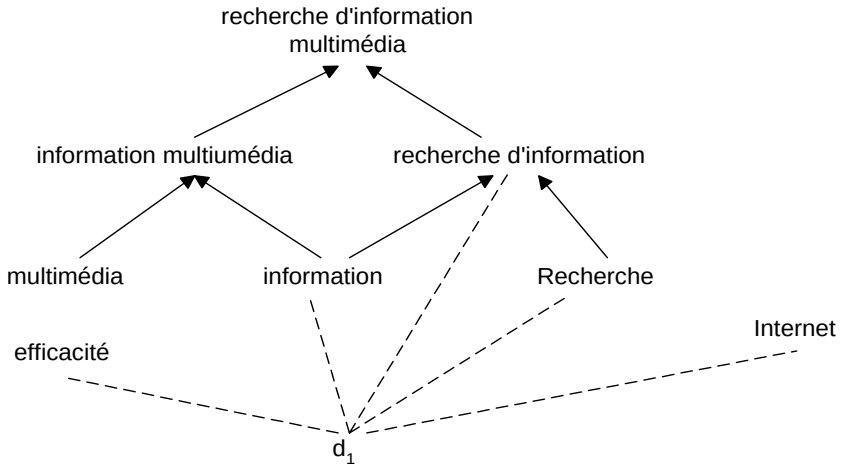


Figure 5.3-2 Réseau de la requête avec le document d1

$Pr(\text{recherche}|D_1) = 1$

$Pr(\text{information}|D_1) = 1$

$Pr(\text{recherche d'information}|D_1)=1$

$$\begin{aligned} Pr(\text{information multimédia}|D_1) &= 0{,}8 \times Pr(\text{information}|D_1) \times Pr(\text{multimédia}|D_1) + \\ & \quad 0{,}7 \times Pr(\text{information}|D_1) \times Pr(\neg\text{multimédia}|D_1) + \\ &= 0{,}5 \times Pr(\neg\text{information}|D_1) \times Pr(\text{multimédia}|D_1) \\ &= 0{,}8 \times 1 \times 0 + 0{,}7 \times 1 \times 1 + 0{,}5 \times 0{,}833 \times 0 \\ & \quad \mathbf{0{,}7} \end{aligned}$$

$$\begin{aligned} Pr(\text{recherche d'information multimédia}|D_1) = {} & 0{,}8 \times Pr(\text{recherche d'information}|D_1) \times \\ & Pr(\text{information multimédia} \mid D_1) + \\ & 0{,}7 \times Pr(\text{recherche d'information} \mid D_1) \times \\ & Pr(\neg\text{information multimédia}|D_1) + \\ & 0{,}5 \times Pr(\neg\text{recherche d'information}|D_1) \times \\ = {} & Pr(\text{information multimédia}|D_1) \\ & 0{,}8 \times 0{,}7 \times 1 + 0{,}7 \times 1 \times 0{,}3 + 0{,}5 \times 0{,}3 \times 0{,}7 \\ & \mathbf{0.875} \end{aligned}$$

d'où $Pr(q|D_1) = 0{,}875$

- calcul de $Pr(q|D_2)$

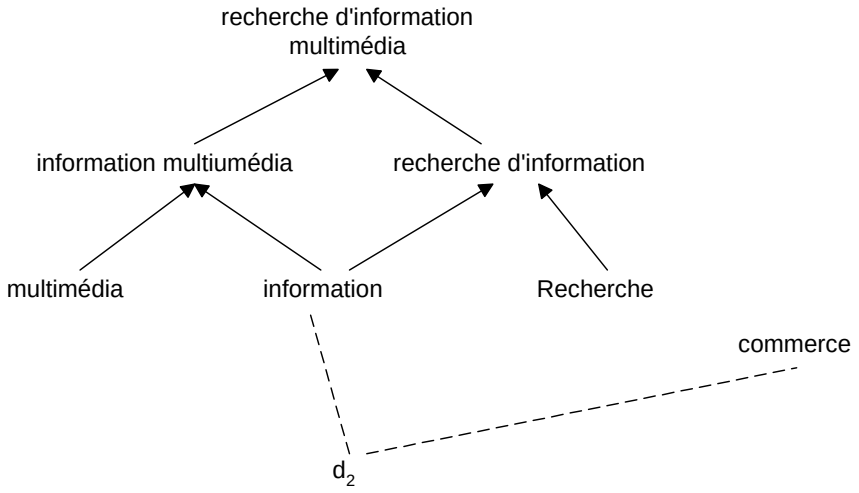


Figure 5.3-3 Réseau de la requête avec le document d2

$Pr(information|D_2) = 1$

$Pr(recherche\ d'information\ |D_2)$ = $0,8 \times Pr(recherche|D_2) \times Pr(information|D_2)+$
$0,7 \times Pr(recherche|D_2) \times Pr(\neg information|D_2)+$
$0,5 \times Pr(\neg recherche|D_2) \times Pr(information|D_2)$
$= 0,8 \times 0 \times 1+ 0,7 \times 0 \times 0 + 0,5 \times 1 \times 1$
$=$ **0,5**

$Pr(information\ multimédia|D_2)$ = $0,8 \times Pr(information|D_2) \times Pr(multimédia|D_2)+$
$0,7 \times Pr(information|D_2) \times Pr(\neg multimédia|D_2)+$
$= 0,5 \times Pr(\neg information|D_2) \times Pr(multimédia|D_2)$
$= 0,8 \times 1 \times 0 + 0,7 \times 1 \times 1 + 0,5 \times 0 \times 0$
0,7

$$\begin{aligned}
\Pr(\text{recherche d'information multimédia}|D_2) = {} & 0,8 \times \Pr(\text{recherche d'information }|D_2) \times \\
& \Pr(\text{information multimédia}|D_2) + \\
& 0,7 \times \Pr(\text{recherche d'information}|D_2) \times \\
& \Pr(\neg\text{information multimédia}|D_2) + \\
= {} & 0,5 \times \Pr(\neg\text{recherche d'information}|D_2) \times \\
& \Pr(\text{information multimédia}|D_2) \\
= {} & 0,8 \times 0,5 \times 0,7 + 0,7 \times 0,5 \times 0,3 + 0,5 \times 0,5 \times \\
& 0,7 \\
& \mathbf{0,56}
\end{aligned}$$

d'où $\Pr(q|D_2) = 0,56$

- calcul de $\Pr(q|D_3)$

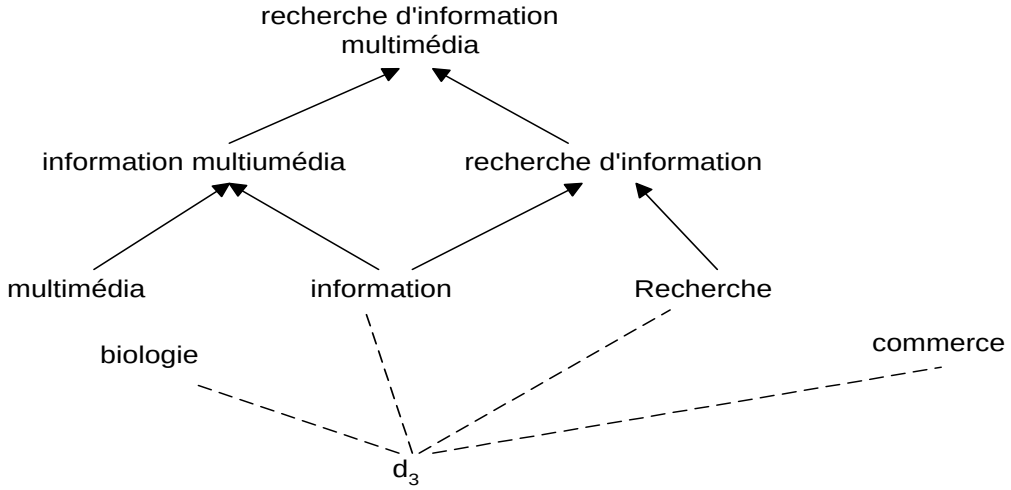


Figure 5.3-4 Réseau de la requête avec le document d3

$\Pr(\text{information}|D_3) = 1$

$\Pr(\text{recherche}|D_3) = 1$

$$\begin{aligned}
\Pr(\text{recherche d'information}|D_3) = {} & 0,8 \times \Pr(\text{recherche}|D_3) \times \Pr(\text{information}|D_3) + \\
= {} & 0,7 \times \Pr(\text{recherche}|D_3) \times \Pr(\neg\text{information}|D_3) + \\
& 0,5 \times \Pr(\neg\text{recherche}|D_3) \times \Pr(\text{information}|D_3) \\
= {} & 0,8 \times 1 \times 1 + 0,7 \times 1 \times 0 + 0,5 \times 0 \times 1 \\
& \mathbf{0,8}
\end{aligned}$$

$$\begin{aligned}\Pr(\text{information multimédia}|D_3) = {} & 0{,}8 \times \Pr(\text{information}|D_3) \times \Pr(\text{multimédia}|D_3) + \\ & 0{,}7 \times \Pr(\text{information}|D_3) \times \Pr(\neg\text{multimédia}|D_3) + \\ & 0{,}5 \times \Pr(\neg\text{information}|D_3) \times \Pr(\text{multimédia}|D_3) \\ = {} & 0{,}8 \times 1 \times 0 + 0{,}7 \times 1 \times 1 + 0{,}5 \times 0 \text{ x } 1 \\ & \mathbf{0{,}7}\end{aligned}$$

$$\begin{aligned}\Pr(\text{recherche d'information multimédia}|D_3) = {} & 0{,}8 \times \Pr(\text{recherche d'information }|D_3) \times \\ & \Pr(\text{information multimédia}|D_3) + \\ & 0{,}7 \times \Pr(\text{recherche d'information}|D_3) \times \\ & \Pr(\neg\text{information multimédia}|D_3) + \\ & 0{,}5 \times \Pr(\neg\text{recherche d'information}|D_3) \times \\ & \Pr(\text{information multimédia}|D_3) \\ = {} & 0{,}8 \times 0{,}8 \times 0{,}7 + 0{,}7 \times 0{,}8 \times 0{,}3 + 0{,}5 \times 0{,}2 \times \\ & 0{,}7 \\ & \mathbf{0{,}686} \\ = {} & \end{aligned}$$

d'où $\Pr(q|D_3) = 0{,}686$

- calcul de $\Pr(q|D_4)$

  $\Pr(q|D_4) = 0$

Constatations

1. Un document qui contient un TIS est plus pertinent qu'un document qui indexé par des termes simples contenus dans le TIS mais qui ne contient pas le TIS :

   a. Le document D_1 est le plus pertinent pour la requête parce qu'il contient le terme d'indexation « recherche d'information »

   b. Le document D_3 et la requête contiennent les termes communs « recherche » et « information » mais D_3 ne contient pas le terme composé « recherche d'information ». C'est pour cette raison que la valeur de pertinence de D_3 est inférieure à celle de D_1

   c. La valeur de pertinence de D_2 est la plus basse car D_2 ne contient qu'un terme de la requête, le terme « information ».

2. A cause de l'affectation à tous les termes qui appartiennent à $\mathcal{X}(D)$ de la valeur de 1, on ne considère pas bien l'importance de ce terme pour représenter la thématique du document. De plus, si un terme apparaît dans plusieurs documents, il aura la même probabilité (égale à 1) pour tous les documents qui le contiennent sans considération du rôle différent qu'il peut avoir dans chaque document.

5.3.5 Exemple 2

Dans cet exemple, nous supposons que la mesure de l'importance d'un terme dans un document peut être utilisée pour mesurer la pertinence entre la requête avec les documents. Cela veut dire que, si un terme *t* d'une requête q appartient à deux documents D_1, D_2, et si *t* est plus important dans D_1 que dans D_2, alors D_1 est plus pertinent pour q que D_2.

Nous utilisons la pondération tf×idf normalisée pour estimer la probabilité des nœuds du réseau qui appartiennent à l'ensemble des termes d'indexation associés aux documents D, $\mathcal{X}(D)$. L'utilisation de la pondération tf×.idf permet de pondérer l'importance d'un terme, ce qui n'est pas réalisé dans l'exemple précédent.

Rappelons la pondération tf×idf. Le facteur tf (term frequency) d'un terme t est la fréquence du terme t dans un document D. Le facteur idf (inverse document frequency) d'un terme t est défini par la formule log(N/n) où N est le nombre de documents du corpus et n est le nombre de documents où le terme t apparaît. La pondération est calculée par la formule tf×idf = tf × log(N/n).

Afin d'estimer la probabilité d'un terme, nous utilisons la forme normalisée de tf×idf :

$$ntf = \frac{tf}{\max\{tf\}}$$

$$nidf = \frac{\log(N/n)}{\log(N)}$$

Pour un terme simple de la requête t_q, $Pr(t_q|D)$ peut être calculée comme la pondération des termes t_q dans le document D par la formule tf ×idf normalisée.

Pour un terme d'indexation syntagmatique de la requête e_q, il y a deux moyens de calculer la probabilité $Pr(e_q|D)$. Le premier est d'utiliser la formule tf×idf pour e_q, le deuxième est de

calculer $Pr(e_q|\pi(e_q))$ par les formules (5.2.2) ou (5.2.3) (dans la page 78) selon la règle de décomposition utilisée. Nous estimons $Pr(e_q|D)$ la valeur maximale des deux : $Pr(e_q|D) = \max \{tf \times idf(e_q), Pr(e_q|\pi(e_q))\}$.

Nous supposons que l'intégration de la pondération d'un terme dans le calcul de la probabilité d'un nœud donnera un résultat meilleur, parce qu'il permet de nuancer l'importance d'un terme dans un document au lieu d'affecter la valeur 1 à tous les termes de $\mathcal{X}(D)$.

Nous reprenons l'exemple précédent en prenant pour tous les termes appartenant à $\mathcal{X}(D)$ la pondération tf×idf.

1. Calcul de $Pr(q|D_1)$

   $Pr(information|D1) = 1 \times \log (4/3)/\log(4) = 0,2$

   $Pr(recherche|D1) = 1 \times \log(4/2)/\log(4) = 0,5$

   $Pr(information\ multimédia|D1) = 0,8 \times 0,2 \times 0 + 0,7 \times 0,2 \times 1 + 0,5 \times 0,7 \times 0$

   $= 0,14$

   Pr(recherche d'information|D1) = max {tf×idf(recherche d'information), Pr(recherche d'information|π(recherche d'information))

   $tf \times idf = 1 \times \log(4/1)/\log(4) = 1$

   Pr(recherche d'information|π(recherche d'information)) = $0,8 \times 0,5 \times 0,2 + 0,7 \times 0.5 \times 0,8 + 0,5 \times 0,5 \times 0,2 = 0,41$

   donc, Pr(recherche d'information|D1) = 1

   Pr(recherche d'information multimédia|D1) = $0,8 \times 1 \times 0,14 + 0,7 \times 1 \times 0,86 + 0,5 \times 0 \times 0,21$

   $= 0,71$

   d'où $Pr(q|D_1) = 0,71$

2. Calcul de $Pr(q|D_2)$

Pr(information|D2) = $1 \times \log(4/3)/\log(4) = 0{,}20$

Pr(information multimédia|D2) $= 0{,}8 \times 0{,}30 \times 0 + 0{,}7 \times 0{,}3 \times 1 + 0{,}5 \times 0{,}7 \times 0$

$= 0.21$

Pr(recherche d'information|D2) $= 0{,}8 \times 0 \times 0{,}20 + 0{,}7 \times 0 \times 0{,}79 + 0{,}5 \times 1 \times 0{,}2$

$= 0{,}1$

Pr(recherche d'information multimédia|D2) $= 0{,}8 \times 0{,}1 \times 0.21 + 0{,}7 \times 0{,}1 \times 0{,}79 + 0{,}5 \times 0{,}9 \times 0\ ,1 = 0{,}12$

d'où $Pr(q|D_2) = 0{,}12$

3. Calcul de $Pr(q|D_3)$

Pr(information|D3) $= 1 \times \log(4/3)/\log(4) = 0{,}2$

Pr(recherche|D3) $= 1 \times \log(4/2)/\log(4) = 0{,}5$

Pr(information multimédia|D3) $= 0{,}8 \times 0{,}5 \times 0 + 0{,}7 \times 0{,}5 \times 1 + 0{,}5 \times 0{,}8 \times 0$

$= 0{,}35$

Pr(recherche d'information|π(recherche d'information)) $= 0{,}8 \times 0{,}5 \times 0.2 + 0{,}7 \times 0.5 \times 0{,}8 + 0{,}5 \times 0{,}5 \times 0{,}2 = 0{,}41$

Pr(recherche d'information multimédia|D3) $= 0{,}8 \times 0{,}35 \times 0{,}41 + 0\ ,7 \times 0{,}41 \times 0{,}65 + 0{,}5 \times 0{,}59 \times 0{,}41 = 0\ ,276$

d'où $Pr(q|D_3) = 0{,}42$

Constatation

Dans cet exemple, nous avons utilisé la tf×idf, mesure de l'importance d'un terme, pour estimer la probabilité que ce terme soit un bon représentant du document, Pr(t|D). Le résultat est le même que dans l'exemple 1 : le système trouve que le document D1 est le plus pertinent pour la requête q et D3 est le moins pertinent. Dans cet exemple, cela ne montre pas clairement les rôles différents d'un terme dans plusieurs documents, mais, si l'on travaille

avec un corpus réel la mesure tf×idf peut distinguer bien les rôles d'un terme dans des documents qui le contiennent. Cela a été validé dans des expérimentations présentées dans le chapitre 8.

5.4 Deuxième solution.

Dans cette deuxième solution, nous considérons les deux réseaux, le réseau de requête et le réseau de document, pour estimer la valeur de pertinence entre la requête et les documents. Nous intégrons les deux réseaux dans un seul réseau comme suit :

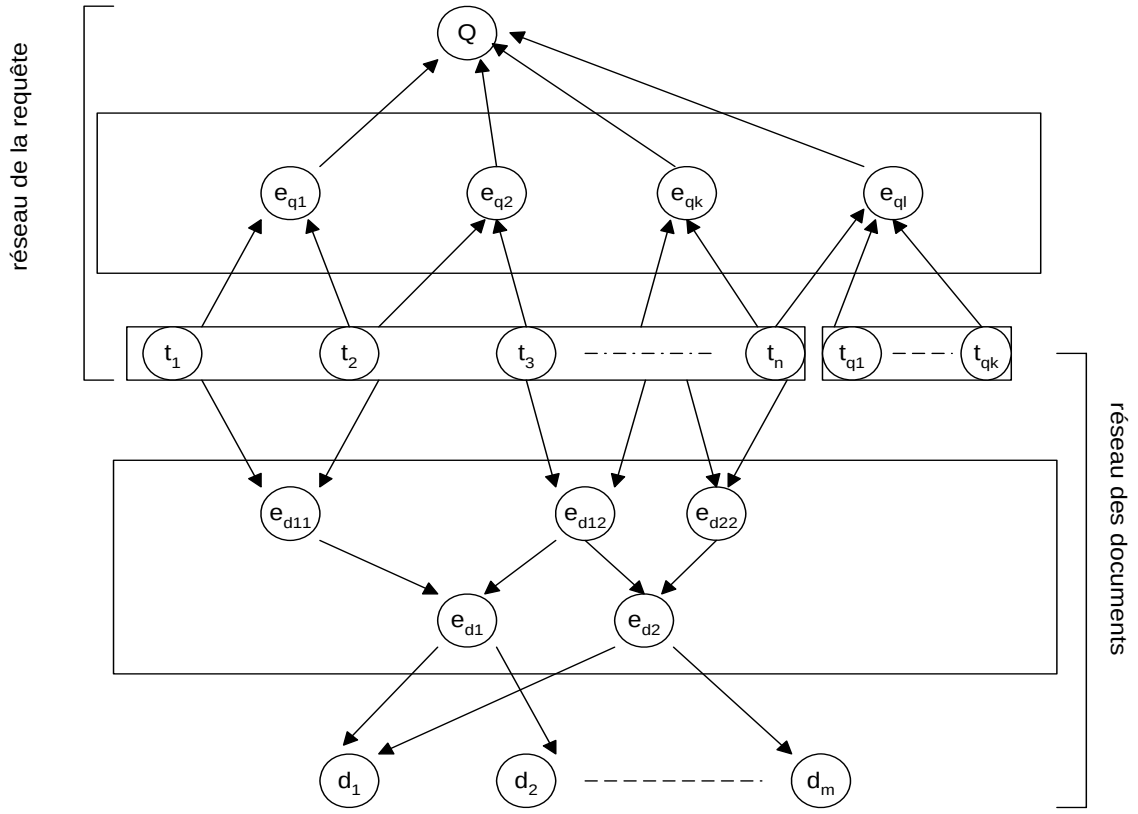


Figure 5.4-1 Réseau du système

Nous représentons la manière d'évaluer la pertinence entre la requête et les documents par des calculs dans un espace probabiliste défini ci-après.

5.4.1 Espace probabiliste

5.4.1.1 Définition univers du discours

$U=\{t_1,\ldots,t_m\}$ est l'ensemble de tous les atomes syntagmatiques (cf. 5.1.1 dans la page 67) d'un corpus. U forme l'univers du discours d'un espace probabiliste. Comme dans tous les systèmes de recherche d'information, nous supposons que les t_i sont indépendants.

Pr est une distribution de probabilités sur U. Nous supposons que U est équiprobable. Cela veut dire que les probabilité de tous les t_i sont égales et $Pr(t_i) = 1/m$ et $Pr(\neg t_i)= 1- Pr(t_i)$.

5.4.1.2 Définition d'un concept

Un concept c est une expression logique entre des ti $\in$ U. Alors, les t_i sont des concepts et les TIS e_q, e_d, la requête Q, le document D sont des concepts aussi [Ribeiro et al. 96].

On a 2^m expressions construites à partir des $t_i \in U$.

$c_1 = t_1 \wedge \neg t_2 \wedge \ldots . \wedge \neg t_m$

$c_2 = \neg t_1 \wedge t_2 \wedge \ldots \wedge \neg t_m$

$c_2{}^m = t_1 \wedge t_2 \wedge \ldots . \wedge t_m$

Chaque c_i est appelé un concept atomique.

5.4.2 Probabilité de pertinence

Dans notre modèle, la mesure de pertinence entre une requête Q avec un document D est estimée par la probabilité conditionnelle Pr(Q|D). Les concepts atomiques c_i sont « disjoints » (c_i, c_j ne sont pas réalisés en même temps pour touts j $\neq$i), la probabilité de pertinence Pr(Q|D) est calculée en se basant sur des c_i comme suit :

$$\Pr(Q \mid D) = \frac{\Pr(D,Q)}{\Pr(D)} = \frac{\sum_{c_i} \Pr(D,Q,c_i)}{\Pr(D)} = \frac{\sum_{c_i} \Pr(D,Q \mid c_i) \times \Pr(c_i)}{\Pr(D)} \qquad (5.4.1)$$

Selon le principe de l'entropy maximum, la probabilité $Pr(D,Q|c_i)$ est approximée par le produit de $Pr(D|c_i)$ et $Pr(Q|c_i)$:

$$\Pr(D,Q \mid c_i) = \Pr(D \mid c_i) \times \Pr(Q \mid c_i) \qquad (5.4.2)$$

Cela veut dire que D et Q sont indépendants d'un c_i donné. Cette approximation est cohérente avec la phase d'indexation dans laquelle le document et la requête sont indexés indépendamment.

En remplaçant l'équation (2) sur l'équation (1), on a :

$$\Pr(Q \mid D) = \frac{\sum_{ci} \Pr(D, Q, ci) \times \Pr(ci)}{\Pr(D)}$$

$$\approx \frac{\sum_{ci} \Pr(D \mid ci) \times \Pr(Q \mid ci) \times \Pr(ci)}{\Pr(D)}$$

$$\approx \sum_{ci} \Pr(ci \mid D) \times \Pr(Q \mid ci) \qquad (5.4.3)$$

Dans la partie à droite de l'équation précédente $Pr(c_i|D)$ est considérée comme la probabilité du concept atomique c_i dans une représentation du document D. Et la $Pr(Q|c_i)$ est la représentation de la requête Q dans l'espace U.

5.4.3 Conclusion

Dans cette solution, la probabilité $Pr(Q|c_i)$ peut être calculée en se basant sur la structure du réseau des termes de la requête, et la probabilité $Pr(c_i|D)$ est calculée en se basant sur le réseau des termes d'indexation du document. Dans le cas particulier où les concepts c_i sont réduits à des termes simples t_i, et où l'on suppose que les t_i sont indépendants, la formule (5.4.3) ressemble à la formule du calcul la pertinence entre les requêtes et les documents dans le modèle vectoriel traditionnel, où les probabilités $Pr(t_i|D)$ et $Pr(Q|t_i)$ sont estimées comme les pondérations du terme t_i dans le document D et la requête Q.

Chapitre 6 Recherche d'information multilingue

Dans ce chapitre, nous étendons le modèle proposé dans le chapitre précédent à un système de recherche d'information multilingue. L'idée principale de notre approche est de se baser sur la structure de termes d'indexation syntagmatique sous forme de réseau bayésien pour 'traduire' des termes d'indexation d'une requête Q dans une langue L1 en des termes d'indexation correspondants dans une langue L2.

Nous rappelons qu'une requête Q est représentée par un ensemble de réseaux de TIS, nous traduirons donc réseau par réseau. Pour chaque réseau, le processus de traduction est réalisé à partir des nœuds feuilles jusqu'à la racine du réseau.

6.1 La fonction de correspondance multilingue

Tout d'abord, nous présentons la fonction de correspondance pour la RI multilingue.

La pertinence de la requête Q dans une langue L1 (par exemple le français) pour un document D dans une langue L2 (par exemple le vietnamien), notée P(D → Q), selon J.Y. Nie [Nie 00], peut être évaluée par la formule :

$$P(D \to Q) = \sum_{Q' \in Trad(Q)} [P(D \to Q') \times P(Q' \to Q)] \qquad (6.1.1)$$

Q' représente une traduction possible de Q.

Trad(Q) = { Q' | Q' est une traduction de Q}

Dans notre approche, P(D→Q') est estimée par la probabilité conditionnelle Pr(Q'|D), et P(Q' →Q) est estimée par la probabilité conditionnelle Pr(Q|Q') la formule précédente devient :

$$\Pr(Q \mid D) = \sum_{Q' \in Trad(Q)} [\Pr(Q' \mid D) \times \Pr(Q \mid Q')] \qquad (6.1.2)$$

Dans cette formule, la probabilité Pr(Q'|D) est la mesure de pertinence entre un document D et une requête Q' dans la même langue, et la probabilité Pr(Q|Q') mesure la relation existant entre une requête Q dans la langue source et sa traduction Q' dans la langue cible. Cette

similarité sera définie à la section 6.1.2. Afin d'estimer cette formule, nous utilisons deux modèles : un modèle de la RI monolingue et un modèle de traduction de la requête.

6.1.1 Modèle de la RI monolingue

La probabilité Pr(Q'|D) est calculée par le modèle de recherche d'information monolingue qui a été présenté dans le chapitre précédent. Nous rappelons que la fonction de correspondance de ce modèle est :

$$\Pr(Q' \mid D) = \prod_{e_{q'} \in Racines(Q)} \Pr(e_{q'} \mid D) \qquad (6.1.3)$$

La probabilité $\Pr(e_{q'}|D)$ est calculée par un processus de propagation des probabilités en parcourant les nœuds du réseau à partir des nœuds feuilles jusqu'à la racine du réseau.

6.1.2 Modèle de traduction de la requête

Le problème est de savoir comment déterminer l'ensemble des traductions Q'. Dans notre approche, nous proposons un processus de construction de Q' la plus proche de Q. Une requête Q est constituée d'un ensemble de TIS $\mathcal{X}(Q)$ auquel est associé un ensemble de réseaux.

Rappelons que l'ensemble Racines(Q) est l'ensemble des racines de tous réseaux associés à Q. Pour chaque TIS $e_q \in$ Racines(Q), on a un ensemble $\wp(e_q)$ de tous les sous-TIS de e_q. La requête traduite Q' est représentée par un ensemble de TIS $e_{q'}$ traduction de e_q

Dans la partie ci-dessous, nous représentons en détail le processus de construction d'un réseau associé à un TIS e_q, appelé réseau traduit.

6.2 Processus de construction d'un réseau traduit

Afin de simplifier la phase de traduction, nous réduisons la structure de TIS en enlevant les symboles de relations Ri. Par exemple, le TIS T R [A] devient T [A]. Cela veut dire que nous gardons implicitement la relation syntaxique tête-modifieur. Cette simplification ne change pas la fonction de correspondance entre deux TIS présentée précédemment parce qu'elle ne base pas sur les relations de TIS.

Nous construisons le réseau traduit à partir du niveau les plus bas du réseau d'origine. Le processus de construction du réseau traduit consiste deux phases :

Phase 1 : Traduction des atomes syntagmatique couple par couple :

Nous traduisons chaque couple de deux termes N_1 et N_2 ayant un fils N qui est le terme composé de N_1, N_2 en utilisant un dictionnaire bilingue et un thésaurus de cooccurrence de termes dans la langue cible . Nous cherchons dans le dictionnaire les traductions possibles de N_1 et de N_2 ce qui donne les deux ensemble Trad(N_1) et Trad(N_2) . Ensuite, nous utilisons le thésaurus de cooccurrences pour choisir $N_1' \in$ Trad(N_1) et $N_2' \in$ Trad(N_2) qui sont les traductions les plus proches de N_1 et N_2 selon une mesure de similarité de traduction présentée dans la section suivante.

Phase 2 : Reconstruction des niveaux supérieurs.

Après avoir obtenu toutes les traductions des atomes syntagmatiques et, des TIS en forme $T_1[T_2]$ où T_1, T_2 sont des atomes, nous reconstruisons des niveaux supérieurs en se basant sur la règle de décomposition utilisée. Par exemple, un TIS en forme $T_1[T_2[A_2]]$ été décomposé en deux TIS $T_1[T_2]$ et $T_2[A_2]$. Après la phase 1, nous avons obtenu des traductions T_1' de T_1, T_2' de T_2, A_2' de A_2. Les nœuds du réseau traduit sont reconstruits selon le même ordre que dans le TIS initial $T_1'[T_2']$, $T_2'[A_2']$ et après $T_1'[T_2'[A_2']]$ en se basant sur la règle de décomposition utilisé pour chaque nœud correspondante du réseau origine. Il s'agit d'une hypothèse simplificatrice, en effet l'on suppose que les deux langues ont le même « mécanisme » de composition ou de décomposition pour les TIS.

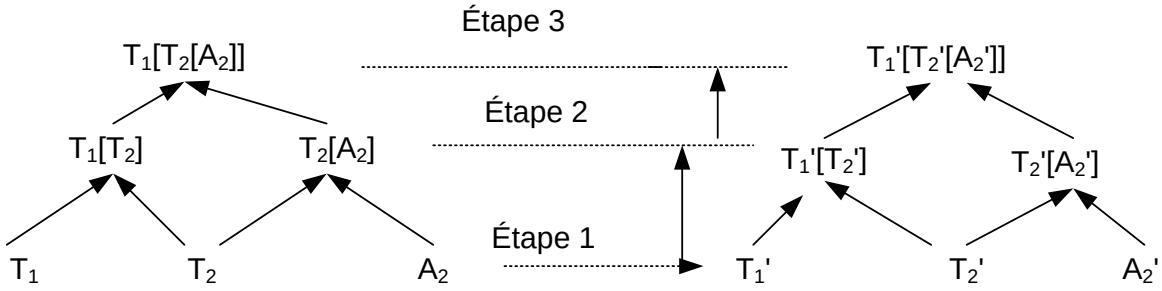


Figure 6.2-1 Reconstruction le réseau traduit

6.2.1 Traduction un couple d'atomes syntagmatiques

Supposons d'abord que N : T [A] est représenté par un réseau de la forme suivante, et ne possède qu'un seul niveau.

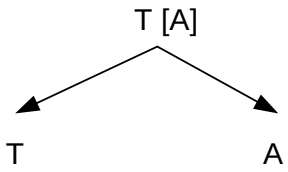


Figure 6.2-2 Un exemple d'un réseau dans langue source

Nous utilisons un dictionnaire bilingue et le corpus dans la langue cible pour choisir la traduction de N, N' : T'[A'] telle que la mesure de traduction (6.2.1) est maximum

$$\text{ScoreTradNoeud}(N|N') = \max_{T',A'} \{w_T \times \Pr(T \mid T') + w_A \times \Pr(A \mid A') + w_C \times Coocc(T', A')\} \quad (6.2.1)$$

où Pr(T|T') est la probabilité que T' est une traduction de T

Coocc(T',A') est la cooccurrences de T' et A' dans le corpus en langue cible. Cette mesure peut être calculé simplement par la formule suivante :

$$Coocc(T, A) = \frac{|T \cap A|}{|T| + |A|}$$

|T∩A| : nombre de documents contiennent T et A

|T| : nombre de document contiennent T

|A] : nombre de document contiennent A

w_T est un coefficient pour la traduction de mot tête

w_A est un coefficient pour la traduction du modifieur

w_C est un coefficient pour la mesure de co-occurrence de deux termes traduits

et $w_T+w_A+w_C = 1$

Nous utilisons les coefficients w_T,w_A,w_C pour distinguer le rôle de la traduction du mot tête, du modifieur et le facteur de désambiguïsation en utilisant la mesure de cooccurrences de deux termes traduits dans la langue cible.

La probabilité de traduction Pr(T|T') est calculée en se basant sur les ressources bilingues disponibles: un dictionnaire bilingue ou un corpus parallèle. Dans notre travail, nous utilisons un dictionnaire bilingue pour créer un lexique bilingue statistique en supposant que la probabilité de tous les termes correspondants à un terme donné dans le dictionnaire est identique. La probabilité Pr(T|T') peut être calculé par la formule suivante :

On définit les deux ensembles suivants :

Trad(T) = {T' | T' est une traduction de T}

TradInv(T) = {T' ∈ Trad(T) | T est une traduction de T'}

$$\Pr(T \mid T') = \frac{\delta(T',T)}{\sum_{T^* \in TradInv(T)} \delta(T',T)} \qquad (6.2.2)$$

$$\delta(T',T) = \begin{cases} 1 \text{ si T' est une traduction de T dans le dictionnaire} \\ 0 \text{ si non} \end{cases} \qquad (6.2.3)$$

On peut dire que Pr(T|T') représente une estimation de la probabilité d'obtenir le terme source T sachant que l'on a choisi T' comme traduction.

Si on peut choisir T',A' qui rend maximale l'expression (6.2.1), nous construisons le réseau traduit comme :

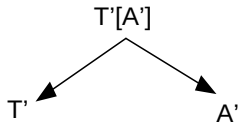


Figure 6.2-3 Le réseau traduit

6.2.2 Algorithme de traduction d'un TIS sous forme T [A]

Nous résumons sous forme d'un algorithme le processus présenté précédemment.

Entrée : un TIS sous forme e_q=T[A]

Sortie : une traduction $e_{q'}$ de e_q

1. A T correspond à un ensemble de traductions possibles Trad(T). Pour chaque traduction T'∈Trad(T) on associe une probabilité Pr(T'|T) (formule 6.2.2). Nous désignerons par TradProb(T) = {(T',Pr(T'|T))} l'ensemble des doublets (T',Pr(T'|T))

Pour tout T' ∈ Trad(T)

Pour tout A' ∈Trad(A)

i. Créer $e_{q'}$ = T' [A']

ii. Calculer ScoreTradNoeud(e_q | $e_{q'}$) selon la formule (6.2.1)

Fin de pour tout A'∈Trad(A)

Fin de pour tout T' ∈ Trad(T)

2. choisir les couples (T',A') ayant obtenu le plus grand score.
3. choisir un couple (T',A') de l'ensemble obtenu dans l'étape 2 ayant la probabilité Pr(T|T') la plus grande.

6.2.3 Reconstruction des niveaux supérieurs

Les nœuds des niveaux supérieurs du *réseau traduit* sont construits selon la structure syntaxique que celle du réseau d'origine (c'est à dire en regroupant les termes traduits selon cette structure et en se basant sur la règle de décomposition utilisée).

6.2.4 Calcul la mesure de traduction

Pour un réseau dont le TIS dans la langue source a pour racine e_q, nous construisons un réseau traduit dans langue cible qui a pour racine $e_{q'}$ .Nous pouvons dire que la mesure de traduction entre e_q et $e_{q'}$ dépend seulement des mesures de similarité entre les nœuds associés aux termes ayant une longueur inférieure ou égale à 2. En effet, lorsque nous avons trouvé la traduction de tous les termes de la requête dans la langue source, le réseau traduit est construit selon la structure syntaxique des nœuds du réseau initial de la langue source. Alors, la mesure de traduction du réseau du TIS ScoreTradReseau(e_q|$e_{q'}$) peut être calculée comme la moyenne de tous les ScoreTradNoeud(Ni|Ni') où Ni' est un nœud de $e_{q'}$ associés aux termes ayant une longueur de 2:

$$ScoreTradReseau(e_q \mid e_{q'}) = \frac{1}{m}\sum_{i=1}^{m} ScoreTradNoeud(N_i \mid N_i') \qquad (6.2.4)$$

où m est le nombre de nœuds du réseau traduit de longueur de 2. Le score de traduction d'un nœud ScoreTradNoeud($N_i|N_i$') est calculée par la formule (6.2.1)

Généralement, pour une requête Q dans la langue source est traduit par une requête Q' dans la langue cible, la mesure de traduction est calculée par la formule

$$Pr(Q|Q') = \max \{ScoreTradReseau(e_q|e_{q'}) \mid e_q \in Racines(Q), e_{q'} \text{ est la traduction de } e_q\} \quad (6.2.5)$$

Par exemple, reprenons la requête Q en français « recherche d'information multimédia » avec la structure

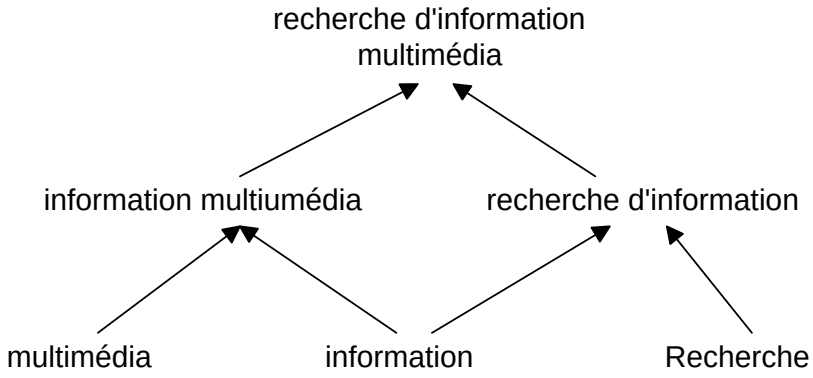


Figure 6.2-4 Réseau de la requête origine

Nous supposons qu'on peut trouver les correspondances suivantes dans un dictionnaire :

TradProb(Recherche) = {(research 0,5), (search 0,3), (retrieval 0,2) }

TradProb(Information) = {(information 1,0)}

TradProb(Multimédia) = {(mulitmedia 1,0)}

En considérant le terme composé « recherche d'information », on calcule la mesure de cooccurrences de trois couples (research, information), (search, information) et (retrieval, information). Supposons que l'on trouve :

Coocc(research, information) = 0,2

Coocc(search, information) = 0,3

Coocc(retrieval, information)=0,7

Coocc(information multimedia)=0,6.

Supposons que wt = 0,5, wA=0,1 et wc=0,4. On a alors :

Pour le TIS « recherche d'information », on a

Pr(information retrieval|recherche d'information) = 0,5× 0,2 + 0,1×1+0,4×0,7 = 0,48

Pr(information search|recherche d'information) = 0,5 × 0,3 + 0,1×1 + 0,4 × 0,3 =0,37

Pr(information research|recherche d'information) = 0,5×0,5 + 0,1×1+0,4×0,2=0,43

Alors, on choisit de traduire « recherche d'information » par « information retrieval ». Cela implique la traduction de « recherche » par « retrieval » et de « information » par « information ».

Pour le TIS « information multimédia», on a une seule traduction possible, « multimedia information », et

Pr(multimedia information »|information multimédia) = 0,5 ×1 + 0,1 × 1 + 0,3× 0,6 = 0,78

On choisit le couple (retrieval, information) comme la traduction de « recherche d'information » et le couple (multimedia, information) comme la traduction de « information multimédia ». Et jusqu'ici, on a l'ensemble {retrieval, information, multimedia} qui est formé des traductions de tous les termes simples de la requête d'origine {recherche, information, multimédia}. En se basant sur cet ensemble de traductions, on construit la traduction du TIS le plus grand « retrieval[information[multimedia]] » qui est la traduction du TIS « recherche de [information [multimedia]] ». Le réseau de la requête traduite est représenté comme suit :

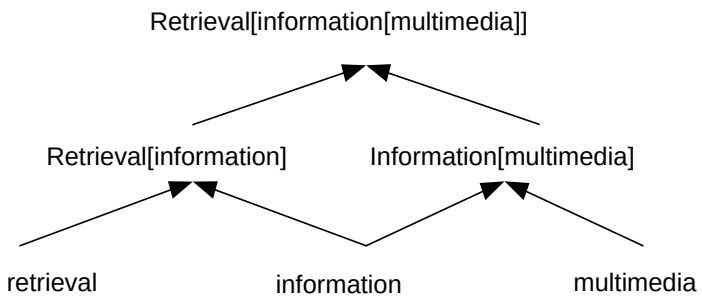


Figure 6.2-5 Réseau de la requête traduit

$$\Pr(Q' \mid Q) = \frac{1}{m}\sum_{i=1}^{m} \text{ScoreTradReseau}(e_i \mid e_i') = \frac{0,2+1+1+0,48+0,78}{5} = 0,692$$

6.3 Conclusion

Nous avons proposé une méthode de recherche d'information multilingue se composant de deux modèles : un modèle de recherche d'information monolingue basé sur des termes d'indexation syntagmatiques et un modèle de traduction de requêtes. Le modèle de recherche d'information monolingue est présenté en détail dans le chapitre précédent. Le modèle de traduction de requêtes permet de traduire des termes composés en considérant le rôle et l'importance du mot Tête et la mesure de cooccurrence des termes traduits. Ces deux points sont originaux par rapport aux autres méthodes de traduction de la requête. Nous avons également utilisé des coefficients pour examiner le rôle du mot tête dans la phase de traduction, afin de ne pas obtenir trop de « dérives » dans la traduction.

Partie IV : Expérimentation

Chapitre 7 Recherche d'information en vietnamien

7.1 Spécificités du vietnamien

Le vietnamien est une langue tonale et isolante. Une **langue à tons** (ou « langue tonale ») est une langue dans laquelle un changement de *tonalité* (« hauteur ») dans la prononciation d'un ou plusieurs sons d'un mot entraîne un changement de *sens* de ce mot. Une **langue isolante** est une langue dans laquelle les mots sont ou ont tendance à être invariables.

Le vietnamien utilise l'alphabet latin en ajoutant des accents sur les voyelles pour créer de nouvelles voyelles comme Ă, Â, Ê, Ô, Ơ, Ư. De plus, le vietnamien a six tons différents notés : ‘, ` , ?, ~ , . sur des voyelles. Ces accents toniques modifient le sens des mots, par exemple : ma (fantôme), má (joue), mà (dans le mot « mà còn » : encore), mả (tombe), mã (apparence), mạ (semis).

La grammaire vietnamienne est assez simple. Les structures se présentent généralement sous la forme : Sujet + Verbe + Complément, par exemple :

Tôi / đi / học
(Je/ vais /à l'école)
Sujet/ verbe /complément

Du fait du caractère isolant, il n'y a pas de conjugaison ni de déclinaison, pas de pluriels irréguliers, etc. Dans cette langue, les mots ne changent jamais, c'est à dire que chaque mot n'a toujours qu'une forme.

Dans ce chapitre, nous représentons les spécificités du vietnamien du point de vue de traitement linguistique pour la recherche d'information.

7.1.1 Mot vietnamien

Une phrase en vietnamien se compose de mots, eux mêmes composés de ce que nous appellerons *unités linguistiques* séparées par un espace. Chaque unité est une chaîne de caractères qui peut ne pas avoir de signification propre. Pour faciliter la présentation, nous utilisons les caractères "[ ]" pour désigner les mots composés d'au moins deux unités

linguistiques. Par exemple, dans le mot [khủng hoảng] (la crise), la première unité n'a pas de sens quand elle est utilisée seule, et la deuxième signifie « affolé ». C'est pour cette raison que la segmentation d'un texte en mots est plus difficile que pour les langues européennes. C'est une difficulté pour le traitement automatique du vietnamien en général, et pour la recherche d'information (RI) en particulier.

Les mots vietnamiens sont morphologiquement invariables, il n'y a donc pas besoin de phase de lemmatisation dans l'indexation. Il existe pourtant des suffixes et préfixes en vietnamiens, mais ils ne sont pas couramment utilisés. Par exemple, le préfixe « sự » transforme un verbe en substantif : [lựa chọn] (choisir) et [sự [lựa chọn]] (choix), tandis que le suffixe « hóa » réalise l'opération inverse : [tin học] (informatique) et [[tin học] hóa] (informatiser).

En recherche d'information, si l'on veut indexer les documents vietnamiens par un « sac de mots », on doit faire face au problème de la segmentation du texte en mots significatifs. Les tâches comme la lemmatisation, l'extraction des racines (racinisation) ne sont pas efficaces pour le vietnamien. Nous posons comme hypothèse que l'utilisation des mots composés et des syntagmes nominaux est plus efficace pour l'indexation de documents vietnamiens.

Environ 80 % des mots vietnamiens sont des mots comportant deux unités linguistiques [Nguyen Quynh 2001]. Partant de ce constat, nous avons réalisé une première expérimentation qui permet de mesurer l'importance du choix du terme d'indexation. En effet, le choix classique pour les langues européennes consiste à prendre pour termes d'indexation les unités linguistiques séparées par des espaces. Ce choix est correct pour ces langues où l'unité linguistique correspond la plupart du temps à un mot. Par exemple, en français, les mots composés non liés par un tiret (comme "pomme de terre"), sont assez peu courants, et le fait de ne pas les reconnaître à l'indexation a peu d'impact sur la qualité du SRI. Par contre, en vietnamien, sachant que la majorité des mots sont composés d'au moins deux unités, nous pouvons nous attendre à une influence sur la qualité des réponses du SRI.

Dans l'expérimentation présentée dans la dernière section de ce chapitre, nous avons comparé une indexation à base d'unités linguistiques avec une indexation basée sur des couples d'unités, que nous appelons « bigrammes »

7.1.2 Catégories de mots

Le vocabulaire vietnamien est classé en catégories classiques : substantif, verbe, adjectif, adverbe, préposition …[Nguyen Than 97] La catégorie d'un mot ne peut pas être reconnue grâce à des marqueurs morphologiques comme dans les langues possédant des variations morphologiques. Donc, nous ne pouvons déterminer la catégorie d'un mot que par le contexte où ce mot apparaît. Par exemple :

thành công (nom : réussite) của dự án đã tạo tiếng vang lớn

(*La réussite du projet crée un grand écho*.)

Bạn đã ***thành công*** (verbe : réussir) trong nghiên cứu khoa học

(*Vous avez réussi dans la recherche scientifique*.)

Buổi diễn rất ***thành công*** (adjectif : réussi).

(*Le concert est réussi*.)

Le mot « thành công » dans la première phrase est un substantif, dans la deuxième c'est un verbe et il est adjectif dans la troisième.

7.1.3 Spécificités de syntagme nominal vietnamien

La structure d'un syntagme nominal vietnamien est un problème encore discuté par les linguistes vietnamiens. Nous avons donc adopté un point de vue raisonnable qui est le suivant.

Un syntagme nominal vietnamien possède trois parties. Une partie principale considérée comme la tête du syntagme, une partie qui précède la tête et une partie qui suit la tête. La partie "tête du syntagme" est un substantif simple ou un substantif composé. La partie précédant la tête contient des déterminants et la partie suivant la tête contient des modifieurs.

Par exemple :

Partie précédente	Tête	Partie suivante	
tất cả các cuốn	sách (livre)	tin học	Tous les livres d'informatique
L3oại	khoai tây (pomme de terre)	vừa mua	La pomme de terre achetée
	tin học (informatique)	công nghiệp	Informatique industrielle

Tableau 2 Exemples de syntagmes nominaux vietnamiens

La partie qui suit la tête est une partie complexe. Elle peut contenir des mots de catégories différentes comme substantif, adjectif, verbe, pronom, nombre et même un syntagme complet. Par exemple :

Sách thiếu nhi (SUBC SUBC) = livre d'enfant

Sách cũ (SUBC ADJ) = ancien livre

Phòng đọc (SUBC VERB) = salle de lecture

Xe mười lăm (SUBS NUM) = véhicule numéro quinze

Nhà chúng tôi (SUBS PROM) = notre maison

Sách báo thư viện đặt mua (SUBC [SUBC VERB]) = les livres et journaux réservés par la bibliothèque.

Actuellement, il n'existe pas encore d'étude complète du syntagme nominal en vietnamien. Il n'existe donc pas de structure prédéfinie d'un syntagme nominal basée sur les catégories grammaticales comme on peut en trouver dans les langues européennes. La méthode classique d'automate d'états finis utilisée dans ces langues n'est donc pas applicable. D'autre part, nous n'avons pas de corpus d'apprentissage assez grand pour utiliser la méthode « memory-based » [Sang 2002]. C'est pour ces raisons que nous avons choisi la méthode par règles de transformation présentée ci-dessous pour l'extraction des syntagmes.

7.2 Méthode d'apprentissage de règles de transformation

La méthode d'apprentissage de règles de transformation, proposée par Brill en 1995, est une des meilleures méthodes d'apprentissage automatique de règles syntaxiques (Florian et al, 2001). Elle est souple et puissante pour les tâches de traitement automatique de la langue comme l'étiquetage (morphosyntaxique) [Brill, 1995], l'analyseur de surface [Brill, 1996], et l'extraction de syntagmes [Ramshaw et al. 95]

L'idée principale de cette méthode est de considérer que le traitement des tâches de segmentation, d'étiquetage, et d'extraction de syntagmes, sont des tâches de classification. Par exemple, la segmentation des mots peut être vue comme une tâche de classification des unités qui le composent selon trois classes : D le début d'un mot, M le milieu d'un mot et F la fin d'un mot. L'étiquetage est une classification des mots par un ensemble de catégories et l'extraction de syntagmes est considérée comme une segmentation en mots de taille plus grande.

Cette méthode comporte deux phases : la phase d'apprentissage et la phase d'application. Dans la phase d'apprentissage, un corpus d'apprentissage doit être construit manuellement pour permettre d'apprendre automatiquement des règles de transformation. Ces règles extraites sont ensuite utilisées pour traiter un nouveau corpus.

La production des règles se fait par rapport à un ensemble de modèles. Chaque modèle fixe le contenu d'une règle à l'aide de méta symboles indiquant que l'on se réfère au mot (word), à sa classe (chunk) ou à sa catégorie grammaticale (pos pour *part of speech*). Ces métas symboles sont associés à une position relative à la position courante d'application de la règle. Une règle, lorsqu'elle est appliquée, change la classe du mot courant. Par exemple, la règle :

word-1="la" word_0="voiture" pos_-1=ART pos_0=VERB => pos_0=SUB

exprime que, si on trouve le mot *voiture* catégorisé comme un verbe, précédé par le mot *la* catégorisé comme un article, alors on peut changer la catégorie du mot courant *voiture* en substantif. La classification concerne ici l'attribution d'une catégorie grammaticale. Cette règle peut correspondre au modèle suivant :

word_-1 word_0 pos_-1 pos_0 => pos

Ces modèles peuvent être classés en quatre types : les modèles concernant le mot, ceux concernant le contexte du mot, ceux concernant de catégorie du mot, et ceux concernant le contexte de catégorie d'un mot. Ces types de modèles vont être présentés plus en détail dans les sections suivantes. On utilise une fonction d'évaluation pour valider l'intérêt d'une règle. Cette phase est illustrée par la figure suivante :

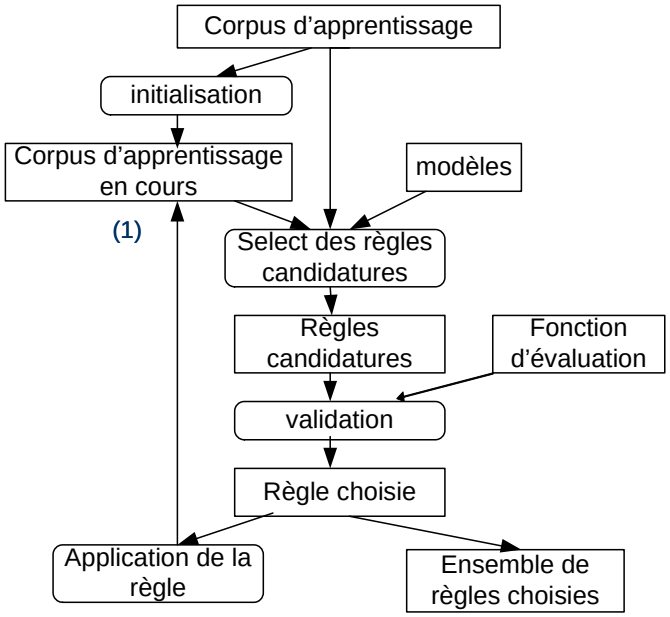


Figure 7.2-1 La phase d'apprentissage

De manière générale, un corpus d'apprentissage est un ensemble de mots associés à une classe qui dépend du type d'apprentissage en cours. Nous appelons *classe correcte* la classe qui a été affectée manuellement à chaque mot du corpus. Nous appelons *classe inférée* la classe qui est produite par l'application des règles. Dans notre cas, la classe peut être une catégorie grammaticale (partie du discours), ou une classe de construction des mots (début, milieu, fin de mot). Les règles sont apprises sur les exemples puis appliquées à l'ensemble du corpus d'apprentissage de manière à minimiser les erreurs. Le processus s'arrête lorsqu'aucune amélioration ne peut être apportée à la production des classes.

Au cours de l'apprentissage, la classe inférée est comparée à la classe correcte du corpus pour évaluer l'intérêt d'une règle. Pour cela, une fonction *Score* évalue l'impact d'une règle sur le corpus d'apprentissage. Ce score est calculé sur la différence, après application de cette règle, entre le nombre de bonnes corrections de classes et le nombre de mauvaises modifications de classes par rapport à la classe correcte du corpus d'apprentissage.

Etant donnée une règle r, un mot m, x(m) est la classe courante associée à m, x(r, m) est la classe associée au mot m après avoir appliqué la règle r.

$$Score(r) = \sum good(r) - \sum bad(r)$$

où

$$good(r) = \begin{cases} 1 \text{ si } (x(m) \text{ est incorrect}) \wedge (x(r, m) \text{ est correct}) \\ 0 \text{ sinon} \end{cases}$$

$$bad(r) = \begin{cases} 1 \text{ si } (x(m) \text{ est correct} \wedge x(r,m) \text{ est incorrect}) \\ 0 \text{ sinon} \end{cases}$$

L'algorithme de la phase d'apprentissage est le suivant :

1 Création d'un état initial des classes du corpus d'apprentissage

2 Parcours de l'ensemble des modèles

2.1 Pour un modèle donné, parcours du corpus unité par unité

2.1.1 Production d'une nouvelle règle

2.1.2 Application de cette règle à la copie courante du corpus

2.1.3 Calcul du score de cette règle, basé sur la fonction d'évaluation

2.1.4 On ne garde cette règle que si son score est positif

2.2 Pour un modèle, choisir la règle ayant le score le plus élevé, et l'ajouter à l'ensemble des règles apprises

2.3 Appliquer la règle choisie à la copie courante du corpus

3 Si les modèles ne produisent plus de nouvelles règles, arrêter la phase d'apprentissage, sinon répéter tout le processus depuis l'étape 2

L'étape initiale 1 consiste à affecter à chaque mot la classe la plus probable statistiquement. Cela signifie que pour chaque mot du corpus, on comptabilise les classes possibles à partir des classes correctes, puis on initialise la classe inférée à la classe la plus fréquente. A l'étape 2.1.1, la production d'une règle consiste à instancier le modèle pour extraire du corpus à la position donnée, les éléments (mots, classe ou pos) en partie gauche de la règle. Dans la partie gauche, la classe correspond à la classe inférée dans l'état actuel du corpus. La partie droite correspond à la classe correcte. Pour un modèle et une position dans le corpus, il n'y a donc qu'une seule règle produite. On peut noter que cet algorithme est très coûteux, car il parcourt tout le corpus pour produire une seule règle. Il y a donc autant de parcours de corpus que de règles apprises.

Dans la suite, nous présentons plus en détail des modèles et des règles utilisées dans cette méthode. Un modèle est représenté sous la forme suivante :

<trigger$_1$>…<trigger$_n$> <classe-courante$_1$>…<classe-courante$_n$>

=><classe-correcte$_1$>…<classe-correcte $_n$>

où

<trigger$_i$> est soit une unité linguistique soit une catégorie grammaticale

<classe-courante$_i$> est la classe associée au trigger i à un instant donné

<classe-correcte $_i$>est la classe correcte du trigger i .

Par exemple, on traite la tâche d'extraction de syntagmes nominaux comme la classification d'un mot en une de trois classes B (begin, début d'un syntagme), I (in, milieu d'un syntagme) et O (out, dehors d'un syntagme). Les modèles dans ce cas peuvent être de la forme suivante :

word_0 chunk_0 => chunk (1)

word_-1 word_0 chunk_-1 chunk_0 => chunk (2)

pos_0 chunk_0 => chunk (3)

pos_-1 pos_0 pos_1 chunk_-1 chunk_0 chunk_1 => chunk (4)

Dans un modèle de règle, *word* représente une unité linguistique, *chunk* est la classe, et *pos* est la catégorie grammaticale (partie du discours, part of speech). Le chiffre qui suit représente la position de l'élément dans le corpus par rapport à la position courante de construction d'une nouvelle règle.

Dans le premier exemple, 'word_0' est le mot examiné, 'chunk_0' désigne la classe courante du mot examiné, et 'chunk' est la classe correcte pour le mot examiné.

Dans le deuxième exemple, 'word_-1' est le mot précédant le mot examiné, ' word_0' est le mot examiné, 'chunk_-1' est classe courante du mot précédent, 'chunk_0' est la classe courante du mot examiné et 'chunk' est la classe correcte pour le mot examiné. En d'autres termes, ce modèle concerne des règles de contexte d'un mot.

Le troisième exemple est un modèle de règles concernant la catégorie grammaticale du mot examiné. Le dernier concerne les règles de contexte pour les catégories des mots.

Si l'on utilise les modèles de l'exemple ci-dessus, on peut trouver des règles suivantes :

word_0 = « le » chunk_0 = 'O' => chunk = 'I' (1')

pos_-1="PRO" pos_0 ="SUBC" pos_1="ADJ" chunk_0 = 'O' => chunk ='I' (2')

La règle (1') signifie que, si on trouve le mot « le » étiqueté 'O' (en dehors d'un syntagme) alors, il faut changer sa classe en 'I' (dans le contexte d'un syntagme)

La règle (2') signifie que, si la catégorie du mot précédent est un pronom, si la catégorie du mot examiné est un substantif, si la catégorie du mot suivant est un adjectif, et, si le mot examiné est étiqueté 'O', alors, il faut se changer sa classe en 'I'

Dans la phase d'application, on ne parcourt qu'une seule fois le corpus et on applique les règles extraites de la phase d'apprentissage pour modifier l'état du corpus. Les règles sont appliquées dans l'ordre décroissant de leurs scores (cf. figure 7.2-2).

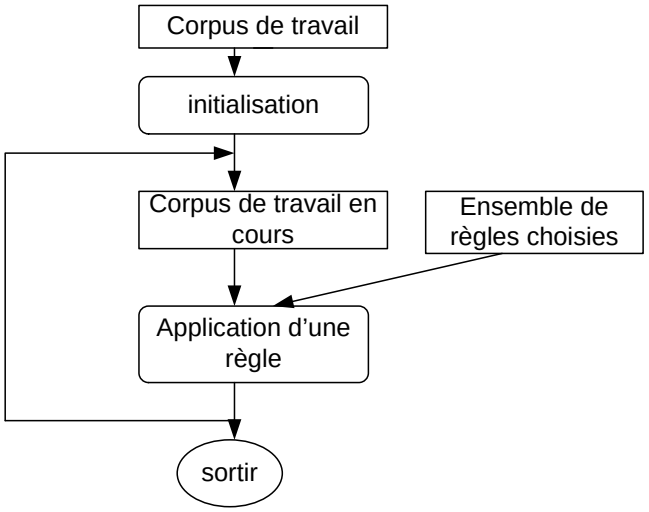


Figure 7.2-2 Phase d'application des règles d'étiquetage

Dans des parties suivantes, nous présentons les expérimentations que nous avons réalisées pour le vietnamien.

7.3 Analyseur syntaxique du vietnamien

Nous avons construit un analyseur du vietnamien qui permet l'étiquetage des mots selon leur catégorie grammaticale (parties du discours) et ensuite permet l'extraction des syntagmes nominaux. Nous avons utilisé la méthode d'apprentissage des règles de transformation présentée précédemment pour construire cet analyseur syntaxique. Nous avons choisi cette

méthode parce qu'elle permet d'apprendre des règles d'étiquetage ou des règles d'extraction de syntagme selon le contexte du mot dans le texte. Cela permet, en l'absence de marqueurs morphologiques en vietnamien, de reconnaître les catégories des mots. Afin de réaliser la phase d'apprentissage, nous avons construit manuellement un corpus d'apprentissage.

7.3.1 Corpus d'apprentissage

Nous avons catégorisé un corpus de journaux de 37 000 mots et extrait les syntagmes nominaux manuellement, avec l'aide de linguistes [Lai 2002]. Nous avons ensuite décomposé ce corpus en deux parties : la première partie contient 90% du corpus et est considérée comme le corpus d'apprentissage; la deuxième partie (10% du corpus) est le corpus de test.

L'ensemble des catégories de mots vietnamiens que nous avons utilisé pour cette expérimentation est le suivant :

SUBC : Substantif commun
SUBP : Substantif propre
VERB : Verbe
ADJ : Adjectif
ADV : Adverbe
PRO : Pronom
AUX : Auxiliaire
CONJ : Conjonction
INJ : Interjection
NUM : Nombre

7.3.2 Initialisation

La première tâche est la création de l'état initial du corpus d'apprentissage. Cette tâche est sophistiquée : si les *classes inférées initiales (les classes affectées la première fois pour des mots du corpus)* sont la plupart correctes, alors, on n'apprendra pas plusieurs règles. Le temps d'apprentissage peut être trop long. La création de l'état initial des classes inférées du corpus d'apprentissage est normalement calculée par la statistique de répartition des classes comme décrites précédemment. Pour diminuer le temps d'apprentissage, nous avons initialisé l'ensemble des règles apprises, par un ensemble de règles construites manuellement comme suit :

pos_1= « PRO » word_1= « này »=>chunk= I

(si le catégorie du mot suivant du mot examiné est un pronom et si le mot suivant est « này », alors le mot examiné est classé I (au milieu d'un syntagme)

pos_1="PRO" word_1="nào" => chunk="I"

pos_1="PRO" word_1="kia" => chunk="I"

pos_1="PRO" word_1="ấy" =>chunk="I"

pos_-1="ADV" word_-1="cái" =>chunk="I"

pos_0=ADV word_0="cái" =>chunk="I"

chunk_-1= «I » word_0= « đó » =>chunk= « I »

(si le mot précédent est au milieu d'un syntagme (I) et si le mot examiné est « đó », alors le mot examiné est aussi au milieu d'un syntagme (I))

chunk_-1= « I » word_0= « này » =>chunk= « I"

chunk_-1= « I » word_0= « nào » =>chunk= « I"

chunk_-1="I" word_0="kia" =>chunk="I"

chunk_-1="I" word_0="ấy" =>chunk="I"

word_1= « của » =>chunk= « I »

(si le mot suivant est 'của', alors le mot examiné est au milieu d'un syntagme)

word_-1="sự" =>chunk="I"

word_-1="cuộc"=>chunk="I"

word_-1="những"=>chunk="I"

word_-1=”các”=>chunk=”I”

chunk_-1=”I” pos_0=”PRO” word_1=”,”=>chunk=”I”

(si le mot précédent est au milieu d’un syntagme, si la catégorie du mot examiné est un pronom et si le mot suivant du mot examiné est un « , », alors le mot examiné est au milieu d’un syntagme)

chunk_1=”O” pos_-1=”VERB” pos_0=”NUM”=>chunk=”O”

chunk_-2=”I” chunk_-1=”I” pos_0=”ADJ” word_1=”,”=>chunk=”I”

chunk_-2=”I” chunk_-1=”I” pos_0=”ADJ” pos_1=”PREP”=>chunk=”I”

Ces règles on été appliquées une première fois sur le corpus. Dans cette phase, si nous utilisons seulement l’information statistique, les erreurs représentent 8,7% de la taille du corpus. Lorsque nous combinons la statistique avec les règles prédéfinies, les erreurs tombent à 6,8% de la taille du corpus. Le format du corpus d’apprentissage est le suivant : chaque ligne est composée d'un mot, de la catégorie de ce mot, et de sa classe (B,I,O).

Par exemple :

Nguyên nhân (la cause)	SUBC	I
là (est)	VERB	O

7.3.3 Modèle de règle

Nous avons adapté au vietnamien l’ensemble des modèles proposés par Brill [Brill 96]. En particulier, nous avons supprimé les modèles qui concernent un contexte trop large (≥ 3 mots précédents ou ≥ 3 mots suivants) pour diminuer le temps de la phase d’apprentissage. De plus, nous avons ajouté des modèles qui examinent le mot et la catégorie du mot en même temps, par exemple :

pos_0 word_1 => chunk

7.3.4 Résultat

Cet outil a été programmé en Delphi sur Windows par un étudiante de Mastère [Lai 2002]. Le test donne une précision de 70% et un temps d'apprentissage de 10 heures 35 minutes. Par exemple, les 2 règles apprises ayant les scores les plus élevés sont :

1. Good :524 Bad :101 chunk_0= O pos_-1=SUB pos_0=ADJ => chunk= I (si le mot examiné est classé O, si la catégorie du mot précédent est un substantif, et si la catégorie du mot examiné est un adjectif, alors on change la classe du mot examiné en I)

2. Good: 80 Bad:0 chunk_0=O chunk_1=I pos_1=ADJ => chunk= I (si le mot examine est classé O, et si le mot suivant est classé I, et si sa catégorie est ADJ, alors on change la classe du mot examiné en I, c'est-à-dire qu'il fait partie d'un syntagme)

Nous constatons que ces deux règles apprises concernent le cas des syntagmes de la forme : SUB ADJ. C'est une forme connue de syntagme.

Nous pensons que la précision peut être améliorée en augmentant la variété des catégories grammaticales et la taille du corpus d'apprentissage.

Grâce à cet outil, nous avons pu expérimenter l'impact de l'indexation des documents vietnamiens par les syntagmes nominaux.

7.4 Expérimentations de la RI en vietnamien

7.4.1 Collection test

Nous avons construit une collection test pour pouvoir réaliser des mesures sur la qualité de l'indexation. Une collection test en RI est constituée d'un ensemble de documents, d'un ensemble de requêtes, et d'un ensemble de résultats, c'est à dire de l'ensemble des documents jugés manuellement pertinents pour les requêtes. La collection test est constituée d'articles en vietnamien sélectionnés à partir de journaux (La jeunesse) de l'an 2000 qui proviennent de l'Institut Linguistique Vietnamien de Hanoi. Cette collection contient 10 750 articles et a une taille de 23 Mo, elle utilise le code des caractères TCVN3 utilisé dans la plupart des documents électroniques vietnamiens. A partir de ce corpus de documents, nous avons proposé 14 requêtes et nous avons évalué manuellement la pertinence avec les documents du corpus. Comme il est difficile de parcourir manuellement toute la collection, nous avons

réalisé une première indexation des unités linguistiques avec SMART avec la pondération ltc. Pour toutes les requêtes, la liste des 20 meilleurs documents a été examinée manuellement pour construire la liste des réponses pertinentes.

Cette collection test est un outil de mesure précieux et unique pour la langue vietnamienne. Elle nous a permis de mettre en place les expérimentations que nous décrivons ci-après.

7.4.2 Méthodes

Nous avons réalisé cette expérimentation selon trois méthodes d'indexation : unigrammes, bigrammes et bigram combiné avec un lexique. La notion d'unigramme correspond aux unités linguistiques. Comme nous l'avons décrit précédemment, ces unités ne se placent pas au niveau du mot de la langue. C'est pour cette raison que nous avons testé une méthode de regroupement des unités en couples. Nous avons finalement utilisé une ressource linguistique (un lexique) d'environ 30 000 items pour réaliser l'indexation. C'est le système SMART qui a été utilisé pour les tests, avec une mesure de pondération des termes par la méthode *lt*c

7.4.2.1 Expérimentation avec des unigrammes

Dans cette première expérimentation, nous avons indexé la collection test en utilisant les unités linguistiques, les unigrammes comme termes d'indexation. Le résultat est faible, mais il nous servira de base pour mesurer l'amélioration des résultats obtenus par les autres méthodes. La précision moyenne est **0.3636**. Comme la plupart des mots vietnamiens sont des mots ayant deux unités, nous pensons qu'il est facile d'améliorer cette mesure.

7.4.2.2 Expérimentation avec des bigrammes

L'utilisation de bigrammes comme termes d'indexation, sous la forme de couples d'unités linguistiques, permet de retrouver l'équivalent des mots dans les langues européennes. Pour cela, nous avons parcouru le texte de gauche à droite en extrayant tous les bigrammes possibles. Par exemple, étant donnée une phrase ABCDE, les bi-gram extraits sont AB, BC, CD, DE. Cette méthode, bien que plus adaptée à la langue vietnamienne, produit beaucoup de bruit dans l'ensemble des termes d'indexation. En effet, si l'on considère, par exemple, le mot composé « công nghệ thông tin » (technologie d'information), ce dernier est découpé en les bigrammes suivants : « công nghệ » (technologie), « nghệ thông » (qui n'a pas de signification), et « thông tin » (information). La précision moyenne est néanmoins légèrement augmentée et vaut **0.3778**, ce qui prouve à notre avis, l'intérêt d'un découpage en mots.

7.4.2.3 Combinaison des bigrammes avec un lexique

Pour enlever les termes d'indexation non significatifs produits par la méthode des bigrammes, nous avons utilisé un lexique constitué d'environ 30 000 items pour ne garder que les bigrammes existant dans le lexique. Cette méthode a donné une très nette amélioration, puisque la précision moyenne est de **0.5625 (+20%).**

	unigrammes	bigrammes	combinaison
Précision moyenne	0.3636	0.3778	0.5625
% amélioration		1%	20%

Tableau 3. Pourcentage d'amélioration

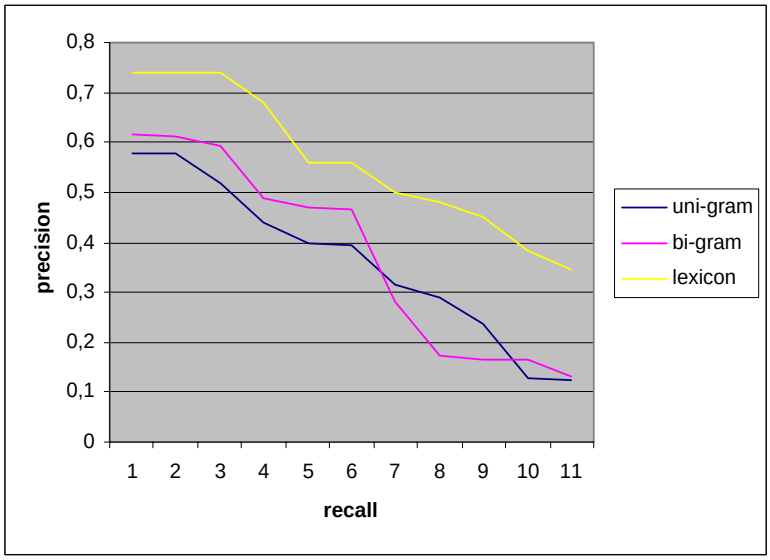


Figure 7.4-1 Courbe rappel – précision

Nous constatons que l'utilisation des bigrammess combinés avec un lexique est, pour l'instant, la meilleure stratégie d'indexation. Ce résultat semble donc indiquer que, dans cette langue, le découpage en mots corrects est très important pour la tâche de recherche d'information. La technique du lexique a ses limites, car seuls les mots appartenant au lexique sont indexés. Or dans la production de texte, il y a une part non négligeable de production de termes nouveaux.

C'est même souvent sur ces mots nouveaux qu'une recherche d'information est pertinente pour l'utilisateur.

7.4.3 Conclusion

Dans ce chapitre, nous avons présenté les spécificités du vietnamien du point de vue de la RI, et réalisé les premières expérimentations d'un système de RI vietnamien. De plus, nous avons construit un analyseur vietnamien pour catégoriser les mots et pour extraire des syntagmes nominaux, qui seront le support des étapes suivantes pour l'indexation par les syntagmes nominaux.

La première étape du traitement de la langue vietnamienne est terminée, et donne des résultats d'extraction satisfaisants. C'est la première fois qu'existent des outils automatiques de traitement de cette langue. Ils sont suffisamment généraux pour être utilisés dans un autre contexte. Nous avons spécialisé l'étape d'extraction des syntagmes pour la tâche de RI.

Le résultat présenté dans la figure 7.4-1 montre que l'utilisation de termes composés est plus efficace que celle de termes simples pour la RI en vietnamien.

Chapitre 8 Expérimentations

8.1 Introduction

Afin de tester le modèle proposé dans cette thèse, nous avons réalisé des expérimentations selon une démarche progressive. En premier lieu, nous avons testé les différents types de termes d'indexation : mot simple, syntagme, combinaison des deux pour vérifier notre hypothèse qui suppose que les termes d'indexation complexes permettent d'obtenir un SRI plus précis (au sens du rappel et précision). De manière pratique, nous avons utilisé le système XIOTA, basé sur le modèle vectoriel, et développé dans notre équipe de recherche (MRIM). Nous avons défini et implémenté plusieurs modules complémentaires à ceux de XIOTA, pour mettre en œuvre le modèle proposé dans le chapitre 5. Ce nouveau système a été expérimenté en monolingue et nous l'avons étendu à la multilingue.

8.2 Système XIOTA

Dans un premier temps, nous allons présenter les principales caractéristiques du système XIOTA. Ce système est un système de recherche d'information orienté vers la mise en place rapide d'expérimentations. Il est développé au sein de l'équipe de recherche MRIM [Chevallet 2004]. Toutes les données du système sont exprimées sous forme de fichiers XML. Tous les modules de ce système acceptent en entrée et produisent en sortie du XML. Pour réaliser une expérimentation, il suffit de constituer une chaîne de modules de traitements XML. En effet, l'intégration des modules se fait par le flot des données. Cette organisation simplifie la mise en œuvre des expériences, et permet d'avoir accès à tous les résultats intermédiaires, l'inconvénient est, par contre, un ralentissement dû aux codages et décodages successifs des données XML, mais le temps perdu dans le fonctionnement du système est gagné lors du développement de nouvelles expériences. De plus, le choix du langage de programmation des modules n'est pas imposé. Actuellement les modules sont programmés en C++ ou en PERL. De même, le langage pour l'assemblage des modules nécessaires pour une expérience est libre. Actuellement ce sont principalement des « scripts » du Shell Unix.

Nous présentons brièvement la structure des données et les processus principaux du système.

8.2.1 Structure de données

Le document (et/ou la requête) est représenté par un fichier XML qui contient les termes d'indexation sous la forme d'un vecteur. La DTD d'un vecteur est définie de la façon suivante :

```
< ! -- DTD pour un vecteur des termes d'indexation d'un document >
< ! -- Un vecteur -- >
<! ELEMENT VECTOR (C)+>
   <! ATTLIST VECTOR ID VALUE CDATA #REQUIRED
                     SIZE VALUE CDATA #REQUIRED >
   <! – ID est nom du vecteur, SIZE est nombre de termes du vecteur>
< ! -- Un terme -- >
< ! ELEMENT C EMPTY>
   < ! ATTLIST C ID VALUE CDATA #REQUIRED
                 W  VALUE CDATA #REQUIRED >
    < --! ID est un terme, W est la fréquence de ce terme dans le document>
```

Par exemple : la requête C091 du corpus CLEF2002,« Rapport dressé par Amnesty international sur les droits de l'homme en Amérique latine » peut se représenter par le fichier XML suivant :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vector id="C091" size="13">
   <c id="Amnesty" w="1" />
   <c id="Amnesty en Amérique" w="1" />
   <c id="Amnesty international" w="1" />
   <c id="Amnesty sur droit" w="1" />
   <c id="Amérique" w="1" />
   <c id="Amérique latin" w="1" />
   <c id="droit" w="1" />
   <c id="droit de homme" w="1" />
   <c id="droit en Amérique" w="1" />
   <c id="homme" w="1" />
   <c id="homme en Amérique" w="1" />
   <c id="rapport" w="1" />
   <c id="rapport dresser" w="1" />
</vector>
```

8.2.2 Schéma de base : indexation, interrogation

Nous supposons disposer d'un corpus organisé sous la forme de vecteurs, comme présenté précédemment, dans un seul fichier nommé « *corpus.vector* »[1]. Nous allons présenter plus précisément l'indexation et l'interrogation.

[1] Le fait de grouper tous les vecteurs dans un seul fichier n'est pas une obligation, mais il accélère les traitements car cela supprime les appels système d'ouverture et de fermeture de fichiers.

8.2.2.1 Indexation

La phase d'indexation est réalisée simplement par les étapes suivantes :

1. Création du fichier inverse

   corpus.vector | **xmlInverseMatrix** –w ltc > *corpus.inverse.ltc*

   où -w ltc est la méthode de pondération. Cette opération réalise une inversion de la matrice d'entrée et calcule la pondération affectée aux termes d'indexation.

2. Processus d'indexation de la matrice

   Cette opération générique, (c'est à dire applicable à tout fichier XML), a pour but d'en accélérer l'accès. On construit un index afin d'accéder de manière directe à un sous-arbre du fichier XML. Cela permet de constituer le classique "fichier inverse" d'un système de RI. Nous utilisons l'opération suivante :

   xmlIndex *corpus.inverse.ltc* vector id

   Cette opération construit l'index sur le fichier XML, c'est-à-dire la matrice inverse. En l'occurrence, il s'agit d'accéder de manière directe à partir d'un terme (balise id), au vecteur (balise vector) de tous les documents indexés par ce terme.

8.2.2.2 Interrogation

Pour cela, nous réalisons une multiplication vectorielle de la requête (ou des requêtes) par la matrice inverse en utilisant l'index du fichier de la matrice inverse pour accélérer le calcul. La multiplication d'une matrice Requête×Terme par la matrice Terme×Document donne la matrice Requête×Document, et constitue la réponse à la requête. Nous utilisons l'opération suivante :

cat requets.vector| **xmlVectorQuery** *corpus.inverse.ltc* | **xmlVector2trec** > *trec_top_file*

Le module final sert uniquement de mise au format nécessaire pour l'utilitaire de construction de la courbe rappel/précision *treceval*.

8.3 Notre système

Notre système utilise l'analyseur syntaxique XIP (Xerox Incremental Parser) de Xerox. XIP analyse des documents en langue naturelle et produit en sortie un ensemble de fichiers contenant dans une première partie une analyse de dépendance. Cette analyse catégorise tous les termes des phrases et propose pour chaque phrase une liste de dépendances. Cette liste de dépendances est résumée par un arbre de dépendance ou de constituants enregistré dans la

deuxième partie des fichiers (voir figure 1.3.1). C'est à partir de ces fichiers que nous réalisons la phase d'indexation. Nous présentons dans la suite les résultats de l'analyseur XIP, et donnons un exemple des résultats obtenus ainsi que l'architecture de notre système.

8.3.1 Résultats de l'analyseur XIP

La collection test traitée entièrement par l'analyseur XIP produit un ensemble de « fichiers XIP». Cet ensemble est constitué d'un ou plusieurs groupes analysés, chaque groupe correspond à une phrase dans le texte d'origine. Un groupe est formé de deux parties : une « partie dépendance » entre des mots du groupe et une partie « arbre de constituants » correspondant à la structure de la phrase.

1. La structure d'une relation dans la « partie dépendance » est représentée par :
   Nom_de _relation(<Terme^Lem+Categorie syntaxique : position>[,<>]…)

2. La partie « arbre de dépendance » entre des mots d'une phase est représentée par :
   numéro de groupe >GROUPE(position de debut : position de la fin) {{…}}

L'arbre de constituants propose une structuration arborescente complète de chaque phrase, la partie dépendance fournit des informations complémentaires sur les relations qu'entretient chaque constituant avec les autres, (en particulier le type de la relation, les variables grammaticales sont indiquées dans la relation). Le système peut proposer simultanément plusieurs types de relations pour un même couple de constituants.

8.3.2 Exemple d'analyse

Soit par exemple la phrase « *Rapport dressé par Amnesty international sur les droits de l'homme en Amérique latine* ». Elle est analysée et stockée sous la forme d'un fichier XIP comme suit :

```
DEEPSUBJ(<dressés^dresser^+SN+aSVINF+avoir+se+contreSN+aSN+PaPrt+Masc+PL+Ve
rb+PAP_PL:1>,<Amnesty^Amnesty^+InvGen+InvPL+Guessed+Noun+NOUN_INV:3,Interna
tional^international^+Masc+SG+Noun+NOUN_SG:4>)
VMOD_POSIT1_CLOSED_NOUN_INDIR(<dressés^dresser^+SN+aSVINF+avoir+se+contreSN
+aSN+PaPrt+Masc+PL+Verb+PAP_PL:1>,<par^par^+Prep+PREP:2>,<Amnesty^Amnesty^+
InvGen+InvPL+Guessed+Noun+NOUN_INV:3,International^international^+Masc+SG+N
oun+NOUN_SG:4>)
VMOD_POSIT2_NOUN_INDIR(<dressés^dresser^+SN+aSVINF+avoir+se+contreSN+aSN+Pa
Prt+Masc+PL+Verb+PAP_PL:1>,<sur^sur^+Prep+PREP:5>,<droits^droit^+deSN+aSVIN
F+deSVINF+surSN+Masc+PL+Noun+NOUN_PL:7>)
VMOD_POSIT2_NOUN_INDIR(<dressés^dresser^+SN+aSVINF+avoir+se+contreSN+aSN+Pa
Prt+Masc+PL+Verb+PAP_PL:1>,<en^en^+Prep+PREP_EN:11>,<Amérique^Amérique^+Fem
+SG+Proper+Noun+NOUN_SG:12>)
```

```
NMOD_POSIT1_RIGHT_CLOSED_ADJ(<Amérique^Amérique^+Fem+SG+Proper+Noun+NOUN_SG
:12>,<latine^latin^+Fem+SG+Adj+ADJ_SG:13>)
NMOD_POSIT1_RIGHT_ADJ(<Rapports^rapport^+entreSN+etSN+deSN+parSN+Masc+PL+No
un+NOUN_PL:0>,<dressés^dresser^+SN+aSVINF+avoir+se+contreSN+aSN+PaPrt+Masc+
PL+Verb+PAP_PL:1>)
NMOD_POSIT1_CLOSED_NOUN_INDIR(<Amnesty^Amnesty^+InvGen+InvPL+Guessed+Noun+N
OUN_INV:3,International^international^+Masc+SG+Noun+NOUN_SG:4>,<sur^sur^+Pr
ep+PREP:5>,<droits^droit^+deSN+aSVINF+deSVINF+surSN+Masc+PL+Noun+NOUN_PL:7>)
NMOD_POSIT1_CLOSED_NOUN_INDIR(<homme^homme^+Masc+SG+Noun+NOUN_SG:10>,<en^en
^+Prep+PREP_EN:11>,<Amérique^Amérique^+Fem+SG+Proper+Noun+NOUN_SG:12>)
NMOD_POSIT2_NOUN_INDIR(<droits^droit^+deSN+aSVINF+deSVINF+surSN+Masc+PL+Nou
n+NOUN_PL:7>,<en^en^+Prep+PREP_EN:11>,<Amérique^Amérique^+Fem+SG+Proper+Nou
n+NOUN_SG:12>)
NMOD_POSIT3_NOUN_INDIR(<Amnesty^Amnesty^+InvGen+InvPL+Guessed+Noun+NOUN_INV
:3,International^international^+Masc+SG+Noun+NOUN_SG:4>,<en^en^+Prep+PREP_E
N:11>,<Amérique^Amérique^+Fem+SG+Proper+Noun+NOUN_SG:12>)
NARG_POSIT1_CLOSED_NOUN_INDIR(<droits^droit^+deSN+aSVINF+deSVINF+surSN+Masc
+PL+Noun+NOUN_PL:7>,<de^de^+Prep+PREP_DE:8>,<homme^homme^+Masc+SG+Noun+NOUN
_SG:10>)
NN_NOUN(<Amnesty^Amnesty^+InvGen+InvPL+Guessed+Noun+NOUN_INV:3>,<Internatio
nal^international^+Masc+SG+Noun+NOUN_SG:4>)
PREPOBJ_CLOSED(<par^par^+Prep+PREP:2>,<Amnesty^Amnesty^+InvGen+InvPL+Guesse
d+Noun+NOUN_INV:3,International^international^+Masc+SG+Noun+NOUN_SG:4>)
PREPOBJ_CLOSED(<sur^sur^+Prep+PREP:5>,<droits^droit^+deSN+aSVINF+deSVINF+su
rSN+Masc+PL+Noun+NOUN_PL:7>)
PREPOBJ_CLOSED(<de^de^+Prep+PREP_DE:8>,<homme^homme^+Masc+SG+Noun+NOUN_SG:1
0>)
PREPOBJ_CLOSED(<en^en^+Prep+PREP_EN:11>,<Amérique^Amérique^+Fem+SG+Proper+N
oun+NOUN_SG:12>)
DETERM_CLOSED_DEF_NOUN_DET(<les^le^+InvGen+PL+Def+Det+DET_PL:6>,<droits^dro
it^+deSN+aSVINF+deSVINF+surSN+Masc+PL+Noun+NOUN_PL:7>)
DETERM_CLOSED_DEF_NOUN_DET(<l'^le^+InvGen+SG+Def+Det+DET_SG:9>,<homme^homme
^+Masc+SG+Noun+NOUN_SG:10>)
CLOSEDNP_NOUN(<Rapports^rapport^+entreSN+etSN+deSN+parSN+Masc+PL+Noun+NOUN_
PL:0>)
CLOSEDNP_PROPER_NOUN(<Amnesty^Amnesty^+InvGen+InvPL+Guessed+Noun+NOUN_INV:3
,International^international^+Masc+SG+Noun+NOUN_SG:4>)
CLOSEDNP_PROPER_NOUN(<Amérique^Amérique^+Fem+SG+Proper+Noun+NOUN_SG:12>)
CLOSEDNP_DEF_NOUN_DET(<homme^homme^+Masc+SG+Noun+NOUN_SG:10>)
CLOSEDNP_DEF_NOUN_DET(<droits^droit^+deSN+aSVINF+deSVINF+surSN+Masc+PL+Noun
+NOUN_PL:7>)
MWEHEAD_NOUN(<Amnesty^Amnesty^+InvGen+InvPL+Guessed+Noun+NOUN_INV:3,Interna
tional^international^+Masc+SG+Noun+NOUN_SG:4>,<Amnesty^Amnesty^+InvGen+InvP
L+Guessed+Noun+NOUN_INV:3>)

0>GROUPE(1:89){NP(1:9){NOUN{Rapports^rapport^+entreSN+etSN+deSN+parSN+Masc+
PL+Noun+NOUN_PL:0}},VERB{dressés^dresser^+SN+aSVINF+avoir+se+contreSN+aSN+P
aPrt+Masc+PL+Verb+PAP_PL:1},PP(18:43){PREP{par^par^+Prep+PREP:2},NP(22:43){
NOUN(22:43){NOUN{Amnesty^Amnesty^+InvGen+InvPL+Guessed+Noun+NOUN_INV:3},NOU
N{International^international^+Masc+SG+Noun+NOUN_SG:4}}}},PP(44:58){PREP{su
r^sur^+Prep+PREP:5},NP(48:58){DET{les^le^+InvGen+PL+Def+Det+DET_PL:6},NOUN{
droits^droit^+deSN+aSVINF+deSVINF+surSN+Masc+PL+Noun+NOUN_PL:7}}},PP(59:69)
{PREP{de^de^+Prep+PREP_DE:8},NP(62:69){DET{l'^le^+InvGen+SG+Def+Det+DET_SG:
9},NOUN{homme^homme^+Masc+SG+Noun+NOUN_SG:10}}},PP(70:81){PREP{en^en^+Prep+
PREP_EN:11},NP(73:81){NOUN{Amérique^Amérique^+Fem+SG+Proper+Noun+NOUN_SG:12
}}},AP(82:88){ADJ{latine^latin^+Fem+SG+Adj+ADJ_SG:13}},SENT{.^.^+SENT:14}}
```

1. Nous explicitons à titre d'exemple une des relations de ce fichier :

```
NN_NOUN(<Amnesty^Amnesty^+InvGen+InvPL+Guessed+Noun+NOUN_INV:3>,<Internatio
nal^international^+Masc+SG+Noun+NOUN_SG:4>)
```

Le nom de la relation est NN_NOUN. Elle représente la relation entre le terme « Amnesty » à la position 3 dans la phrase et le terme « international » à la position 4.

2. L'arbre de la phrase précédente est le suivant :

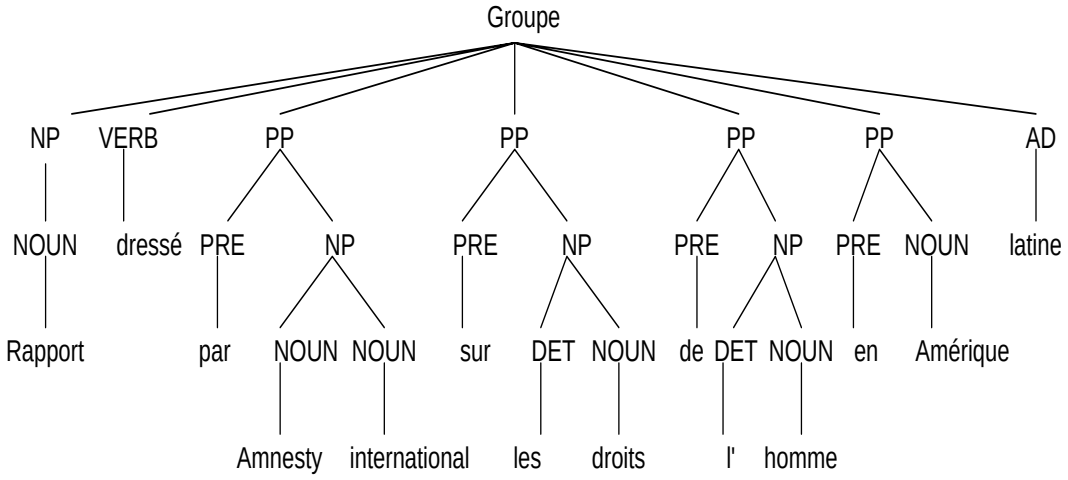


Figure 8.3-1 Arbre de dépendance

Sur cet exemple, on peut s'apercevoir que l'analyseur XIP, a bien reconnu le terme « Amnesty international », mais pas le terme « droit de l'homme » ni le terme « Amérique latine ». Cette faiblesse de l'analyse est préjudiciable à notre système, en effet, si les syntagmes ne sont ni reconnus ni structurés correctement, les hypothèses que nous avons faites pour notre modèle ne peuvent pas être parfaitement testées. Les résultats obtenus en utilisant cette structure sont dégradés.

8.3.3 Architecture de notre système

L'architecture de notre système est représentée par la figure suivante :

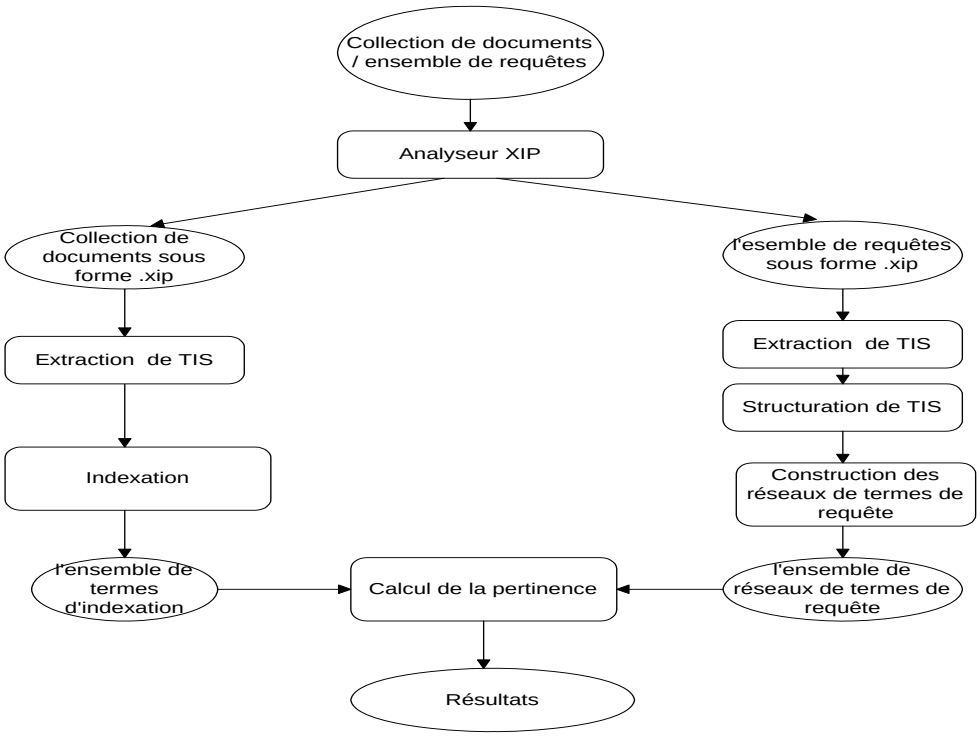


Figure 8.3-2 Architecture du système

Les documents et les requêtes sont analysés par le système XIP. Les termes d'indexation syntagmatiques sont ensuite extraits. Conformément à notre modèle, seuls les termes des requêtes servent à la construction du treillis de dérivation (par variation des termes simples aux termes composés). Ce choix se justifie par des considérations pragmatiques : pour l'implantation de notre solution 1 du chapitre 5, nous n'avons pas besoin du treillis des termes d'indexation des documents, mais uniquement de celui de la requête. Cette simplification réduit la quantité de calculs et la place occupée par les index des documents. Ainsi, dans cette expérience, seuls les termes de la requête sont organisés en treillis. Le calcul de pertinence se réalise entre ce treillis et le vecteur d'index du document. Nous pouvons ainsi, avec cette simplification, présélectionner les documents susceptibles d'être pertinents à l'aide du fichier inverse classique. Les dimensions de l'espace vectoriel sont représentés par des termes composés (TIS). Le calcul de la valeur de pertinence (Relevance Status Value), se réalise sur ce sous-ensemble. De manière pratique, nous avons ajouté les modules suivants :

1. ***xip2vector*** : Module qui prend en entrée un document au format XIP et crée en sortie un fichier contenant le vecteur des termes d'indexation syntagmatiques (TIS) du document. Chaque vecteur est un fichier XML de vecteurs conforme à la structure de vecteurs du système XIOTA présenté précédemment. Ce module réalise donc

l'extraction des termes d'indexation syntagmatiques à partir de l'analyse de dépendance sans utiliser l'arbre de dépendance construit.

2. ***inverse*** : Module qui crée une inversion de la matrice Document*Terme, calcule une pondération et construit le fichier d'inverse. Ce module a la même fonction que le module **xmlInverseMatrix** existant dans XIOTA, sauf que l'entrée de ce module est un ensemble de fichiers (résultats de XIP) et que la pondération utilisée est la formule ***tf*idf*** normalisée.

3. ***xip2syn*** : Module d'extraction des TIS à partir du fichier XIP. Ce module extrait des TIS en utilisant la deuxième partie des fichiers XIP, c'est-à-dire l'arbre de dépendance. La stratégie est d'extraire les plus grands TIS possibles à partir de l'arbre de dépendance fourni par XIP. Par rapport au module ***xip2vector***, il extrait des TIS à partir de la structure construite par XIP. Nous avons choisi deux méthodes différentes d'extraction de TIS afin de comparer la qualité des TIS obtenus en utilisant les deux types d'information contenus dans les fichiers XIP, résultats du traitement.

4. ***syn2reseau*** : Module de construction des réseaux à partir de TIS.

5. ***interrogationTIS*** : Module de calcul de la pertinence entre les requêtes et les documents en utilisant le réseau des TIS.

6. ***reseau_transformation*** : Module de transformation du réseau d'une requête en français en un réseau de la requête 'traduite ' en vietnamien.

8.4 Expérimentations monolingues

La collection test utilisée dans notre expérimentation est la collection de CLEF2002 [http://clef.iei.pi.cnr.it/]

8.4.1 Utilisation du modèle vectoriel

Comme nous l'avons précisé, nous utilisons le modèle vectoriel. Les expérimentations ont été conçues de la façon suivante :

1. Lors du premier test (RUN1), nous avons utilisé des termes simples avec les catégories syntaxiques substantif, adjectif et verbe. La pondération mise en œuvre est

« ltc ». Ce premier test nous a permis de mesurer l'amélioration (ou la dégradation) entraînée par l'utilisation de notre modèle.

2. Lors du deuxième test (RUN2), nous avons utilisé uniquement les TIS ayant une taille d'au moins de 2 mots ; les termes simples sont donc écartés. La pondération mise en œuvre est « ltc ».

3. Lors du troisième test (RUN3), nous avons utilisé une combinaison des deux indexations précédentes, c'est-à-dire que les termes d'indexation peuvent être des TIS ou des termes simples du corpus. La pondération est aussi « ltc ».

Les résultats sont les suivants :

	RUN1	RUN2	RUN3
Précision moyenne	0.2459	0.2225	0.2627

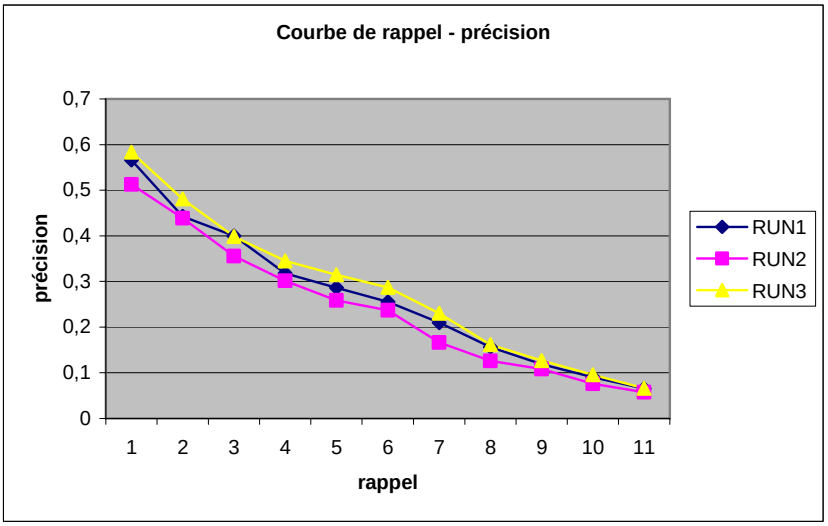


Figure 8.4-1 La courbe de rappel – précision

Nous constatons que l'utilisation des TIS seuls (RUN2) donne un résultat plus faible qu'une indexation par termes simples. Cela était prévisible, car on perd une partie de l'information véhiculée par les termes simples. Une indexation par la combinaison des termes simples avec des TIS (RUN3) produit un meilleur résultat. Ce résultat valide notre l'hypothèse que l'utilisation de syntagmes nominaux augmente la précision lorsqu'ils sont adjoints aux termes simples.

Dans la suite, nous présentons des expérimentations en utilisant le modèle proposé. Nous présenterons tout d'abord la structure de données utilisée pour représenter un réseau, puis les expérimentations faites avec ce modèle.

8.4.2 Structure de donnée d'un réseau

Le réseau est directement codé en XML. La structure comporte deux parties : une partie description des nœuds du réseau, et une partie description du treillis lui-même. Nous donnons ci-après la structure d'un réseau sous forme d'une DTD XML:

```
< ! -- réseau de termes de la requête>
< ! ELEMENT BAYESIANNETWORK (VARIABLE |
                              STRUCTURE)+>
   < ATTLIST BAYESIANNETWORK NAME ID #REQUIRED >
< ! -- partie de declaration des variables/noeuds du réseau>
< ! ELEMENT VARIABLE (VAR)+>
< ! ELEMENT VAR EMPTY >
   < ! ATTLIST VAR NAME ID #REQUIRED
                TYPE VALUE CDATA "discrete">
< ! - Partie de declaration de structure du réseau>
< ! ELEMENT STRUCTURE (CHILD)+>
< ! ELEMENT CHILD (PARENT)+>
   < ! ATTLIST CHILD ID ID #REQUIRED
               RULE VALUE CDATA #REQUIRED >
   < ! -- ID est nœud fils>
   < ! -- RULE est la règle de décomposition appliqué sur ce nœud >
< ! ELEMENT PARENT EMPTY>
   < ! ATTLIST PARENT ID ID #REQUIRED
               ROLE CDATA #REQUIRED
               P    VALUE CDATA>
   < ! -- ID est nœud obtenu après la décomposition >
   < ! -- ROLE est le rôle de terme : tête ou modifieur >
   < ! -- P est la probabilité de nœud >
```

Par exemple : La requête C091 du CLEF2002 est représentée sous forme de réseau bayésien comme suit :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE BAYESIANNETWORK PUBLIC "SYSTEM" "XBN.DTD">

<BAYESIANNETWORK NAME="C091" ROOT="root">
  <VARIABLE>
    <VAR NAME="Amnesty international" TYPE="discrete"></VAR>
    <VAR NAME="Amnesty international droit" TYPE="discrete"></VAR>
    <VAR NAME="Amnesty international droit homme Amérique latine"
TYPE="discrete"></VAR>
    <VAR NAME="Amérique" TYPE="discrete"></VAR>
    <VAR NAME="Amérique latine" TYPE="discrete"></VAR>
    <VAR NAME="droit" TYPE="discrete"></VAR>
    <VAR NAME="droit homme" TYPE="discrete"></VAR>
    <VAR NAME="droit homme Amérique latine" TYPE="discrete"></VAR>
    <VAR NAME="homme" TYPE="discrete"></VAR>
    <VAR NAME="homme Amérique" TYPE="discrete"></VAR>
    <VAR NAME="homme Amérique latine" TYPE="discrete"></VAR>
```

```
    <VAR NAME="latine" TYPE="discrete"></VAR>
  </VARIABLE>
  <STRUCTURE>
    <CHILD id="Amnesty international droit homme Amérique latine" rule="2"
p="0">
      <PARENT id="Amnesty international droit" role="tete" p="0" />
      <PARENT id="droit homme Amérique latine" role="modi" p="0" />
    </CHILD>
    <CHILD id="Amnesty international droit" rule="3" p="0">
      <PARENT id="Amnesty international" role="tete" p="0" />
      <PARENT id="droit" role="modi" p="0" />
    </CHILD>
    <CHILD id="droit homme Amérique latine" rule="2" p="0">
      <PARENT id="droit homme" role="tete" p="0" />
      <PARENT id="homme Amérique latine" role="modi" p="0" />
    </CHILD>
    <CHILD id="droit homme" rule="3" p="0">
      <PARENT id="droit" role="tete" p="0" />
      <PARENT id="homme" role="modi" p="0" />
    </CHILD>
    <CHILD id="homme Amérique latine" rule="2" p="0">
      <PARENT id="homme Amérique" role="tete" p="0" />
      <PARENT id="Amérique latine" role="modi" p="0" />
    </CHILD>
    <CHILD id="homme Amérique" rule="3" p="0">
      <PARENT id="homme" role="tete" p="0" />
      <PARENT id="Amérique" role="modi" p="0" />
    </CHILD>
    <CHILD id="Amérique latine" rule="3" p="0">
      <PARENT id="Amérique" role="tete" p="0" />
      <PARENT id="latine" role="modi" p="0" />
    </CHILD>
  </STRUCTURE>
</BAYESIANNETWORK>
```

8.4.3 Mise en œuvre de la correspondance par le treillis bayésien.

8.4.3.1 Test sur le corpus vietnamien

Nous avons utilisé le corpus vietnamien présenté dans le chapitre 7 pour ce test. Avec notre analyseur du vietnamien nous avons extrait les syntagmes nominaux. Nous avons ensuite structuré manuellement les requêtes sous la forme tête modifieur.

Nous avons réalisé les expérimentations suivantes sur le corpus vietnamien :

1. l'expérimentation (nommée RUN) utilise le modèle vectoriel basé sur des termes simples. Nous avons utilisé le schéma de pondération ltc ;
2. l'expérimentation (nommée RUN1) utilise le modèle proposé en pondérant un terme de la requête par 1 ou 0 selon sa présence dans le document examiné ;
3. l'expérimentation (nommée RUN2) utilise le modèle proposé en pondérant un terme de la requête par la valeur tf×idf normalisée pour représenter la présence et l'importance d'un terme de la requête dans le document examiné.

La pondération tf×idf normalisée est calculée par :

tf normalisé = tf / max(tf)

idf normalisé =log(N/n)/log(N)

où N est le nombre de documents du corpus

n est le nombre de documents qui contiennent le terme

Les résultats sont présentés dans la table et les courbes suivantes.

	Vectoriel (RUN)	Réseau (RUN 1)	Réseau (RUN2
Précision moyenne	0.4717	0.4086	0.5113

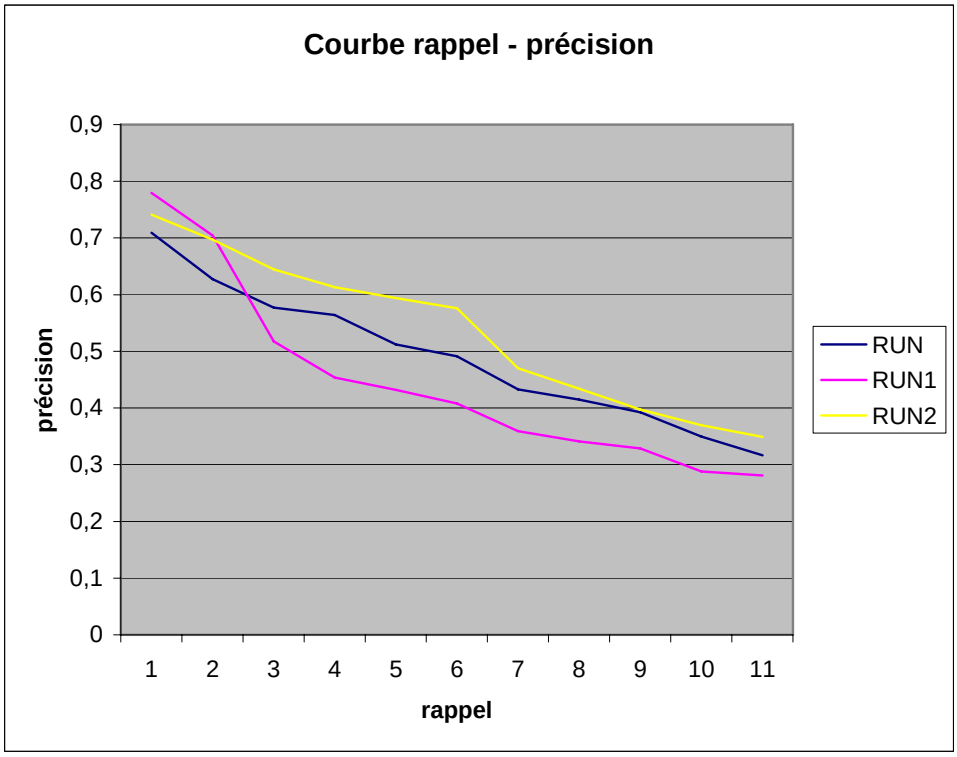


Figure 8.4-2 Courbe rappel - précision

Nous constatons que l'utilisation de notre modèle à base de treillis, avec une pondération des termes des requêtes par la valeur tf×idf normalisée, donne une meilleure précision moyenne que les deux autres méthodes. La pondération dans la dérivation bayésienne semble avoir beaucoup d'influence. En effet, une pondération booléenne (0 ou 1) pour les termes de la requête donne un résultat plus faible que le modèle vectoriel. L'importance relative d'un terme dans le document doit être explicitement prise en compte dans le calcul de la valeur de pertinence (Relevance Status Value).

8.4.3.2 Test sur le corpus français

Dans cette partie, nous décrivons la mise en œuvre et les résultats d'une indexation en utilisant le treillis des TIS et le réseau bayésien sur un corpus français. Nous avons réalisé ce test sur le corpus sda94 de CLEF formé de 43.178 documents et d'une taille de 2,7GB (format XIP) avec les 50 requêtes de la campagne CLEF2002.

Nous parcourons les réseaux des requêtes sur tous les documents contenant un ou plusieurs termes de la requête comme terme d'indexation du document. Pour chaque document, nous calculons la probabilité de pertinence selon la méthode proposée dans le chapitre 5. Nous avons testé deux méthodes de calcul de la probabilité d'un terme de la requête par rapport à un document :

1. d'une part, l'utilisation de 1 ou 0 comme probabilité de présence d'un terme de la requête pour un document donné : 1 si ce terme existe dans le document et 0 sinon. Pour les autres nœuds du réseau, la probabilité est calculée selon les probabilités de ses pères comme nous l'avons présentée dans le chapitre 5 (RUN1).

2. d'autre part, l'utilisation de tf×idf comme probabilité de présence et d'importance d'un terme de la requête par rapport au document examiné. Pour les autres nœuds dans le réseau, la probabilité est calculée selon les probabilités de ses pères, comme nous l'avons présentée dans le chapitre 5 (RUN2).

Nous avons utilisé le résultat de l'indexation par termes simples sur le même corpus et la même requête avec le modèle vectoriel (RUN) comme base de comparaison.

Avec la méthode de correspondance utilisant le réseau bayésien, nous avons obtenu un résultat bien plus faible que celui du modèle vectoriel de référence (voir les courbes dans la figure 1.2). La cause de ce résultat négatif peut être due à deux problèmes :

1. Les syntagmes nominaux produits par XIP ne sont pas corrects ;

2. La structuration d'un syntagme sous la forme tête modifieur n'est pas correcte.

Notre modèle calcule la valeur de pertinence entre la requête et le document en considérant le rôle du terme et la pondération du mot tête. Par conséquent, si la tête est incorrecte, une importance sera donnée à un terme qui n'en n'a pas réellement et la précision diminuera.

	RUN	RUN1	RUN2
Précision moyenne	0.2706	0.1035	0.0608

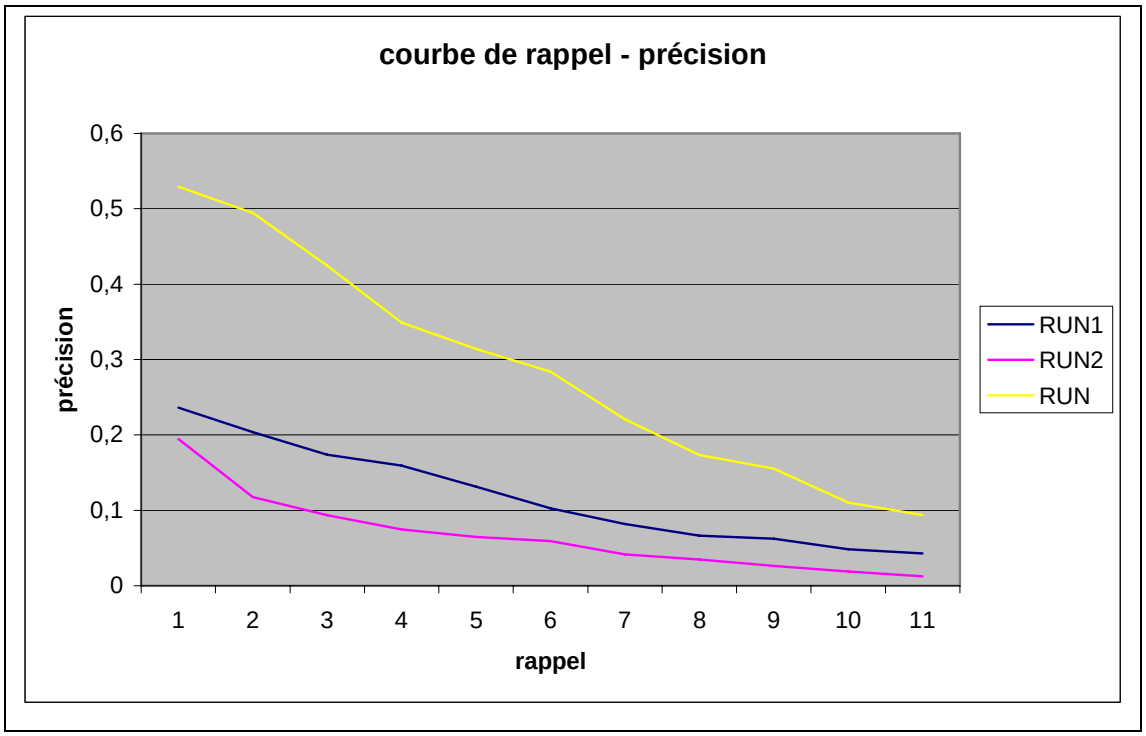


Figure 8.4-3 Courbe de rappel – précision

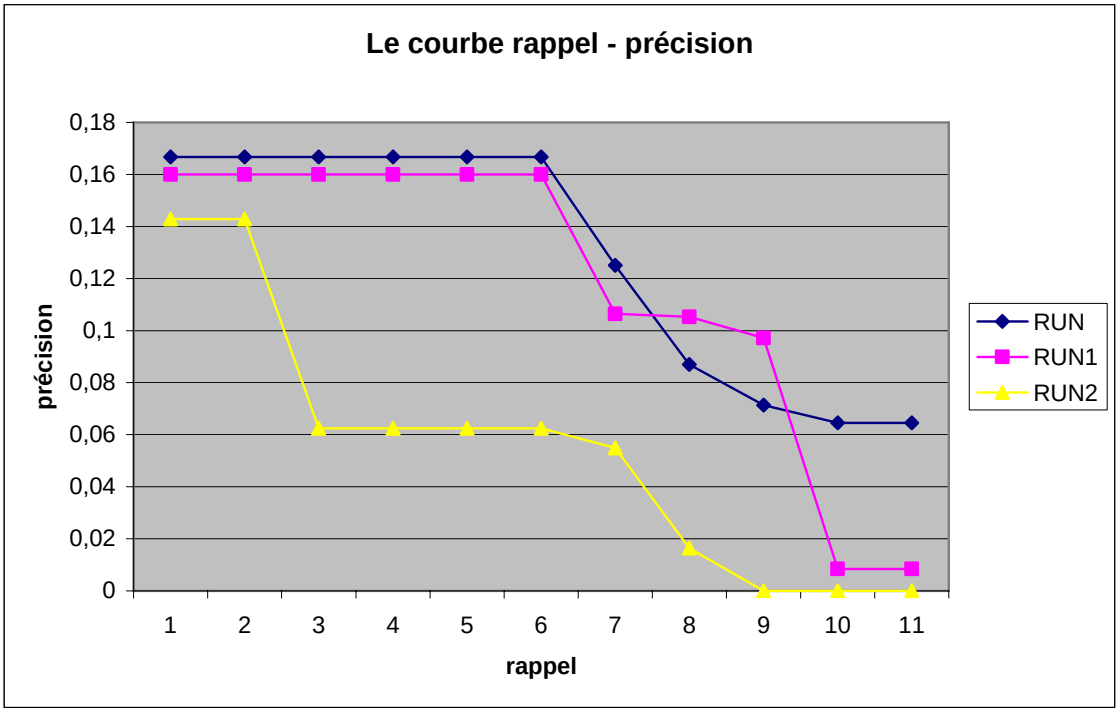
RUN1 : utilisation du réseau et pondération booléenne (0 ou 1)

RUN2 : utilisation du réseau et pondération tf*idf normalisée

RUN : modèle vectoriel traditionnel avec termes simples

Afin d'aller plus loin dans l'investigation des causes de cette mauvaise performance, nous avons modifié manuellement la structure de la requête, pour donner en entrée au système une requête correctement structurée. Nous ne l'avons fait que pour la requête C091 et relancé les trois tests RUN, RUN1, RUN2 précédents. Les résultats, sur cette requête seule, sont les suivants.

	RUN	RUN1	RUN2
Précision moyenne	0.1074	0.1133	0.0454



L'expérimentation RUN1 (réseau et pondération booléenne) donne le meilleur résultat. La précision moyenne est 0,1133 par rapport à 0,1074 pour le modèle vectoriel classique (RUN). Cela confirme donc, d'une part, l'importance d'une bonne structuration et, d'autre part, la validité de notre approche.

De manière pratique, nous en concluons que la sortie du système XIP, n'est pas directement exploitable pour la RI, avec le modèle que nous avons proposé. De plus, l'utilisation de tf*idf dans l'expérimentation RUN2 donne un résultat faible. Nous en déduisons que cette pondération n'est pas adaptée pour estimer l'importance des termes composés. En effet, la valeur *idf* des termes composés est plus grande que celle des termes simples. Si un document contient un ou plusieurs termes composés en commun avec la requête, et même si ces termes ne sont pas importants pour la requête, ce document sera évalué comme plus pertinent que des documents qui contiennent des termes simples importants pour la requête.

Par exemple, le document ATS.940201.0052 contient deux TIS en communs avec la requête C091 : « Amnesty international » et « droits de l'homme ». Les valeurs tf*idf normalisées de ces termes sont respectivement 0,0189 et 0,1229. Le score de pertinence de ce document pour

la requête est rangé au deuxième rang alors qu'il n'est vraiment pas pertinent pour la requête C091.

Le contenu du document ATS.940201.0052 est :

```
<DOCID>ATS.940201.0052</DOCID>
<DOCNO>ATS.940201.0052</DOCNO>
<KW>
zaire amnesty droits de l homme
</KW>
<TI>
Violations des droits de l'homme au Zaïre Amnesty lance un appel à la
communauté internationale embargo mercredi 2 février 01.00 heure.
</TI>
<LD>
Londres/Berne, 1er ftv (ats) Amnesty international (AI) a lancé mercredi un
appel à la communauté internationale pour qu'elle exhorte les autorités
zaïroises à respecter les droits de l'homme. Selon AI, des millier
d'assassinats politiques ont été perpétrés depuis 1990.  Les forces l'ordre
toléreraient ces exactions ou en seraient directement responsables.
</LD>
<TX>
Des opposants et des journalistes ont récemment été victimes de mauvais
traitements, de torture et d'assassinat politique, indique l'organisation
de défense des Droits de l'homme par communiqué. Ses constatations
ressortent de son dernier rapport sur le Zaïre intitulé "Zaïre: un pays qui
s'effondre".
</TX>
<TX>
Le président Mobutu s'est jusqu'ici borné à accuser AI de voul déstabiliser
le Zaïre. Aucune mesure n'a été prise pour enquêter sur les faits dénoncés
par l'organisation humanitaire. En septembre dernier, les évêques zaïrois
déclaraient que "ces pillages, ces conflits ethniques, ces enlèvements et
ce carnage attestaient de la folie et de la mort morale de l'Etat zaïrois
déchaîné contre sa propre population".
</TX>
<TX>
Amnesty appelle les gouvernements à envoyer leurs représentants dans la
régions où se commettent les exactions, à condamner les violations des
droits de l'homme et à faire pression sur le président Mobutu pour que ces
violations cessent. AI demande également aux Nations Unies et
l'Organisation de l'unité africaine (OUA) de nommer des représentant
officiels chargés de surveiller la situation des droits de l'homme au Zaïre.
</TX>
</DOC>
```

Il est représenté par le vecteur suivant :

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<vector id="ATS.940201.0052" size="155">
  ……..

  <c id="Amnesty international" w="0.2" />
  …….
  <c id="droit homme" w="1" />
  ……..
  <c id="homme" w="1" />
  ……
```

```
</vector>
```

La requête C091 est :

« Rapport dressée par Amnesty international sur droits de l'homme en Amérique latine » est représentée comme dans la section 1.4.2.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE BAYESIANNETWORK PUBLIC "SYSTEM" "XBN.DTD">

<BAYESIANNETWORK NAME="C091" ROOT="root">
  <VARIABLE>
    <VAR NAME="Amnesty international" TYPE="discrete"></VAR>
    <VAR NAME="Amnesty international droit" TYPE="discrete"></VAR>
    <VAR NAME="Amnesty international droit homme Amérique latine"
TYPE="discrete"></VAR>
    <VAR NAME="Amérique" TYPE="discrete"></VAR>
    <VAR NAME="Amérique latine" TYPE="discrete"></VAR>
    <VAR NAME="droit" TYPE="discrete"></VAR>
    <VAR NAME="droit homme" TYPE="discrete"></VAR>
    <VAR NAME="droit homme Amérique latine" TYPE="discrete"></VAR>
    <VAR NAME="homme" TYPE="discrete"></VAR>
    <VAR NAME="homme Amérique" TYPE="discrete"></VAR>
    <VAR NAME="homme Amérique latine" TYPE="discrete"></VAR>
    <VAR NAME="latine" TYPE="discrete"></VAR>
  </VARIABLE>
  <STRUCTURE>
    <CHILD id="Amnesty international droit homme Amérique latine" rule="2"
p="0">
      <PARENT id="Amnesty international droit" role="tete" p="0" />
      <PARENT id="droit homme Amérique latine" role="modi" p="0" />
    </CHILD>
    <CHILD id="Amnesty international droit" rule="3" p="0">
      <PARENT id="Amnesty international" role="tete" p="0" />
      <PARENT id="droit" role="modi" p="0" />
    </CHILD>
    <CHILD id="droit homme Amérique latine" rule="2" p="0">
      <PARENT id="droit homme" role="tete" p="0" />
      <PARENT id="homme Amérique latine" role="modi" p="0" />
    </CHILD>
    <CHILD id="droit homme" rule="3" p="0">
      <PARENT id="droit" role="tete" p="0" />
      <PARENT id="homme" role="modi" p="0" />
    </CHILD>
    <CHILD id="homme Amérique latine" rule="2" p="0">
      <PARENT id="homme Amérique" role="tete" p="0" />
      <PARENT id="Amérique latine" role="modi" p="0" />
    </CHILD>
    <CHILD id="homme Amérique" rule="3" p="0">
      <PARENT id="homme" role="tete" p="0" />
      <PARENT id="Amérique" role="modi" p="0" />
    </CHILD>
    <CHILD id="Amérique latine" rule="3" p="0">
      <PARENT id="Amérique" role="tete" p="0" />
      <PARENT id="latine" role="modi" p="0" />
    </CHILD>
  </STRUCTURE>
</BAYESIANNETWORK>
```

8.4.4 Constat

Nous constatons que la performance de notre modèle dépend directement de la qualité de la structuration en syntagmes nominaux de la requête, ainsi que de la structuration des termes d'indexation des documents. Si ces syntagmes sont bien structurés, nos expériences montrent que la précision du système est améliorée par rapport à celle du modèle vectoriel avec des termes simples. Par contre, il semble que la pondération classique tf*idf ne soit pas adaptée à notre approche. Un autre schéma de pondération reste donc à définir et à tester.

8.5 Expérimentation multilingue

Cette partie décrit une application de notre modèle à un SRI multilingue. Plus précisément, nous nous sommes concentré sur deux langues : le français et le vietnamien. Pour l'heure, il s'agit donc d'un SRI bilingue. La première tâche consiste à identifier les ressources linguistiques nécessaires à la traduction des requêtes, et à les mettre en forme (format XML).

8.5.1 Construction d'un lexique d'associations

À partir d'un dictionnaire bilingue français - vietnamien disponible sur Internet à l'adresse *http://www.informatik.uni-leipzig.de/~duc/software/Dict/vietdict.html*, nous avons construit un dictionnaire de traduction. Chaque entrée est associée à un ensemble de traductions possibles. La valeur p correspond à la probabilité de cette traduction. Par exemple, pour le mot "action" en français, nous avons les traductions suivantes en vietnamien :

```
<Entry="action">
   <equiv="sự hoạt động" p="0.0714285714285714"/>
   <equiv="sự thực hành" p="0.0714285714285714"/>
   <equiv="hành động" p="0.0714285714285714"/>
   <equiv="tác dụng" p="0.0714285714285714"/>
   <equiv="công trạng" p="0.0714285714285714"/>
   <equiv="cuộc chiến đấu" p="0.0714285714285714"/>
   <equiv="bộ điệu" p="0.0714285714285714"/>
   <equiv="nhiệt tình" p="0.0714285714285714"/>
   <equiv="sự hùng biện" p="0.0714285714285714"/>
   <equiv="cốt truyện" p="0.0714285714285714"/>
   <equiv="tiến trình" p="0.0714285714285714"/>
   <equiv="văn kiện" p="0.0714285714285714"/>
   <equiv="tố quyền" p="0.0714285714285714"/>
   <equiv="cổ phiếu" p="0.0714285714285714"/>
</Entry>
```

La valeur d'attribut "equiv" contient une traduction de l'attribut "Entry", et la valeur d'attribut " p" est la probabilité que " equiv " soit une traduction de " Entry " (notée $Pr(c_j|s_i)$ à la page 91 dans le chapitre 6). En l'absence d'autres informations, nous supposons que tous les termes traduits ont la même probabilité d'être une traduction de l'entrée ("Entry"). Si nous

disposons d'autres informations, comme un corpus parallèle aligné, nous pouvons utiliser cette connaissance et apprendre les probabilités de traduction entre des termes. Il faut noter que cette information probabiliste est contextuelle, et ne peut pas se généraliser de façon fiable, en dehors de tout domaine applicatif.

8.5.2 Construction d'un thésaurus de cooccurrences de termes vietnamiens

Afin de choisir la bonne traduction d'un couple de mots, nous avons construit un thésaurus de cooccurrences de termes dans le corpus vietnamien. Nous avons utilisé la mesure de cooccurrence simple :

$$C(x,y) = \frac{|\,x \text{ and } y\,|}{|\,x\,|}$$

où |x and y| est le nombre d'occurrences où x et y se situent côte à côte dans le corpus

|x] est le nombre d'occurrences de x dans le corpus

Ce thésaurus est stocké dans un fichier XML :

```
<vector id="Chi_cục" >
   <c id="Bảo vệ"  w="0.0479704797047971"/>
   <c id="Kiểm lâm"  w="0.040590405904059"/>
   <c id="thuế"  w="0.044280442804428"/>
   <c id="trưởng"  w="0.0590405904059041"/>
</vector>
```

8.5.3 Transformation du réseau de la requête

En se basant sur le lexique bilingue et le thésaurus de cooccurrences de termes, nous transformons le réseau de la requête en français vers un réseau correspondant en vietnamien selon l'algorithme présenté dans le chapitre 6.

Pour l'expérimentation, nous avons traduit 14 requêtes du corpus vietnamien vers le français. Les requêtes traduites sont structurées manuellement sous la forme tête, modifieur. Automatiquement, nous avons créé les réseaux représentant ces requêtes et transformé ces réseaux en des réseaux correspondants en vietnamien.

Les réseaux obtenus en vietnamien sont ensuite confrontés au corpus vietnamien. Nous testons ainsi deux méthodes de pondération :

1. MULT1 : correspond à une pondération binaire. Elle est à comparer avec l'expérimentation MONO1 monolingue, car elle utilise la même méthode de pondération.

2. MULT2 : correspond à une pondération de type tf∗idf normalisée. Elle est à comparer avec MONO2, l'expérimentation correspondante en monolingue.

Les résultats comparatifs de ces quatre expérimentations sont présentés dans le tableau suivant en utilisant les valeurs de précision sur 11 points de rappel :

	0,00	0,10	0,20	0,30	0,40	0,50	0,60	0,70	0,80	0,90	1,00
MUL1	0,2228	0,1581	0,1458	0,1384	0,1310	0,1293	0,1238	0,1228	0,1219	0,1057	0,0698
MUL2	0,4209	0,3105	0,2717	0,2676	0,2651	0,2581	0,1909	0,1823	0,1635	0,1416	0,098
MON1	0,7796	0,7046	0,5177	0,4537	0,4317	0,4082	0,3591	0,3409	0,3287	0,2877	0,2811
MON2	0,7413	0,6973	0,6444	0,6129	0,5940	0,5757	0,4704	0,4343	0,3972	0,3696	0,3493

La table de comparaison des valeurs de précision moyenne :

	Méthode 1	Méthode 2
Monolingue	0,4086	0,5113
Multilingue	0,1181	0,2256
%	28%	44%

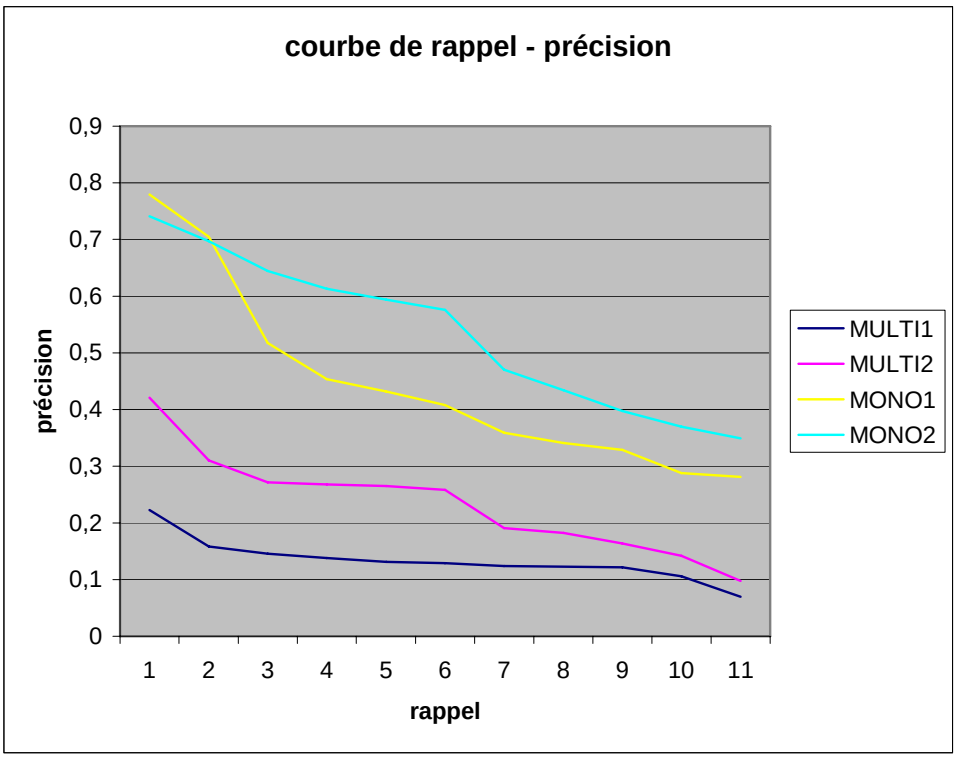


Figure 8.5-1 Courbe de rappel - précision

Les résultats du système bilingue sont plus faibles que celles du monolingue. Ce constat est en général normal. Nous expliquons cette différence par les raisons suivantes :

1. Beaucoup de termes sont absents du dictionnaire utilisé. Le dictionnaire de traduction est de trop faible taille. Beaucoup de mots importants n'y apparaissent pas. Par exemple : la requête 12 « l'investissement étranger au Vietnam ». Le terme «investissement » ne se trouve pas dans le dictionnaire. Dans ce cas, nous avons choisi de garder le mot original, ce qui évidemment ne donnera pas de bons résultats, puisque ce mot est important dans la requête ;

2. Le thésaurus de cooccurrences de termes n'est pas assez important, du fait de la faible taille du corpus dont nous disposons en vietnamien. Plusieurs couples de termes dans la requête ne s'y trouvent pas. Dans ce cas, le système choisit le premier terme donné par le dictionnaire bilingue.

Nous avons vérifié sur des cas particuliers que les raisons que nous invoquions précédemment lors de l'analyse de la faiblesse des résultats sont valides. Soit par exemple la requête 1 en

vietnamien « kinh tế tri thức ». Elle a été traduite en français par « économie des connaissances ». Le réseau représentant cette requête en français est le suivant :

```
<?xml version="1.0"  encoding="UTF-8"?>
<BAYESIANNETWORK NAME="01" ROOT="root">
  <VARIABLE>
    <VAR NAME="connaissance" TYPE="discrete"></VAR>
    <VAR NAME="économie" TYPE="discrete"></VAR>
    <VAR NAME="économie connaissance" TYPE="discrete"></VAR>
  </VARIABLE>
  <STRUCTURE>
    <CHILD id="économie connaissance" rule="3" p="0">
       <PARENT id="économie" role="tete" p="0"/>
       <PARENT id="connaissance" role="modi" p="0"/>
    </CHILD>
  </STRUCTURE>
</BAYESIANNETWORK>
```

Aux deux termes français « économie » et «connaissance » correspondent deux listes de termes vietnamiens données par le dictionnaire :

économie	connaissance
kinh tế kinh tế học tính tiết kiệm tính dè sẻn tiền tiết kiệm	hiểu biết nhận thức tri thức kiến thức tri giác quen biết giao thiệp người quen

Le couple {kinh tế ; tri thức} apparaît dans le thésaurus de cooccurrences avec la plus grande valeur de cooccurrence parmi tous les couples possibles. Il est donc choisi comme traduction du couple {économie ; connaissance}. Le réseau traduit est donc le suivant :

```
<?xml version="1.0"  encoding="UTF-8"?>
<BAYESIANNETWORK NAME="01" ROOT="root">
  <VARIABLE>
    <VAR NAME="kinh_tê" TYPE="discret"></VAR>
    <VAR NAME="kinh_tê tri_thúc" TYPE="discret"></VAR>
    <VAR NAME="tri_thúc" TYPE="discret"></VAR>
  </VARIABLE>
  <STRUCTURE>
    <CHILD id="kinh_tế tri_thúc" rule="3" p="0">
      <PARENT id="kinh_tế role="tete" p="0"/>
      <PARENT id="tri_thúc" role="modi" p="0"/>
    </CHILD>
   </STRUCTURE>
</BAYESIANNETWORK>
```

Les deux réseaux (vietnamien et français) sont identiques et correspondent à une bonne traduction. Le système bilingue donne le même résultat que le système monolingue. Dans cette expérimentation, nous constatons que le nombre de requêtes bien traduites (c'est-à-dire que tous les nœuds du français sont transformés vers des nœuds corrects en vietnamien) est seulement 2 sur 14. Pour les autres, on observe qu'une partie des nœuds constitue une mauvaise traduction. Cette observation explique, en partie, la faiblesse du système bilingue par rapport à notre système monolingue.

C'est pour cela que nous pensons que l'impact de la qualité du dictionnaire bilingue est non négligeable, en particulier son degré de couverture de la langue, mais aussi la qualité des traductions proposées. Les ressources linguistiques jouent, en effet, un rôle prépondérant dans la qualité des résultats. Nous pensons également que les corpus parallèles alignés permettent de compléter efficacement des ressources linguistiques statistiques, et doivent fournir une information pragmatique contextuelle sous la forme de probabilités de traduction, à même d'orienter favorablement le système vers les meilleures traductions.

8.5.4 Conclusions

Si l'on se réfère aux résultats obtenus dans la campagne CLEF 2004 [Chevallet et al. 2004], nous nous apercevons que notre système se situe dans la moyenne de la plupart des systèmes ayant travaillé sur le bilinguisme.

Chapitre 9 Conclusion & perspectives

9.1 Conclusion

Dans cette thèse, nous avons proposé un modèle de recherche d'information à base de syntagmes nominaux structurés. L'objectif global de la thèse est de définir un système de recherche d'information « réellement multilingue ». Dans un premier temps, nous avons mesuré l'impact des syntagmes nominaux pour des langues comme le vietnamien où les unités linguistiques porteuses de sens sont les termes composés. Nous avons ensuite étudié l'adaptation d'un modèle de RI pour prendre en compte un contexte plus large que le terme isolé classiquement utilisé comme terme d'indexation. De ce fait, nous proposons de représenter le terme d'indexation avec un peu plus de sémantique. Pour cela, nous avons structuré les syntagmes nominaux. La traduction des requêtes est alors moins approximative que lorsque l'on traduit simplement les mots-clés de la requête, car le syntagme nominal structuré est lui même plus précis. Les caractéristiques de ce modèle peuvent être résumé par les points suivants.

9.1.1 Terme d'indexation

Dans notre modèle, nous utilisons des syntagmes nominaux comme termes d'indexation. Les expérimentations effectuées sur le vietnamien (chapitre 7) nous ont permis de vérifier la validité de ce choix. En effet, l'utilisation des syntagmes nominaux combinés avec des termes simples a augmenté la précision du système. Une autre idée, que nous avons partiellement validée, concerne l'apport de la structuration des syntagmes nominaux à la qualité de l'indexation.

9.1.2 Structure de termes d'indexation

9.1.2.1 Structure interne

Nous avons proposé une structuration des syntagmes nominaux, sous la forme de tête et modifieur. Nous avons proposé un ensemble de règles qui permettent de construire un réseau de termes d'indexation à partir d'un terme donné. Ce réseau représente les relations de déduction possibles entre les termes et leurs variations syntaxiquement correctes. Cette structure nous permet alors de construire un nouveau schéma de pondération en fonction de la

position du terme et de son rôle dans le syntagme. En effet, nous avons supposé que la tête d'un syntagme a plus d'importance que ses modifieurs et nous en avons déduit des fonctions de pondération.

9.1.2.2 Structure externe

Les termes d'indexation sont extraits directement du contenu des documents ou bien sont créés par applications des règles de décomposition (chapitre 4). Les termes d'indexation sont organisés en treillis de type réseau bayésien. Chaque réseau permet de déduire un terme plus grand (donc plus précis) à partir de termes plus petits (donc plus généraux). Cette structure sert de base au calcul de la fonction de correspondance (ou valeur de pertinence) entre une requête et les documents.

9.1.3 Fonction de correspondance

La valeur de pertinence est calculée par un processus de propagation de la probabilité des nœuds du réseau construit pour la requête. La probabilité des nœuds, qui existent dans le document examiné, est estimée selon la présence et l'importance du terme correspondant de la requête dans le document. Les probabilités conditionnelles sont estimées en tenant compte de la structure du réseau en intégrant le rôle des nœuds pères dans un nœud fils.

Notre proposition a pour but d'augmenter la performance du système de recherche d'information, en particulier sa précision. La performance de ce modèle est donc fortement liée à la phase de construction et de structuration d'un syntagme nominal sous la forme de tête modifieur. Si la structuration est correcte, le résultat est meilleur qu'avec le modèle vectoriel traditionnel, car nous capturons directement dans l'index et dans la fonction de correspondance, l'usage et le contexte des termes. Ce point a été constaté expérimentalement dans les tests effectués avec une seule requête bien structurée C091. Cependant, si la structuration est incorrecte, le résultat s'avère plus faible qu'avec un modèle à base de termes isolés, car l'association ne reflète plus la sémantique du terme et produit du bruit néfaste au système. Nous l'avons également constaté dans une expérimentation sur 50 requêtes. Pour qu'un tel modèle fonctionne correctement, il faut donc disposer d'un module de structuration performant. Malheureusement, force est de constater que nous sommes loin de ce cas de figure avec l'utilisation du système d'analyse XIP. Nous pensons qu'il est possible par un post traitement (basé sur des connaissances explicites ou des statistiques issues du corpus, ou du Web), d'améliorer nettement les sorties de ce module, ce que nous n'avons pu faire faute de

temps, pour attester que cette piste est vraiment intéressante. Nous savons également que la construction d'un module de qualité de structuration des syntagmes est un effort non négligeable qui va au delà de ce travail et de l'objectif de cette thèse.

9.1.4 Transformation de la requête

Notre modèle a ensuite été étendu pour une recherche multilingue en introduisant une phase de transformation d'une requête dans une langue vers une autre langue. Cette phase de transformation utilise les réseaux de termes de la requête. Les deux niveaux les plus bas du réseau sont alors transformés en couplant un dictionnaire bilingue avec des informations statistiques de cooccurrence des termes dans la langue cible.

9.2 Perspectives

Parmi les questions qui restent en suspens et qui méritent plus de réflexion et d'expérimentation, nous pouvons noter :

1. l’estimation de la probabilité des nœuds du réseau de la requête qui existent dans un document examiné. En effet, plusieurs autres schémas de probabilité sont possibles et devraient être testés.

2. l’utilisation des réseaux de termes d’indexation d’un document. Il est possible de poursuivre la réflexion en incluant, non pas seulement les syntagmes nominaux, mais une plus grande partie des phrases, en particulier (pour tenir compte, en particulier, de la variation par verbalisation nominalisation). Par exemple la phrase « le pétrole pollue l'eau de mer », pourrait être considérée pour l’indexation au même titre que le temrme « la pollution de l'eau de mer par le pétrole ».

3. la réalisation et l'évaluation la deuxième solution proposée dans cette thèse concernant le calcul de la correspondance.

A plus long terme, on peut se poser la question d'une formulation plus générique de notre modèle. En particulier, de manière théorique, il faudrait établir sa relation avec le modèle logique de RI. En effet, nous pensons que le calcul de correspondance que nous proposons s'apparente effectivement à une opération de déduction pondérée (ou floue). Une formalisation plus précise nécessiterait de fournir une sémantique formelle au modèle des

termes structurés; et cela peut probablement conduire à une description dans un formalisme logique.

Concernant l'aspect "traitement de langue naturelle", nous ne pouvons qu'espérer disposer, dans un futur proche, d'analyseurs plus performants au niveau de la qualité de la structuration des syntagmes : la qualité du SRI en dépend directement. La communauté RI n'a malheureusement que peu d'emprise sur l'évolution de ce domaine.

Finalement, nous pensons que l'extension de ce modèle pour une recherche multilingue n'est pas encore assez étayée. De plus, nous pressentons le besoin d'une étude plus approfondie pour trouver une transformation plus efficace d'une requête d'une langue vers une autre langue.

A ce propos, nous avons dessiné les premières étapes à mettre en place pour l'utilisation d'un langage pivot pour une recherche "pleinement" multilingue, et le langage UNL (Universal Network Language) étudié pour ce but, est un candidat potentiel prometteur. Cependant, nous ne pouvons pas prévoir ni son intérêt réel dans la tâche de RI, et encore moins prédire son développement à un niveau d'utilisation en vraie grandeur pour l'expérimentation en RI. Il est actuellement impossible de disposer d'un corpus de la taille de ceux utilisé en RI, codé dans ce formalisme du fait de la trop faible couverture des « enconvertisseurs » actuels.

Index

Table de figures

Bibliographie

[Adriani et al. 2000] *Mirna Adriani, C.J. van Rijsbergen* – Phrase Identification in Cross-Language Information Retrieval - *Proceeding of RIAO'2000, 2000.*

[Arampatzis 2000] *A.T. Arampatzis, Th.P. van der Weide, P. van Bommel et C.H.A. Koster*- Linguistically Motivated Information Retrieval- Encylopedia of Library and Information Science*, Marcel Dekker, Inc, New York, Basel, 2000.*

[Baeza et al. 99] *Ricardo Baeza-Yates et Berthier Riberio-Neto*- Modern Information Retrieval – *Addison Weslay – New York 1999.*

[Ballesteros et al. 96] *Lisa Ballesteros, Bruce Croft* – Dictionary methods for cross-lingual information retrieval – *Proceeding of the 7th International DEXA Conference on Database and Expert Systems, pages 791-801, 1996.*

[Ballesteros et al. 97] *Lisa Ballesteros, W. Bruce Croft* – Phrasal Transalation and Query Expansion Techniques for Cross-Language Information Retrieval – *ACM SIGIR 97.*

[Ballesteros et al. 98] *Lisa Ballesteros, W. Bruce Croft* – Resolving Ambiguity for Cross-Language Retrieval – *ACM SIGIR 98.*

[Berger et al. 99] *Berger A., Lafferty J.*- Information retrieval as statistical translation - *Hearst et al. 99, pages 222-229, 1999.*

[Brill 95] *Eric Brill* – Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging – *Computational Linguistic, 1995.*

[Brill 96] *Eric Brill* – Recent Advances in Parsing technology – *chapter Learning to Parse with Transformations, Kluwer, 1996*

[Brown et al. 93] *P.F.Brown, S.A.D. Pietra, V.D.J. Pietre, P.L. Mercer*- The mathematics of machine translation: Parameter estimation - *Computational Linguistics, vol. 19, pp. 263-312, 1993.*

[Bruza 94] *Bruza P. D, IJdens J. J.* - Efficient Probabilistic Inference through Index Expression Belief Networks.- *Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence (AI94), pages 592--599. World Scientific, 1994.*

[Bruza et al 93a] *Bruza P.D, van der Gaag L.C* – Index Expression Belief Networks for Information Disclosure - *International Journal of Expert SystemsVolume 7 , Issue 2 1994, Pages: 107 - 138*

1994 .

[Bruza et al. 93b] *Bruza P.D. , van der Gagg L.C.*– Efficient Context-Sensitive

Plausible Inference for Information Disclosure – *ACM SIGIR'93, 1993.*

[Campos 00] *Luis M. de Campos, Juan M. Fernandez et Juan F. Huete* – Building Bayesian Network-Based Information Retrieval System – *Proceedings of the 11th International Workshop on Database and Expert Systems Applications (DEXA'00)- 2000.*

[Chevallet et al. 01] *Jean-Pierre Chevallet, Hatem Haddad* - Proposition d'un modèle relationnel d'indexation syntagmatique : mise en œuvre dans le système IOTA - *INFORSID 2001, Genève-Martigny, pp. 465, 483, 2001.*

[Chevallet 04] *Jean-Pierre Chevallet* - X-IOTA Une plate-forme distribuée ouverte pour l'expérimentation en Recherche d'Information*, in COnférence en Recherche Information et Applications CORIA'2004, Toulouse, 10 - 12 mars, 2004.*

[Chevallet et al. 04] *Gilles Sérasset, Jean-Pierre Chevallet* – Using surface-syntactic parser and Derivation from Randomness X-IOTA IR system used for CLIPS Mono & Bilingual Experiments for CLEF 2004 – *proceedings of CLEF 2004, 2004.*

[Croft 93] *Croft, W. B.*- Knowledge-based and statistical approaches to text retrieval- *IEEE Expert 8(2), 1993.*

[Davis et al. 97a] *Mark W. Davis, William C. Ogden* – Implementing Cross-Language Text Retrieval Systems for Large-scale Text Collections and the Word Wide Web –*Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 1997.*

[Davis et al. 97b] *Davis Mark W., Ogden William C.* – Free Ressources and Advanced Alignment for Cross-Language Text Retrieval – *Proceedings of the Sixth Text REtrieval Conference (TREC-6), 1997.*

[Deerwester et al. 90] *S. Deerwester, S.T. Dumais, G.X.Furnas, T.K. Landauer, R. A. Harman*- Indexing by latent semantic analysis - *Journal of the American Society for Information Science, 41(6):391-407, 1990.*

[Dumais et al. 96] *Susan T. Dumais, Thomas K. Landauer, Michael L. Littman* – Automatic Cross-Lingustic Information Retrieval using Latent Semantic Indexing – *Proceedings of SIGIR'96, 1996.*

[Federico et al. 02] *M. Federico, N. Bertoldi* – Statistical Cross-language Information Retrieval using N-Best Query Translations – *SIGIR'02, 2002.*

[Gao et al. 02] *Jianfeng Gao, Jian-Yun Nie, Hongzhao He, Weijun Chen, Ming Zhou* – Resolving Query Trnaslation Ambiguity using a Decaying Co-occurrence Model and Syntactic Dependence Relations- *ACM*

SIGIR 02, 2002.

[Greiff et al. 03] *Warren R. Greiff and William T. Morgan* – Contributions of Language modeling to the Theory and Practice of Information Retrieval – *Kluuwer Academic Publishers,* Language Modeling for Information Retrieval*, 2003.*

[Hiemstra 01] *Djoerd Hiemstra. Using Language Models for Information Retrieval – Ph.D. Thesis at Unviversity Twente Netherlands, 2001.*

[Hiemstra 99] *Djoerd Hemstra, Franciska de Jong* – Disambiguation Strategies for Cross-Language Information retrieval – *S. Abitboul, A.-M. Vercoustre (Eds): ECDL'99, LNCS 1696, pp. 274-293, 1999.*

[IJden 95] *IJdens J.J., Bruza P.D., Harper D.J.* – Probabilistic Inference Experiments using the ÉCLAIR Framework – *Technical Report No. 95/7, Faculty of science and technology, The Robert Gordon University, 1995.*

[Jacquemin 00] *Christian Jacquemin et Pierre Zweigenbaum* – Traitement automatique des langues pour l'accès au contenu des documents – *Jacques Le Maître, Jean Charlet et Catherine Garbay, éditeurs,* Le document en sciences du traitement de l'information*, chapitre 4, pages 71-109, Cepadues, Toulouse, 2000.*

[Jones et al. 03] *Karen Sparck Jones, Stephen Robertson, Djoerd Hiemstra, Hugo Zaragoza* – Language Modeling and Relevance - – *Kluuwer Academic Publishers, Language* Modeling for Information Retrieval*, 2003.*

[Lafferty et al. 01] *Laffertyn J., Zhai C.* – Document language models, query models, and risk minimization for information retrieval – *SIGIR 2001, pp. 111 – 119.*

[Lai 02] *Lại Thị Hạnh* – Trích cụm danh từ tiếng Việt nhằm phục vụ các hệ thống tra cứu thông tin đa ngữ - *Luận văn cao học ĐH. KHTN, 2002.*

[Lamas 98] *Mounia Lamas* – Logical models in Information Retrieval: Introduction and Overview- *Information Processing and Management: an International Journal Volume 34, Issue 1, pp. 19 – 33, 1998.*

[Lavrenko et al. 01] *Lavrenko V., Croft W. B.* – Relevance based language modèle – *SIGIR 2001, pp. 120-127.*

[Lavrenko et al. 02] *Victor Lavrenko, Martin Choquette, W. Bruce Croft* – Cross-Lingual Relevance Models – *AMC SIGIR'02, 2002.*

[Lovin 68] *Judith Beth Lovins*- Development of a stemming algorithm.

Translation and computational linguistics, 11(1):22-31, 1968.

[Miller et al. 99] *Miller D.R.H., Leek T., Schwart R. M.* – A Hidden Markov Model Information Retrieval System- *Hearst et al. 99, pp. 214-21.*

[Neapolitan 04] *Richard E. Neapolitan* – Learning Bayesian Networks – *Prentice Hall series in Artificial Intelligence, 2004.*

[Nguyen Than 97] *Nguyễn Kim Thản* – Nghiên cứu ngữ pháp tiếng Việt – *Nhà xuất bản khoa học xã hội, 1997.*

[Nguyen Quynh 01] *Nguyễn Hữu Quỳnh* – Ngữ Pháp Tiếng Việt – *Nhà xuất bản từ điển bách khoa, 2001.*

[Nguyen Hai 97] *Van Be Hai Nguyen* – A Vietnamese-Based Information Retrieval System – *Master thesis, Royal Melbourne Institute of Technology – Autralia, 1997.*

[Nie 90] *Jian-Yun Nie.* Modèle logique pour la Recherche d'information – *Thèse à l'université Joseph Fourrier- Grenoble I, 1990*

[Nie 00] *Jian-Yun Nie*- CLIR as Query Expansion as Logical Inference- *Workshop on Mathematical/Formal Methods in Information Retrieval (MF/IR) July 28, 2000 Athens, Greece-SIGIR 2000.*

[Nie 02] *Jian-Yun Nie* – Torwards a Unified Approach to CLIR and Multilingual IR – *SIGIR 2002, 2002.*

[Nie et al. 99] *J.Y. Nie, P. Isabelle, M. Simard, R. Durand* – Cross-language information retrieval base on parallel texts and automatic mining of parallel texts from the Web – *ACM-SIGIR conference, Berkeley, CA, pp. 74-91, 1999.*

[Nie 04] *J.Y.Nie* – Cross-lingual and Multi-lingual IR – *CORIA'04, Toulouse – France , 2004*

[Oard 96] *Douglas W. Oard et Bonnie J. Dorr* – A survey of Multilingual Text Retrieval – Technical Report *UMIACS-TR-96-19 CS-TR-3615, April 1996.*

[Oard 98] *Douglas W. Oard et Paul Hackett:* Document Translation for Cross-Language Text Retrieval at the University of Maryland. *TREC 1997: 687-696.*

[Patrick 91] *Person Patrick* - Les reseaux bayesiens : un nouvel outil de l'intelligence artificielle – *Thèse de Paris 6, 1991.*

[Pédauque 04] *Roger T. Pédauque, STIC-CNRS - Working paper - 8 juillet 2003. Version du texte, corrigée, enrichie et amendée, provisoirement définitive. Accessible en ligne : http://rtp-doc.enssib.fr/pedauque/pedauque.html.*

[Pirkola et al. 01] *Ari Pirkola, Turid Helund, Heikki Keskustalo, Kalervo Javelin* – Dictionary-Based Cross-Language Information Retrieval: Problems, Methodes, and Research Findings – *Kluwer Academic Publishers, Information retrieval , 4, 209-230, 2001.*

[Porter 80] *M.F. Porter* – An algorithm for suffix stripping. *Program, 14:130-137,1980.*

[Rabiner 90] *L.R. Rabiner* – A tutorial on hidden Markov models and selected applications in speech recognition- *A. Waibel and K. Lee, editors,* Readings in Speech Recognition*, pages 267-269. Morgan Kaufmann, Los ATlos, CA, 1990.*

[Ramshaw et al. 95] *L.A. Ramshaw, M.P. Marcus* – Text chunking using transformation-based learning – *Proceedings of Natural language Processing using Very Large Corpora, 1995.*

[Ribeiro et al. 96] *Berthier A.N. Ribeiro et Richard Muntz* – A Belief network Model for IR – *SIGIR'96.*

[Salton 70] *Gerard Salton* - Automatic processing of foreign language document – *Journal of the American Society for Information Science, 21(3):187-194, May 1970.*

[Salton et al. 83] *Salton Gerald et McGrill Michael J.*- Introduction to Modern Information retrieval – McGraw-Hill*, Jan. 1983.*

[Sanderson 94] *Mark Sanderson* – Word Sense Disambiguation and Information Retrieval – *ACM SIGIR 94, pp. 152-161, 1994.*

[Sang 02] *E.F.T.K. Sang* - Memory-based shallow parsing – *Journal of Machine Learning Research, 2002.*

[Sheridan et al. 96] *Paraic Sheridan, Jean Paul Ballerini* – Experiments in Multilingual Information retrieval using the SPIDER system – *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 58-65.*

[Song et al. 99] *Song F., Croft W. B.* – A general language model for information retrieval - *Hearst et al. 99, pages 270-280.*

[Sowa 84] *John F. Sowa* – Conceptual structures: Information Processing in Mind and Machine - *Addison Wesley, 1984.*

[Turtle 91] *Howard Robert Turtle* – Inference Network for Document retrieval – *Ph.D. thesis 1991.*

[van Rijsbergen 79] *Van Rijsbergen C.J.* – Information retrieval – *second edition, London, 1979.*

[Wong et al. 95] *S.K.M.Wong et Y.Y. Yao* – On Modeling Information Retrieval with Probabilistic Inference – *ACM Transactions on Information Systems, Vol. 1 », No 1, January 1995, pp. 36-68.*

[Xu et al. 01] *Xu J., Weischedel R., Nguyen C.*- Evaluating a probabilistic model for cross-langual information retrieval- *Croft et al. 2001, pages 105-110, 2001.*

Articles publiés

[Ho 04] *Bao-Quoc HO, Jean-Pierre CHEVALLET, Marie-France BRUANDET* - Recherche d'Information Bilingue français-vietnamien - *in Recherche Informatique Vietnam & Francophonie (RIVF'04), 02-05 Février 2004, Hanoi, Vietnam, 2004.*

[Ho 03] *HO Bao-Quoc, DONG Thi Bich Thuy* - Ứng dụng xử lý ngôn ngữ tự nhiên trong hệ tìm kiếm thông tin trên văn bản tiếng Việt - *Hội thảo Quốc gia về Công Nghệ Thông Tin - Thái Nguyên - Việt Nam 8-2003.*

[Ho 03] *Bao-Quoc HO, Jean-Pierre CHEVALLET, Marie-France BRUANDET*- Mise en place d'un Système de Recherche d'Information en vietnamien - *in TALN 2003 Traitement Automatique des Langues Naturelles, Atelier "multilinguisme", Batz-sur-Mer, France, 11-14 juin, 2003.*

[Ho 00] *Marie-France BRUANDET, Jean-Pierre CHEVALLET, Thi Bich Thuy DONG, Bao-Quoc HO.* An Introduction to Vietnamese Information Retrieval - *Workshop PAPILLON 2000.*