RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDELHAMID IBN BADIS DE MOSTAGANEM

Faculté des Sciences Exactes d'Informatique
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en informatique

Option : Ingénierie des Systèmes d'information

Thème

EVALUATION D'UN SYSTEME DE RECHERCHE AD-HOC A BASE DE MODELE DE THEMES LATENTS.

Réalisé par

M^lle^. Zohra Benzineb

M^lle^. Siham Mimouni

Encadré par

Mr. Abderrezak BRAHMI

Promotion : 2011-2012

DEDICACES

Avant tout, je dois rendre grâce à dieu de m'avoir donné le courage de terminer ce travail.
Je dédie ce travail à ma famille

Tout d'abord à mon cher père ,A ma chère mère, aux près de qui je trouve le réconfort et le repos, j'espère être à la hauteur de ses espérances et ne jamais la décevoir.

A mes frères et mes sœurs

A toute la famille Benzineb

A, Siham plus que mon binôme t'a été mon amie, je suis très contente d''avoir comme binôme.

Et tous mes chère amies qui sont m'aidé et m'encouragé.

A mes camardes de l'institut, compagnons de ces années d'études, j'espère qu'on restera camarade.

Zohra

Dédicace

Avant tout, je dois rendre grâce à dieu de m'avoir donné le courage de terminer ce travail.
Je dédie ce travail à ma famille

Tout d'abord à mon père.

A ma mère, au près de qui je trouve le réconfort et le repos, j'espère être à la hauteur de ses espérances et ne jamais la décevoir.

A mes frères : Mohamed, Ibrahim, Yahyahocine(Miloud), mes sœurs Samia, Chaima, HalimaAssadia

A tout la famille Belkacemi et tout la famille Mimouni.

A, Zohra plus que mon binôme t'a été mon amie, je suis très contente d''avoir comme binôme.

Et mes amies Karima, Siham.B, Fatima, Hafida, Fatima.Z, Houria, Fatiha, Nadia à tous je dis merci, de m'avoir réconforté, encouragé et aidé les jours où j'en ai eu besoin.

A mes camardes de l'institut, compagnons de ces années d'études, j'espère qu'on restera camarade.

Siham

REMERCIEMENTS

Nous remercions DIEU pour la science qu'il nous a enseignée et la patience qu'il nous a donnée.

Ce travail n'aurait pas vu le jour sans le concours et la contribution de bon nombre de personnes de près ou de loin.

Nous avons le plaisir de remercier :

-Notre encadreur M r
Abdarrasak Brahmi, pour son encadrement bienveillant, ses critiques toujours constructives, sa simplicité et sa disponibilité, il a été notre soutien moral et notre mentor, nous avons souvent été découragés, tendus fatigués prêts à jeter l'éponge mais vous étiez là à nous encourager et à nous pousser à relever le défi. Merci.

Chef département d'informatique, Aux membres de jurés qui nous ont fait l'honneur d'examiner notre travail

A tous nos enseignants qui nous ont enseignés du primaire jusqu'à ce jour là, leur savoir et leur expérience, nous les en remercions vivement et leur demandons de continuer leur noble tâche.

Que DIEU vous bénisse tous…

SOMMAIRE

LISTES DES FIGURES

LISTE DES TABLES

INTRODUCTION GENERALE

Grâce à l'évolution rapide des moyens d'information et de communication, une masse importante d'information est produite tous les jours à travers des milliers de livres et d'articles de journaux. Cette forte augmentation quantitative d'information à gérer et à consulter ne peut plus se contenter des méthodes et des techniques classiques de stockage et de consultation.

Dans ce contexte, la Recherche d'Information (RI) concerne les méthodes et mécanismes qui permettent la création et l'utilisation d'une base d'information. Une base d'information est un système documentaire permettant d'exploiter une collection de documents. La gestion concerne principalement le stockage des documents, ainsi que leur recherche et leur présentation en vue d'une utilisation ultérieure.

Afin de faciliter l'accès à l'information, plusieurs tâches d'analyse et d'organisation automatique des documents peuvent être affiliées à la RI (classification selon des catégories prédéfinies, agrégation ou clustering, sommairisation, système de question-réponse, …etc). Mais la tâche la plus répandue en RI et celle de la recherche simple qui consiste à répondre, par une liste des documents pertinents, à un besoin en information exprimé par une requête libre en langage naturel. Cette tâche est souvent dénotée par la recherche Ad-Hoc.

Un Système de recherche d'information (SRI) est un ensemble logiciels permettant d'effectuer l'ensemble des tâches nécessaires à la RI, en particulier la recherche Ad-Hoc. Un SRI possède trois fonctions fondamentales qui définissent le modèle de recherche : représenter le contenu des documents, représenter le besoin de l'utilisateur et comparer ces deux représentations. La représentation des documents et de la requête dans le système se fait à l'issue d'une phase appelée indexation qui consiste à choisir les termes représentatifs des documents et à les ajouter à un index qui à chaque terme associe le document dans lequel il se trouve avec éventuellement des informations additionnelles comme la fréquence d'apparition du terme.

Un modèle de recherche doit mettre en correspondance les représentations des documents et la représentation du besoin de l'utilisateur exprimé sous la forme d'une requête afin de retourner à celui-ci les documents en rapport avec sa requête. Généralement, cela se fait à

l'aide d'un calcul de similarité. L'opération de comparaison des représentations est fondamentale en RI. Elle constitue le cœur du modèle de recherche. Les modèles de recherche s'appuient sur des théories mathématiques qui offrent des opérations pour comparer les représentations des documents de la collection et la représentation de la requête de l'utilisateur.

Le but de notre mémoire est de réaliser et d'évaluer l'intégration d'un modèle à base de thèmes, notamment celui de l'allocation latente de Dirichlet (LDA), dans un système de recherche Ad-Hoc. Nous abordons dans les trois premiers chapitres les aspects théoriques relatifs à notre sujet avant de décrire dans les deux derniers la conception, l'implémentation et les résultats des expérimentations effectuées.

Dans le premier chapitre, nous introduirons les concepts de base de la RI au travers de la description du processus de RI. Ce processus consiste en l'acquisition des données que constituent les documents et la requête. Ces données textuelles subissent une série de traitements afin de construire un index. Une fois l'index construit, les représentations des données issues de l'index peuvent être comparées afin de déterminer la liste des documents qui sont pertinents pour la requête. Enfin nous présenterons les collections et les méthodes d'évaluation des SRI.

La méthode de représentation et plus encore la méthode de comparaison des représentations sont au cœur des modèles de RI. C'est pourquoi nous décrirons les principaux modèles de RI dans le deuxième chapitre.

Dans le troisième chapitre, nous introduirons les définitions et notations usuelles concernant les modèles probabilistes à thème latent en s'intéressant de près au modèle LDA. Nous décrivons, dans le quatrième chapitre, notre contribution par la conception et l'implémentation d'un système d'évaluation de la recherche ad-hoc selon différents modèles. Les résultats obtenus, des différentes expérimentations pour une variété de collections, seront dressés dans le dernier chapitre.

CHAPITRE I :
LA RECHERCHE D'INFORMATION

1 Introduction

La recherche d'information RI ou IR (*Information Retrieval* en anglais) [Frakes 1992] [Grossman et al 1998] [Salton 1970] [Yates et al 1999] est l'ensemble des techniques permettant de gérer des textes. C'est la science qui étudie la manière de répondre pertinemment à une requête en retrouvant l'information dans un corpus. Celui-ci est composé de documents d'une ou plusieurs bases de données, qui sont décrits par un contenu ou les métadonnées associées. Le contenu des documents peut être du texte, des sons, des images ou des données. La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie qui visent a représenter des documents dans le but d'en récupérer des informations, au moyen de la construction d'index. L'informatique a permis le développement d'outils pour traiter l'information et établir la représentation des documents au moment de leur indexation, ainsi que pour rechercher l'information.

La Recherche d'Information (RI) est aussi un domaine de recherche en informatique qui s'attache à définir des modèles et des systèmes dont le but est de faciliter l'accès à un ensemble de documents sous forme électronique (corpus), afin de permettre à un utilisateur de retrouver ceux qui sont pertinents pour lui, c'est-à-dire ceux dont le contenu correspond le mieux à son besoin d'information à un moment donné. Le domaine de la RI est donc centré autour des utilisateurs, notamment par le biais de la notion de *pertinence*, c'est-à-dire l'adéquation pour l'utilisateur entre le contenu d'un document et l'information recherchée.

Avec l'émergence des réseaux électroniques et le développement du Web, la quantité d'information disponible pour les utilisateurs est sans cesse croissante, et la RI doit faire face à de nouveaux défis d'accès à l'information, à savoir retrouver une information dans un espace diversifié et de taille considérable.

2 Système de recherche d'information (SRI)

2.1 Définition

Un système de recherche d'information intègre un ensemble de techniques et de processus permettant de sélectionner dans une collection de documents ceux qui sont susceptibles de répondre au besoin d'un utilisateur. Ces processus permettent :

– la représentation des informations et des besoins,

– l'interrogation, la recherche et la sélection des informations pertinentes répondant aux besoins d'un utilisateur.

La problématique majeure émanant de tout système de recherche d'information est de retrouver les quelques dizaines ou milliers de documents pertinents parmi des millions de documents. Cet écart de cardinalité rend cette tâche encore plus difficile.

2.2 Concepts clés

Un système de recherche d'information, intègre un ensemble de modèles pour la représentation des unités d'informations (documents et requêtes). Il intègre également un mécanisme de recherche/sélection. Ce dernier permet de sélectionner l'information pertinente en réponse aux besoins exprimés par l'utilisateur à l'aide d'une requête. Il peut être représenté par le processus en U de recherche d'information [Belkin et al 1992]. La figure (1.1) illustre l'architecture générale d'un système de recherche d'information qui est composé de plusieurs fonctions :

- L'indexation des documents.
- La mise en correspondance (l'appariement) requête-documents avec un ordonnancement des documents quand le modèle le permet.
- La restitution des documents reconnus pertinents par rapport à la requête.

Du point de vue utilisateur, le processus de recherche est composé de deux fonctions principales :

- L'interrogation du système par une requête
- L'analyse des documents restitués par le système et une éventuelle reformulation de requête.

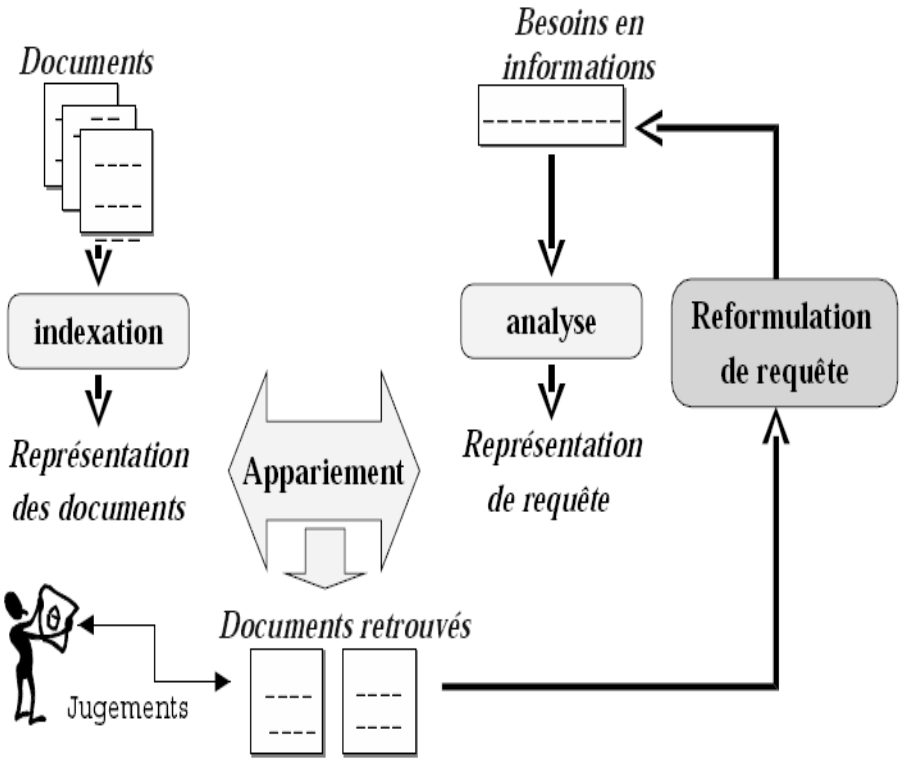


Figure1.1 : Processus en U de la RI

3 La collection de documents (corpus)

La collection de documents constitue l'ensemble des informations exploitables et accessibles par l'utilisateur. Elle est constituée d'un ensemble de documents. Dans le cas général et pour des raisons d'optimalité, la collection constitue des représentations très simplifiées mais suffisantes de ces documents. Ces représentations sont étudiées de telle sorte que la gestion ajout, suppression d'un document et l'interrogation (recherche) de la collection se font dans les meilleures conditions de coût.

3.1 Le document

En se penchant sur l'histoire sémantique du terme *document*, on peut voir que différents sens lui ont été associés au fil du temps, « Document » vient du latin *Documentum* qui signifie leçon, exemple, modèle et secondairement preuve, document, texte. Ce terme partage la même racine que doctrine ou docteur; ces termes sont de la famille du verbe « docere » qui signifie instruire, enseigner [Vignaux, 2003]. Le sens principal du terme « document » est enseignement, jusqu'au 18ème siècle où il devient « preuve ». Sa définition change au 19ème : « Chose qui enseigne ou renseigne ; titre, preuve. Un document précieux. Les documents

font défaut pour établir ce point d'histoire » [Littré]. Une définition est adoptée pour trois langues (Allemand, Anglais, Français) en 1935 : « Document : Toute base de connaissance, fixée matériellement, susceptible d'être utilisée pour consultation, étude ou preuve. Exemples: manuscrits, imprimés, représentations graphiques ou figurées, objets de collections, etc. »

Les idées générales que l'on peut extraire de ces définitions sont qu'un document est une trace d'activité humaine, trace laissée dans l'objectif d'être interprétée par des personnes souvent différentes du ou des personnes à l'origine de cette même trace. En conséquence, on peut voir le document comme une chose porteuse de sens pour un auditoire donné. Son contenu s'exprime en une forme interprétable pour quelqu'un. Avec l'avènement des ordinateurs, le document quitte son support matériel natif (le papier pour l'écrit et les bandes analogiques pour les documents audiovisuels) et devient numérique. Il est alors stockable sous la forme d'une représentation binaire donc l'ensemble des documents sur lesquels portera la recherche est stocké dans une banque de données (sur le Web). Un document est le type d'objet de base géré par le système.

3.2 Le besoin en information

Un besoin en information d'un utilisateur est exprime par une requête. La délimitation du besoin est peut-être l'une des tâches les plus difficiles de la recherche d'information. Elle est seule garante de la rapidité et de l'efficacité d'une recherche. Le besoin d'information est une notion complexe. La prise de conscience de ce besoin nécessite que l'on ait, paradoxalement des connaissances [Tricot, 2003 ; Pochet et Thirion, 2005]. Les personnes ayant très peu de connaissances sur un sujet donné recherchent moins d'informations sur ce sujet que celles qui en possèdent déjà au départ. Elles ont beaucoup plus de mal à se rendre compte de leur ignorance. En effet, pour bien préparer sa recherche il est nécessaire de définir ce besoin d'information, de cerner son sujet, d'en identifier les concepts principaux, de préciser les questions destinées à guider la recherche, d'identifier les bons mots clés correspondant le mieux à la recherche, de rejeter les termes non significatifs et finalement de définir le but de son travail. Ces diverses opérations sont difficiles à mettre en oeuvre sans une certaine expertise de la recherche d'information mais surtout du sujet de recherche lui-même. Cette connaissance est aussi nécessaire pour reconnaître les informations fiables et pertinentes.

3.3 La requête

La requête est créée par l'utilisateur, c'est elle qui initie le processus de recherche. Elle traduit un besoin d'information, c'est-à-dire une nécessité ressentie de combler une déficience

constatée en information, une lacune ou un défaut. C'est une situation problématique qui amène l'utilisateur à formuler une requête [Schutz et al., 1973]. La requête doit contenir les concepts clés du besoin et les relations entre ces concepts. Elle est issue d'une analyse conceptuelle du besoin d'information qui est effectuée dans l'esprit de l'utilisateur de façon plus ou moins précise. En effet, l'utilisateur fait face à un « problème de vocabulaire » quand il tente de traduire son besoin d'information en une requête. Les mots sont souvent polysémiques et les concepts peuvent être décrits par un grand nombre de mots.

Les travaux de [Kuhlthau et al., 1990] montrent que le besoin d'information de l'utilisateur devient plus clair et plus précis au cours du processus de recherche. De façon similaire, les sentiments d'incertitude, de frustration et de confusion présents au début de la recherche décroissent au fur et à mesure que la recherche progresse. D'autres travaux approfondissent la notion de besoin d'information [Dervin, 1983] [Green, 1990] [Kuhlthau, 1993] du point de vue cognitif. Une fois formulée la requête peut avoir la forme d'une expression en langue naturelle, ou encore d'une liste de concepts avec éventuellement un degré d'importance associé, ou encore une formule logique de concepts coordonnés par des opérateurs logiques. Une fois la requête exprimée, il est nécessaire de lui donner une forme utilisable par un SRI pour entamer le processus de recherche.

4 Fonctionnement d'une recherche Ad-Hoc

4.1 Représentation des documents et des requêtes (indexation ou analyse)

La représentation des documents et des requêtes est supportée par un ensemble de règles et notations permettant la traduction d'une requête ou d'un document d'une description brute vers une description structurée. Ce processus de conversion est appelé Indexation. L'indexation est une opération permettant d'extraire d'un document ou d'une requête une représentation paramétrée qui couvre au mieux son contenu sémantique. Le résultat de l'indexation constitue le descripteur du document ou de requête. Ce dernier est souvent une liste de termes ou groupe de termes significatifs pour l'unité textuelle correspondante, généralement assortis de poids représentant leur degré de représentativité du contenu sémantique de l'unité qu'ils décrivent. Les descripteurs des documents (mots, groupe de mots) forment le langage d'indexation. L'indexation est une étape primordiale dans la recherche d'information. De sa qualité dépend en partie la qualité des réponses du système. Conscients de son importance, et soucieux de bien la réaliser, les développeurs des SRI ont

proposé plusieurs manières de procéder l'indexation, Les principales sont l'indexation manuelle et l'indexation automatique. Elles sont définies comme suit :

– **Indexation manuelle :** dans le cas de l'indexation manuelle, chaque document est analysé par un spécialiste du domaine ou par un expert documentaliste. En fonction de ses connaissances, Cet expert détermine les mots clés qui lui semblent les plus significatifs pour représenter le document. L'indexation humaine est une activité fondée sur le jugement d'un être humain. Elle se caractérise par sa profondeur, sa cohérence (ce qui est fondamental pour la cohérence du fond et des fichiers) et sa qualité (exhaustivité -spécificité). Elle est cependant trop dépendante de l'état des connaissances des indexeurs. Cela induit à la subjectivité de ses résultats. Elle nécessite la lecture de l'intégralité des documents. Son application est de ce fait inadaptée à des collections de taille importante. Les collections TREC 1 constituent un exemple significatif. Elles contiennent des millions de documents extraits d'Internet (le web). L'indexation automatique permet de pallier à ce problème.

– **Indexation automatique :** l'indexation automatique reconnaît des chaînes de caractères constitutives de mots non vides. Elle détecte automatiquement les termes les plus représentatifs du contenu du document. Ce type d'indexation est actuellement la méthode la plus répandue. Elle comprend deux étapes fondamentales : l'identification des termes d'indexation et l'évaluation de leurs poids. L'identification des termes d'indexation consiste à analyser le texte du document mot à mot. Son objectif est d'en extraire les mots vides qui ne jouent qu'un rôle syntaxique. Ces mots sont identifiés puis éliminés grâce à un anti-dictionnaire (*stoplist* en Anglais). Les mots apparaissant trop souvent n'ont aucun intérêt. Ils sont également éliminés. Seuls les mots significatifs représentant les concepts du document sont retenus.

4.2 L'appariement requête-document

Le processus d'appariement requête-document est le noyau d'un système de recherche d'information. Il permet d'associer à chaque document une valeur de pertinence vis à vis d'une requête. Les documents ayant une pertinence positive sont sélectionnés. La mesure de pertinence est calculée à partir d'une fonction de similarité, notée ***RSV(Q,d)*** (*Retrieval Sattus Value*), *Q* étant une requête et *d* un document. Elle tient compte des poids des termes déterminés en fonction d'analyses statiques et probabilistes. Notons que ce processus est étroitement lié aux représentations des documents et des requêtes. En effet, si l'opération

[1] T*ext Retrieval Conference, http://trec.nist.gov*

d'indexation est la même dans la plupart des modèles de recherche d'information, ces derniers diffèrent souvent par rapport aux fonctions utilisées pour la mesure des poids et pour l'appariement requête-document.

La notion de pertinence est souvent difficile à appréhender, on parle souvent de deux types de pertinence :

- la pertinence utilisateur : le document est jugé pertinent par l'utilisateur en fonction de son besoin en information,
- la pertinence système : le document est jugé pertinent par le SRI pour une requête sur la base de la fonction de pertinence.

4.3 Reformulation automatique de la requête

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

Cette reformulation intervient dans un processus plus général d'optimisation de la fonction de pertinence. Celle-ci a pour but de rapprocher la pertinence système de la pertinence utilisateur. Elle se présente comme une opération primordiale dans un SRI. La reformulation de requête par injection de pertinence est plus connue sous le nom de Relevance feedback [Boughanem et al 1999] [Buckley et al 1995] [Morita et al 1994] [Tamine et al 2000] est le processus le plus utilisé (aussi utilisé pour l'indexation [Fluhr et al 1990]). Cette méthode permet une modification de la requête initiale, sur la base des jugements de pertinence de l'utilisateur sur les documents restitués par le système.

5 Evaluation des SRI

Depuis la naissance du domaine de la RI, l'évaluation des systèmes de recherche d'information constitue une étape importante dans l'élaboration d'un modèle de recherche d'information. En effet, elle permet de caractériser le modèle et de fournir des éléments de comparaison entre modèles. D'une façon générale, tout système de recherche d'information présente deux objectifs:

– retrouver tous les documents pertinents,

– rejeter tous les documents non pertinents.

Pour réaliser une telle évaluation, une expérimentation qui utilise les éléments suivants doit être établie:

– Un ensemble de documents.

- Un ensemble de requêtes.

- La liste de documents pertinents pour chaque requête (jugements de référence).

- Des mesures et des critères quantifiables.

5.1 Notions de bases

Les systèmes de recherche d'information SRI sont toujours évalués en fonction de la pertinence des documents retrouvés dans une collection ***C*** qui peut être répartie en 3 sous ensembles : Ensemble des documents trouvés pertinents pour le système ***R*** ; Ensemble des documents convenant à l'utilisateur ***P*** ; Ensemble des Documents restituer retrouvés et convenant à l'utilisateur, pertinents pour le système et l'utilisateur ***P∩R*** ; les documents non-trouvés et non-pertinents ***N***.

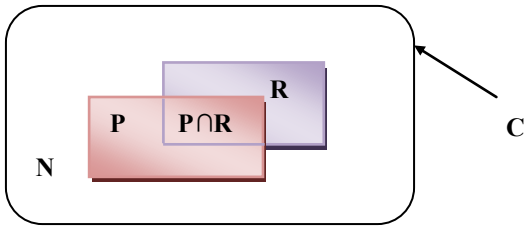


Figure1.2 : les sous-ensembles d'une collection de documents dans un SRI.

Pour mesurer l'efficacité de la recherche d'information ad-hoc, nous avons besoin d'une collection de test constitué de :

1. Une collection de documents.
2. Un ensemble de requêtes qui expriment des besoins d'information.
3. Un ensemble de jugements pertinents indiquent les documents pertinents.

5.2 Les métriques principales

Un système est évalué en comparant les réponses du système pour toutes les requêtes de la collection de tests avec les documents pertinents. L'objectif est d'estimer la

correspondance entre la pertinence utilisateur et la pertinence système. En raison des imprécisions inhérentes à la formulation du besoin de l'utilisateur, à la représentation interne des documents et à la mise en correspondance, ces deux pertinences ne sont pas confondues. La comparaison des réponses d'un système pour une requête avec les réponses idéales permet d'évaluer les métriques principales (la précision et le rappel F-mesure et interpolation à11point) qui sont définies comme suit :

Le rappel [Salton 1971] mesure la capacité du système à retrouver tous les documents pertinents pour une requête, c'est-à-dire la proportion de documents pertinents retrouvés parmi tous les documents pertinents dans la base ou autrement dit le rappel est le rapport du nombre de documents pertinents trouvés par le système au nombre de documents pertinents disponibles.

$$\mathbf{R} = \frac{\textbf{nombre de documents pertinents retrouvés}}{\textbf{nombre de documents pertinents dans la base}} = \frac{P \cap R}{P}$$

Où: ***R*** est l'ensemble des documents retrouvés par le système pour la requête

P est l'ensemble des documents pertinents de la collection pour cette requête.

La précision [Salton 1971] mesure la capacité du système à ne retrouver que les documents pertinents pour une requête, c'est-à-dire la proportion de documents pertinents retrouvés parmi tous les documents retrouvés par le système.

$$\mathbf{P} = \frac{\textbf{nombre de documents pertinents retrouvés}}{\textbf{nombre de documents retrouvés}} = \frac{P \cap R}{R}$$

La précision moyenne est calculée par la précision de mesure aux points de rappel différents (disons 10%, 20%, et ainsi de suite) et la moyenne. [Xing Yi and James Allan, 2009]

F –mesure : [Van Rijsbergen, 1979] a introduit la F-mesure comme combinaison du rappel et de la précision. La F-mesure est définie à travers la formule suivante :

$$F = \frac{1}{\alpha\frac{1}{P} + \frac{(1-\alpha)1}{R}} = \frac{[(\beta]^2 + 1)PR}{\beta^2 P + R} \qquad \text{où} \quad \beta^2 = \frac{1-\alpha}{\alpha}$$

Où $\boldsymbol{\alpha} \in \mathbf{[0, 1]}$ et donc $\beta \in [0, \infty[$. Où **β** traduit l'importance relative du rappel et de la précision.

Par exemple, β=2 représente une précision deux fois plus importante que le rappel. Dans le cas particulier où β=1, la F-mesure définit la moyenne harmonique du rappel et de la précision:

$$F_{\beta=1} = \frac{2PR}{P+R}$$

L'interpolation à 11 points est la précision maximale trouvée dans un point de rappel $r' \geq r$ qui peut être formulé comme suit :

$$Pinterp(r) = \max_{r>r'} P(r')$$

Exemple : Le tableau 1.1 présente une liste de documents triée par ordre décroissant de pertinence, et les mesures de précision et de rappel engendrées. La figure 1.3 illustre la courbe rappel-précision. On calcule la précision pour chacune des valeurs de rappel 0.1, 0.2 . . . 1.0 par interpolation linéaire.

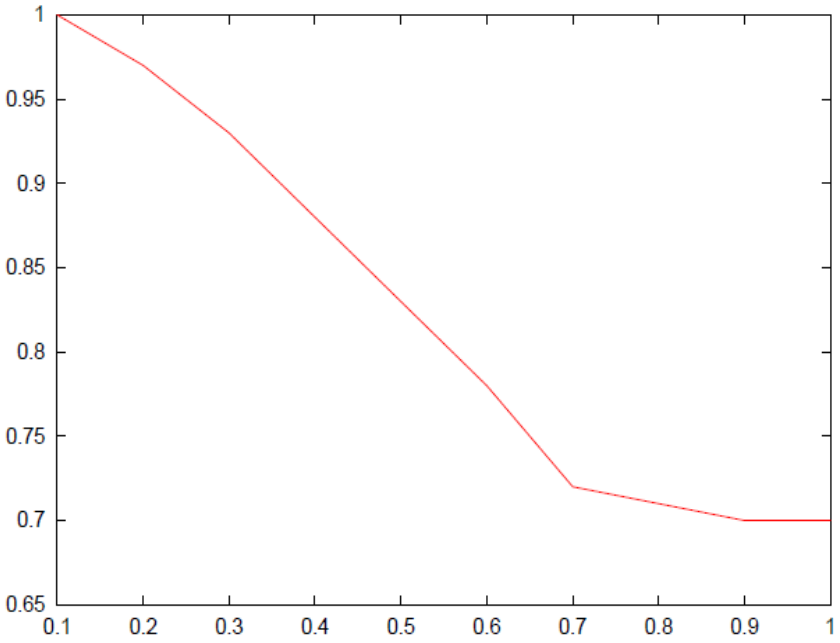


Figure1.3 : la courbe rappel-précision

Cette méthode d'évaluation est très significative. La précision mesurée indépendamment du rappel et inversement sont par contre peu significatives. En effet, un système même peu performant a de très fortes chances d'attribuer la plus grande valeur de pertinence à un document pertinent s'il sélectionne seulement ce document.

document	score	[pertinent]	Précision	Rappel
d1	9.45	*	1	0.14
d2	9.34		0.5	0.14
d3	8.72	*	0.67	0.29
d4	8.66	*	0.75	0.43
d5	7.93	*	0.8	0.57
d6	6.53		0.67	0.57
d7	6.41	*	0.71	0.71
d8	4.87		0.62	0.71
d9	4.87	*	0.67	0.86
d10	4.09	*	0.7	1
d11	3.82		0.64	1
d12	2.21		0.53	1
d13	2.18		0.54	1
d14	1.64		0.5	1
d15	0.02		0.47	1

Tableau 1.1 : exemple de valeurs rappel-précision

Un bon système donne de bons taux de précision et de rappel en même temps. Un système dont la précision vaut 1 pour toutes les valeurs de rappel est un système idéal qui retrouve tous les documents pertinents, et uniquement les documents pertinents – la pertinence système et la pertinence utilisateur sont alors confondues.

6 Corpus de test

L'évaluation des méthodes en RI nécessite un jeu de données (collection de documents) représentatif ayant des jugements de référence. Parmi les défis majeurs, pour développer l'approche expérimentale dans le domaine, était de mettre à la disposition des chercheurs des corpus standard appropriés aux tâches diversifiées de recherche d'information.

Le terme "corpus" peut faire référence à tout ensemble de textes sous forme numérique. Mais pour un linguiste ou toute autre personne impliquée dans la description générale de la langue, un corpus est un vaste échantillon représentatif illustrant le répertoire complet des types de textes dans une langue, comme les romans, le journalisme, l'écriture académique et de nombreuses autres variétés de texte [Evans, 2007].

Parmi les corpus existants pour réaliser l'évaluation on a ***AP89***, ***Genomics-04***, ***CACM*** (corpus Association of Computing Machinery) et ***CF*** (Cystic Fibrosis)

Le corpus TREC-AP89 [Harman, 1995] est une collection des dépêches de presse de l'*Associated-Press* collectées durant l'année 1989. Pour des raisons pratiques (temps de calcul et taille de la mémoire) seuls les 7466 premiers documents de la collection et les 50 premiers requêtes on été conservés.

La collection de documents **Genomics 2004 et 2005** contient un total de 4,591,008 documents. Pour la piste 2004, 50 sujets de test sont disponibles, avec une longueur moyenne de 7 termes.

6.1 Caractéristiques d'un corpus

La crédibilité des expérimentations relatives au traitement automatique de la langue naturelle dépend en partie de la nature des collections de test utilisées. L'évaluation des méthodes d'analyse automatique du texte, dont les tâches de RI en font partie, nécessite l'élaboration préalable de corpus appropriés respectant les critères suivants :

6.1.1 La taille

Le corpus doit évidemment atteindre une taille assez suffisante pour permettre des traitements statistiques fiables. Il est devenu primordial d'apprécier toute méthode d'analyse de texte dans deux configurations possibles : la première traite une collection de taille modérée (des centaines ou quelques milliers de documents). La deuxième concerne l'utilisation d'une collection à grande échelle (des dizaines de milliers voire des centaines de milliers et plus). La taille du document lui-même (nombre de mots ou de phrase) peut aussi être déterminante de la qualité du test dans certaines situations.

6.1.2 La représentativité

La représentativité se réfère à un échantillon de texte incluant la variabilité complète dans une population.

6.1.3 La forme lisible par la machine

De nos jours, le terme "corpus" implique souvent le sens "lisible par la machine". Ce ne fut pas toujours le cas, dans le passé le mot "corpus" n'a été utilisé en référence qu'au texte imprimé. Les données des corpus sont parfois disponibles dans d'autres formes de médias comme les microfiches (concordance complète de mot-clé du corpus LOB, corpus parlé d'anglais par Lancaster / IBM). La lisibilité numérique des corpus permet un traitement accéléré et facilite son enrichissement. La mise à disposition des routines normalisées d'accès aux données du corpus est une option facultative mais très sollicitée par les chercheurs.

6.1.4 La standardisation

Il est souvent souhaitable qu'un corpus constitue une référence standard pour la variété de langue (s) qu'il représente. Cela suppose que le corpus sera mis à la disposition d'autres chercheurs. Un avantage d'un corpus largement diffusé fournit une mesure à laquelle les études successives peuvent être comparées, tant que la méthodologie n'est pas changée, de

nouveaux résultats sur des sujets connexes peuvent être directement comparés avec les résultats déjà publiés sans avoir besoin de les recalculer. En outre, un corpus standard signifie aussi qu'une base de données continue soit utilisée, ceci implique que toute variation entre les résultats des études différentes serait à la cause de la pertinence des hypothèses et de la méthodologie, moins que la différence dans les données.

Parmi les autres caractéristiques on peut ajouter le temps couvert par les textes du corpus, car le temps joue un rôle important dans l'évolution du langage, par exemple le français parlé aujourd'hui ne ressemble pas au français parlé il y a 200 ans, ni au français parlé il y a 10 ans, à cause notamment des néologismes. C'est un phénomène à prendre en compte pour toutes les langues vivantes. Donc un corpus ne doit pas contenir de textes rédigés à des intervalles de temps trop larges.

Aussi, il ne faut pas non plus mélanger des catalogues différents de plusieurs domaines. Par exemple un corpus construit à partir de textes scientifiques ne peut être utilisé pour extraire des informations sur les textes littéraires, et un corpus mélangeant des textes scientifiques et littéraires ne permettra de tirer aucune conclusion crédible, il faut que le corpus soit homogène [Spousta, 2006].

6.2 Construction des corpus

Aujourd'hui, la plupart des corpus sont construit à partir du texte qui est déjà numérisé, le coût de mise en forme de texte électronique qui n'existe que sur papier est beaucoup plus cher que le coût d'une simple copie par téléchargement et la collection de données qui sont déjà numérisés.

Il y a plusieurs avantages à la création d'un corpus à partir des données sur le Web, plutôt que du texte imprimé. Toutes les données sur le Web sont déjà sous forme électronique et donc lisible par les machines. La grande quantité de texte disponible sur le Web (plus de 5 millions Téra octet) est un avantage majeur qui permet de collecter des quantités suffisante dans n'importe qu'elle domaine.

6.3 Techniques de Crawling

Afin de construire un corpus à partir du Web, plusieurs questions et problèmes techniques doivent être résolus. Tout d'abord, le Web Crawler doit être choisi par rapport au volume de données de texte que nous voulons obtenir. Pour certaines tâches, il suffirait de télécharger quelque milliers de documents, pour d'autres tâches, il faudrait une quantité énorme de

données. Les données brutes doivent passer par plusieurs filtres comme le recodage de caractères, détection de la langue, la conversion de format et l'extraction de texte. Le format cible doit être correctement sélectionné selon le style prévu pour l'accès aux données.

Il y a de nombreux défis dans la création d'un corpus Web. Comme le contenu du Web n'est pas structuré et ne possède pas de répertoire définitif, aucune méthode simple et directe n'existe pour rassembler un vaste échantillon représentatif depuis le Web. Une approche radicale consiste à explorer le Web à travers l'échantillonnage des adresses IP du réseau mondial (Internet). Cette méthode requiert des ressources considérables car de nombreuses tentatives sont faites pour chaque site Web trouvé. En répertoriant près de 16 millions de serveurs, une étude a signalé qu'un seul serveur Web a été trouvé sur 269 essais [Lawrence et Giles, 2000].

Nous résumons, dans ce qui suit, l'essentiel des approches d'exploration du Web pour en concevoir un corpus ou indexer son contenu :

Exploration en largeur d'abord : en visitant, à partir d'un ensemble d'adresses initiales, toutes les pages Web du premier niveau avant d'explorer le niveau suivant [Najork et Wiener, 2001].

Exploration du meilleur d'abord : en sélectionnant, dans une frontière de liens, les meilleures pages selon certains critères plus ou moins complexes [Cho et al., 1998].

Exploration du plus important (ou PageRank): en choisissant les pages ayant les plus grand scores de référencement récursive [Brin et Page, 1998].

Marche aléatoire : en échantillonnant un ensemble uniforme de nœuds dans un graphe du Web (ou dans une version simulée) [Henzinger et al., 2000].

Exploration thématique (ou ciblée) : en appliquant des heuristiques pour sélectionner un certain type de pages sur un sujet précis ou dans une langue particulière [Mencser et al., 2003].

7 Conclusion

Nous avons présenté dans ce chapitre les concepts fondamentaux de la RI. Le but d'un SRI est de rechercher l'information pertinente pour une requête utilisateur. Son efficacité est mesurée par des paramètres qui reflètent sa capacité à accomplir un tel but.

La réalisation des systèmes de recherche d'information nécessitent la jonction de plusieurs efforts tant conceptuels que pratiques. En premier lieu, le choix du modèle d'indexation et de recherche de document affecte directement la qualité des résultats obtenus. La prise en charge des aspects sémantiques et multilingues constitue deux autres grand défis pour la RI moderne. Nous présentons dans ce qui suit les modèles de base en RI avant de décrire ultérieurement les modèles de thème.

CHAPITRE II :
LES MODELES DE RECHERCHE D'INFORMATION

1 Introduction

Les modèles de recherche permettent de donner une interprétation des termes choisis pour représenter le contenu d'un document. Étant donné un ensemble de termes pondérés issus de l'indexation, le modèle remplit deux fonctions :

- La première est de créer une représentation interne pour un document ou pour une requête basée sur ces termes,
- La seconde est de définir une méthode de comparaison entre une représentation de document et une représentation de requête afin de déterminer leur degré de correspondance (ou similarité).

Le modèle joue un rôle central dans la RI. C'est lui qui détermine le comportement clé d'un SRI. Dans la suite nous présenterons d'abord le modèle booléen qui est historiquement un des premiers modèles étudiés et qui a servi de point de départ aux recherches du domaine puis le modèle vectoriel et enfin, le modèle probabiliste.

2 Le modèle booléen

Ce modèle est basé sur la théorie des ensembles. Un document est représenté par un ensemble de termes. La requête est représentée par un ensemble de mots clés reliés par des opérateurs booléens (AND, OR et NOT). L'appariement requête-document est strict et se base sur des opérations ensemblistes selon les règles suivantes :

$RSV(d, t_i) = 1$ si $t_i \in d$ 0 sinon (2.1)

$RSV(d, t_i \ AND \ t_j) = 1$ si $[\![t]\!]_i \in d) \wedge [\![t]\!]_j \in d)$, 0 sinon (2.2)

$RSV(d, t_i \ OR \ t_j) = 1$ si $[\![t]\!]_i \in d) \vee [\![t]\!]_j \in d)$, 0 sinon (2.3)

$RSV(d, NOT \ t_i) = si \ t_i \notin d, 0 \ sinon$ (2.4)

Ce type de recherche est à la base des moteurs de recherche comme Altavista1 ou Google. Ce modèle présente de nombreux avantages :

- Il est tout d'abord facile à implémenter et est tout à fait fonctionnel [Frakes et al., 1992].
- Il permet aux utilisateurs d'exprimer des contraintes structurelles et conceptuelles [Marcus, 1991].
- Il est très efficace pour des requêtes utilisant des termes très spécifiques ou portants sur des domaines techniques particuliers avec leur vocabulaire propre.

Les utilisateurs trouvent que l'utilisation de synonymes (grâce à la clause OR) et de groupes de mots (grâce à la clause ET) sont utiles pour la formulation de la requête [Cooper, 1988].

L'approche booléenne possède un grand pouvoir d'expressivité : elle est tout à fait adaptée aux requêtes qui appellent une sélection exhaustive et non ambigüe.

Enfin l'approche booléenne peut être tout à fait utile dans la fin du processus de recherche en raison de la clarté et de l'exactitude avec laquelle les concepts sont représentés. Son intérêt reste néanmoins limité. Il présente cependant les principaux inconvénients suivants :

Les formules de requêtes sont complexes, non accessibles a un large public, la réponse du système dépend de l'ordre de traitement des operateurs de la requête, les modèles de représentation des requêtes et documents ne sont pas uniformes. Ceci rend le modèle inadapté à une recherche progressive.

Le fonctionnement des opérateurs AND et OR pose des problèmes : le AND logique ne fait aucune différence entre deux cas pourtant distincts : si aucun terme ne satisfait la requête ou si tous sauf un satisfont la requête l'opérateur ne retrouve pas de document (Null Output problem). Symétriquement, un usage trop fréquent d'OR ramène trop de documents (Overload Output problem).

Le problème majeur de l'approche booléenne est que les documents qui répondent à la requête sont retournés dans un ordre quelconque, et sont tous identiquement similaires à la requête.

Les deux approches présentées ci-dessous sont largement utilisées en pratique, permettent de remédier à ces inconvénients.

Une extension de ce modèle a été effectuée par Fox et Salton [Fox 1983],[Salton 1971a]. Il s'agit du modèle booléen étendu. Ce dernier complète le modèle de base en intégrant des

poids dans l'expression de la requête et des documents. Ceci induit la sélection de documents sur la base d'un appariement rapproche (fonction d'ordre) et non exact.

Exemple La requête politique définit tout simplement les documents indexés avec le terme politique. En utilisant les operateurs de Boole, les requêtes et les ensembles de documents correspondants peuvent être combinés pour former de nouveaux ensembles de documents. Combiner deux termes avec l'opérateur ET définit un ensemble de documents plus petit que les ensembles de documents indexés par l'un de ces termes, par exemple, la requête politique et gouvernement produit les documents indexés a la fois par les termes politique et gouvernement. Combiner les termes avec l'operateur OU définit un ensemble de documents plus grand que les ensembles de documents indexés par l'un de ces termes. A titre d'exemple, la requête politique ou gouvernement produit les documents indexés par le terme politique ou le terme gouvernement.

3 Le modèle vectoriel

Dans ce modèle, un document est représenté sous forme d'un vecteur dans l'espace vectoriel composé de tous les termes d'indexation [Salton, 1971]. Les coordonnées d'un vecteur document représentent les poids des termes correspondants. Formellement, un document d_i est représenté par un vecteur de dimension n,

d_i = *(wi,1 wi,2, ... , wi,n) pour i = 1, 2, ..., m.*

Où : wi,j est le poids du terme tj dans le document di.

m est le nombre de documents dans la collection.

n est le nombre de termes d'indexation.

Une requête Q est aussi représentée par un vecteur de mots-clés défini dans le même espace vectoriel que le document.

$Q = (w_{Q1}, w_{Q2}, ..., w_{Qn})$

Où w_{Qj} est le poids de terme tj dans la requête *Q*. Ce poids peut être soit une forme *de tf*idf*, soit un poids attribué manuellement par l'utilisateur.

La figure 2.1 montre un exemple d'espace vectoriel composé des trois termes *t 1, t2 et t3*. Les indexes de deux documents *D1et D*2 et une requête *Q* sont représentés dans cet espace.

3.1 La mesure vectorielle de la pertinence

La pertinence du document di pour la requête Q est mesurée comme le degré de corrélation des vecteurs correspondants. Cette corrélation peut être exprimée par l'une des mesures suivantes :

Le produit scalaire : $Sim(d_i, Q) = \sum_{i=1}^{n} w_{Qj} * w_{ij}$ *(2.5)*

La mesure du cosinus: $Sim(d_i, Q) = \frac{\sum_{i=1}^{n} w_{Qj} * w_{ij}}{\left(\sum_{i=1}^{n} w_{Qj}^2\right)^{\frac{1}{2}} * \left(\sum_{i=1}^{n} w_{ij}^2\right)^{\frac{1}{2}}}$ *(2.6)*

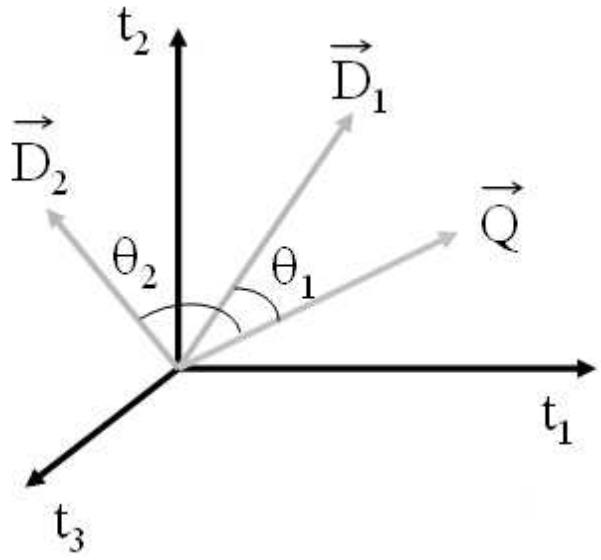


Figure 2.1: Exemple d'espace vectoriel.

L'un des avantages du modèle vectoriel réside dans sa simplicité conceptuelle et de mise en œuvre. En outre, il permet de trier les résultats d'une recherche à travers une mesure de similarité document/requête, en plaçant en tête les documents jugés les plus similaires à la requête. Mais ce modèle ne permet pas de modéliser les associations entre les termes d'indexation. Chacun des termes est indépendant des autres. Le modèle vectoriel généralisé (Generalized Vector Space Model) [Wong et al, 1985] permet de résoudre le problème d'indépendance des termes.

Cependant, l'inconvénient majeur de l'approche vectorielle réside dans le fait que le calcul de la similarité, du centroïde des documents pertinents et des documents non pertinents fait abstraction des valeurs des composantes des vecteurs requête et documents. Par ailleurs, il est

impossible de représenter des phrases ou des mots multi-termes (bases de données, analyse en composantes principales, etc.).

3.2 L'indexation en sémantique latente (LSI)

Toujours dans une approche vectorielle, l'objectif fondamental du modèle LSI [Dumais 1995] est d'aboutir à une représentation conceptuelle des documents, dans ces derniers les effets dus à la variation d'usage des termes dans la collection sont nettement attenues. Ainsi, des documents partageant des termes co-occurents ont des représentations proches dans l'espace défini par le modèle. Ceci permet de sélectionner des documents pertinents même s'ils ne contiennent aucun mot de la requête.

LSI [Dumais 1995] [Foltz 1990] [Furnas et al 1988] est une technique qui tend à implanter partiellement la recherche sémantique. Comparativement au modèle vectoriel, la technique LSI réduit la dimension de l'espace de représentation aux seuls vecteurs de représentation de l'information sémantique et ce en réduisant l'effet de variations d'utilisation des termes. Cette technique propose une représentation optimale à partir d'un espace de représentation de termes-documents. Cette optimisation est basée sur une analyse en composantes principales de la matrice termes-documents [Berry et al 1999].

La méthode LSI n'est pas souple pour certains types d'applications dont le filtrage. En effet, la performance et la stabilité du système dépendent largement de la quantité et la qualité des données traitées.

4 Le modèle probabiliste

Maron et Kuhns [Maron et al 1960] étaient à l'origine de l'apparition du modèle probabiliste. Ce modèle utilise un modèle mathématique fondé sur la théorie de la probabilité. La similarité entre un document et une requête est mesurée par le rapport entre la probabilité qu'un document d donné soit pertinent pour une requête Q, notée p(d/Q), et la probabilité qu'il soit non pertinent $p\left(\frac{\bar{d}}{Q}\right)$. Ces probabilités sont estimées par les probabilités conditionnelles selon qu'un terme de la requête est présent dans un document pertinent ou dans un document non pertinent.

Soient, *P(termepresent/pertinent)* probabilité qu'un terme de la requête apparaisse dans un document pertinent et *P(termepresent/nonpertinent)* la probabilité que ce terme apparaisse

dans un document non pertinent. Cette mesure est souvent donnée par la formulation suivante :

$$RSV(Q,d) = \frac{p\left(\frac{d}{Q}\right)}{p\left(-\frac{d}{Q}\right)} = \sum_{i=1}^{t} \frac{\log(p(1-q))}{q(1-p)} \quad (2.7)$$

Où:

p: est P(terme ti present/ d pertinent),

q : est P(terme ti present/ d non pertinent),

t : nombre total de termes dans la requête.

L'idée générale de cette approche est d'ordonner l'ensemble des documents de la collection selon leurs probabilités de pertinence. Si un système de recherche ordonne les documents selon leurs probabilités de pertinence, où les probabilités sont estimées en fonction des données disponibles au système, pour cet objectif, alors la performance du système sera améliorée. Selon Robertson, deux hypothèses doivent être vérifiées pour garantir l'optimalité d'ordonnancement d'une collection de documents :

- la pertinence des documents doit être une variable aléatoire binaire prenant l'une des valeurs vrai ou faux,
- la pertinence d'un document ne doit pas influencée sur la pertinence d'un autre document.

Selon Jacques Savoy [Savoy, 1994], le modèle de recherche probabiliste est plus efficace que le modèle de recherche booléen, mais moins performant que le modèle de recherche vectoriel. Pour plus d'information, il existe des états de l'art comme [Crestani et al. 1998], ou la présentation des expériences probabilistes en RI de « l'école de Londres » [Sparck Jones, 2000].

Okapi [Robertson et al 1999] est un exemple de système de recherche d'information basé sur l'approche probabiliste. L'approche Okapi aussi connue sous le nom de pondération (ou poids) BM25, a été crée avec le souhait de construire un modèle probabiliste prenant en compte la fréquence des termes ainsi que la taille des documents [Sparck Jones et al., 2000]. La fonction BM25 est utilisée pour la recherche de base selon Okapi :

$$rsv(D,Q) = \sum_{T \in Q} w^{(1)} \frac{(k1+1)tf}{K+tf} \frac{(k3+1)qtf}{k3+qtf} \quad (2.8)$$

$$K = k1\,(1-b+b.\frac{dl}{avdl}) \quad (2.9)$$

Où:

Q: une requête, contenant les termes T,

tf : fréquence d'apparition du terme T dans le document D,

qtf : fréquence d'apparition du terme T dans la requête Q,

k1, b et k3 : paramètres dépendant de la nature des requêtes et du corpus[2]*,*

dl : longueur du document D,

avdl : la longueur moyenne d'un document,

w(1) : poids de Robertson-Sparck Jones du terme T dans la requête Q:

$$w(1) = \frac{\log\left(\frac{r+0.5}{R-r+0.5}\right)}{\frac{n-r+0.5}{N-n-R+r+0.5}} \quad (2.10)$$

r : nombre de documents pertinents contenant le terme T,

R: nombre total de documents pertinents,

n : nombre de documents contenant le terme T,

N : nombre total de documents.

[2] *k*1 *et b valent* 1.2 *et* 0.75 *respectivement pour la base AP de* TREC-8 *[Voorhees et al 1999], Pour de longues requêtes, k*3 *vaut* 7 *ou* 1000.

On constate effectivement que le modèle probabiliste devient encore plus intéressant en présence de documents pertinents et non pertinents dans le cadre par exemple d'un retour de jugements de l'utilisateur. Ceci permet de sélectionner les termes les plus pertinents pour exprimer les besoins de l'utilisateur.

5 Conclusion

Les modèles booléens, vectoriels et probabilistes représentent les modèles classiques de RI. En effet, ils utilisent un procédé en trois étapes (décrites précédemment) qui consiste à normaliser les données textuelles en entrée, puis représenter ces données dans une structure adéquate, et enfin mettre en correspondance les représentations des documents avec la représentation de la requête. Néanmoins, chacune des trois approches présente ses propres avantages et inconvénients. Alors que le modèle booléen est caractérisé par sa simplicité et par conséquent son large application dans les SRI, le modèle vectoriel apporte un amélioration fictive en performances en modélisant la pondération des termes dans les documents et à travers toute la collection. Par ailleurs, l'approche probabiliste présente un cadre théorique plus élégant pour l'estimation de la pertinence bien que les performances restent presque équivalentes à celles du modèle vectoriel.

Toutefois, l'indexation en sémantique latente apporte une solution astucieuse pour prendre en charge certains aspects sémantiques dans la recherche vectorielle. Nous décrivons dans le chapitre suivant une approche probabiliste plus fondée pour prendre en charge la sémantique latente lors de l'indexation des documents.

CHAPITRE III :
LES MODELES PROBABILISTES A THEME LATENT

1 Introduction

La modélisation par thème peut être vue comme un problème de catégorisation non-supervisée de documents, dit en d'autres termes, celui du repérage de classes thématiquement homogènes dans un corpus textuel. Si cette problématique est relativement ancienne en lexicométrie [Benzécri et al, 1981], la disponibilité de larges corpus numérisés et les besoins convergents d'applications de recherche d'information ont suscité de nouvelles propositions : en particulier, le modèle LSI [Deerwester et al., 1990], son équivalent probabiliste [Hofmann, 2001](PLSI), ou encore tout un ensemble de modèles probabilistes génératifs : le modèle de mélange de lois multinomiales [Nigam et al., 2000; Clérot et al., 2004], le modèle d'allocation latente de Dirichlet. Nous décrivons dans ce chapitre trois modèles probabilistes à thème et nous présentons quelques approches de leur intégration dans un SRI.

2 Le modèle de mélange d'uni-gramme

Pour représenter les textes, nous adoptons le modèle du sac-de-mots, c'est-à-dire que le vocabulaire est connu et fini et que chaque document est représenté par un vecteur de fréquence sur cet ensemble.

2.1 Modèle de document dans le cas uni-gramme

Soit D1 = l'ensemble des sous séquences contiguës de D de taille maximale 1.

$$D1 = \{(td,1), (td,2), \ldots, (td,i), \ldots, (td,NdI)\}$$

On appelle MD,1 le modèle uni-gramme du document D. On notera MD,1 par MD en donnant préalablement n = 1.

$$MD = \{(td,i, P(td,i)), \forall i \in [1 \ldots N_1 dI\ ]\} \quad avec\ td,i \in V\ et\ N_{dI} \leq Nd$$

$$MD = \{(td,1, P(td,1)), (td,2, P(td,2)), \ldots (td,i, P(td,i)), \ldots, (td,L, P(td,NdI))\}$$

avec $td,i \in V$ *et* $\forall i, \forall j \quad i \neq j$, $td,i \neq td,i$

$P(t_{d,i}) = \| t_{d,i} \|$; $\| t_{d,i} \|$ = estimation de la vraisemblance maximale, c'est-à-dire la fréquence du terme dans le document D

$\| t_{d,i} \| = tf_{tdi}$

$\| t_{d,i} \| = \frac{t_{d,i}}{D} = \frac{t_{d,i}}{N_D}$

2.2 Fonction de correspondance

La fonction de correspondance se base sur le principe de génération de la requête par le modèle de document. Ainsi, le calcul de la probabilité $P(Q|M_D)$ que la requête Q soit générée par MD s'effectue pour chaque document D :

$$P(Q|M_D) = P(t_{q,1}, t_{q,2}, \ldots, t_{q,Nq} | M_D)$$

Soit le modèle uni-gramme :

$$M_D = \{(t_{d,1}, P(t_{d,1})), (t_{d,2}, P(t_{d,2})), \ldots, (t_{d,i}, P(t_{d,i})), \ldots, (t_{d,NdI}, P(t_{d,NdI}))\}$$

On suppose l'indépendance des termes de la requête. Dans le cas d'un modèle uni-gramme, la fonction de correspondance se généralise par :

$$\text{Pour chaque } D \in C : \quad P(Q|M_D) = \prod_{i=1}^{Nq} [P(t_{qi})] \qquad \textbf{(3-1)}$$

$P(Q|M_D) = P(t_{q,1}) * P(t_{q,2}) * \ldots * P(t_{q,i}) * \ldots * P(t_{q,Nq})$

$P(t_{q,i}) = P(t_{q,i}|M_D)$ par abus d'écriture

La probabilité se base sur le poids du terme de la requête dans le modèle de document. Il existe une relation d'égalité entre terme du document et terme de la requête. Si un terme de la requête n'apparaît pas dans le document, une fonction de lissage est utilisée pour évaluer $P(t_{q,i} | M_D)$ et ainsi éviter une probabilité nulle pour l'ensemble de la requête.

Dans le cas de modèle uni-gramme et avec un lissage par interpolation, le calcul de la probabilité d'un terme de la requête sachant un modèle de document s'exprime donc ainsi :

- Pour tout $t_{q,i} \in Q$, tel que $\exists \; t_{d,j} \in M_D$, $t_{d,j} = t_{q,i}$: $P(t_{q,i}|M_D) = P_{MD}(t_{q,j}) + (P_{MC}(t_{q,j})$
- Pour tout $t_{q,i} \in Q$, tel que $\forall \; t_{d,j} \in M_D$, $t_{d,j} \neq t_{q,i}$: $P(t_{q,i}|M_D) = P_{MC}(t_{q,j})$

Deux dimensions entre donc en compte dans l'estimation de la probabilité qu'une requête générées par un document :

-la probabilité pour chaque terme d'appartenir au modèle de document M_D

-la probabilité pour ces mêmes termes d'appartenir au modèle de corpus M_C.

3 Le modèle de clustering

Le modèle de cluster est connu comme le modèle de mélange d'uni-grammes, qui a été étudié dans les années 60 [Sparck Jones, 1971]. Dans ce modèle, il a été supposé que tous les documents tombent dans un ensemble fini de K groupes (les thèmes). Dans chaque groupe, les documents contiennent un thème particulier z, et chaque thème z associe à une distribution multinomiale P (w | z) sur le vocabulaire. Le processus de générer un document d (w_1 , ... w_{N_d}) est comme suit :

1) Choisir un thème à partir une distribution multinomiale z avec le paramètre θ_z
2) pour i=1… N_d choisir le mot w_i de thème z avec une probabilité P (w_i | z).

La probabilité globale d'observer le document d de ce modèle est:

$$P\left(w_1, \dots w_{N_d}\right) = \sum_{z=1}^{k} P(z) \prod_{i=1}^{N_d} P(w_{i\backslash z}) \qquad (3\text{-}2)$$

L'une des méthodes d'estimation des paramètres pour ce modèle est de grouper les documents de la collection en K groupes, puis utiliser l'estimation de la vraisemblance maximale du modèle à thème P (w \ z) pour chaque groupe.

$$P(w\backslash D) = \frac{N_d}{N_d + \mu} P_{ML}(w\backslash D) + 1 - \left(\frac{N_d}{N_d + \mu}\right) P(w\backslash cluster) \quad (3-4)$$

Avec le nouveau modèle de document, ils ont mené des expériences sur plusieurs collections TREC, estimant que la recherche à base de clustering effectue régulièrement à travers des collections. Des améliorations importantes sont obtenues par rapport au document basé sur la recherche. Ce modèle possède parfaitement la sémantique générative, mais l'hypothèse que « chaque document est généré à partir d'un thème unique » est une limite qui peut devenir un problème pour les longs documents et les grandes collections.

4 L'indexation sémantique latente probabiliste (PLSI)

Le modèle d'indexation sémantique latente probabiliste qui est introduit par Hoffman (2003) a rapidement gagné l'acceptation dans la modélisation des applications textes. PLSI est un modèle qui utilise une variable latente pour une cooccurrence générale des données, il associe une classe inobservée variable (thème) à chaque observation (c'est à dire, à chaque occurrence d'un mot). PLSI modélise chaque document comme un mélange de thèmes.

Ce qui suit le processus pour générer des documents dans le modèle PLSI :

1) Choisir une distribution du mélange de thème *P(.| d)* pour chaque document d
2) Choisir un thème latent z, avec une probabilité *P(z | d)* pour chaque mot symbolique.
3) Générer le mot symbolique w avec une probabilité *P(w | z)*.

La probabilité de générer un document d, comme un sac de mots $w_1 \dots w_{N_d}$ (N_d Est le nombre de mots du document d), est la suivante:

$$\mathbf{P}\left(w_1, \dots w_{N_d}\right) = \prod_{i=1}^{N_d} \sum_{z=1}^{k} P\left(w_{i \setminus z}\right) P(z \setminus d) \qquad (3-5)$$

Hoffman (1999) a exploité PLSI comme un modèle d'uni-gramme pour lisser les distributions de mot empirique et comme un modèle d'espace latent pour fournir un document de faible dimension à la fois. Malgré que des grandes améliorations aient été signalées dans la performance de recherche, la taille des collections et la longueur des documents dans les collections sont loin de représenter la réalité des environnements RI. On outre, le modèle PLSI lui même a un problème dans sa sémantique générative qui n'est pas bien définis (Blei et al, 2003).

Ainsi il n'ya pas de façon naturelle à prédire un document inédit, et le nombre de paramètres du PLSI croît linéairement avec le nombre de documents de formation, ce qui rend le modèle sensible capable à sur-apprendre.

5 L'allocation latente de Dirichlet (LDA)

Latent Dirichlet Allocation [Blei et al, 2003] est un puissant algorithme d'apprentissage pour, de façon automatique et conjointe, classer des mots dans des documents dans des mélanges de contextes. Il a été appliqué avec succès pour modéliser les changements dans les domaines scientifiques au cours du temps. Dirichlet vient de Johann Peter Gustav Lejeune Dirichlet, Mathématicien allemand étudiait en France en 19ème siècle, qui a fait des travaux dans le domaine de l'analyse complexe et les lois de probabilité.

L'Allocation Latente de Dirichlet est un modèle génératif probabiliste. Partant de l'hypothèse que l'ordre des documents dans la collection est celui des mots dans un texte sont indifférents, LDA définit des modèles de mélanges finis sur des ensembles de sujets sous-jacents pour générer la collection, chaque sujet étant modélisé comme un mélange infini sur des probabilités des sujets sous-jacentes. Il a été démontré que PLSI est un cas particulier de LDA [Girolami and Kaban, 2003].

Les termes rares tendent à apparaitre par « rafales » : si un terme rare apparait dans un texte, il est probable qu'il y apparaisse plus qu'une seule fois. Les multinomiales sous-estiment notablement la queue de la distribution.

5.1 Processus d'apprentissage

Le modèle génératif de Dirichlet a été proposé pour produire des distributions à queue épaisse décrivant mieux la distribution des termes [Madsen et al, 2005], mais il entraine des solutions à forme non close, impossibles à calculer de façon exacte. Le modèle « Distribution de Dirichlet lissée » (smoothed Dirichlet allocation) est un modèle génératif sur la base de la KL-divergence. Des approximations théoriquement fondées de ce modèle produisent des solutions à forme close similaires à celles produites par les multinomiales, mais avec des distributions à queue épaisse [Nallapati, 2006].

Le modèle LDA partage avec des nombreux modèles une représentation en « sac-de-mots», selon laquelle chaque document est représenté par un profil d'occurrences qui est associée à un thème. Les différentes occurrences au sein d'un même document restent toutefois liées par une variable latente qui contrôle globalement la distribution des thèmes au sein du document.

Le processus génératif de base LDA ressemble PLSI, dont le mélange de thème en PLSI est conditionné en chaque document. En LDA, le mélange de thème est tiré d'un Dirichlet conjugué prioritaire qui reste le même pour tous les documents. Le processus de générer un corpus est comme suit (le LDA lissé) :

1) Choisir une distribution multinomiale φ_z pour chaque thème z d'une distribution de Dirichlet avec paramètre β.

2) Pour chaque document d, choisir une distribution multinomiale θ_d à partir d'une distribution de Dirichlet avec paramètre α.

3) Pour chaque mot symbolique w dans le document d, choisir un thème z $\in$ {1…k} de la distribution multinomiale θ_d .

4) Choisir un mot w de la distribution multinomiale φ_z , donc la probabilité de générer un corpus est la suivante:

$$P(doc_1 \dots doc_{n\backslash\alpha}, \beta) =$$

$$\iint \prod_{z=1}^{k} P\left(\varphi_{z\backslash\beta}\right) \prod_{d=1}^{n} \left[\left(\theta_{d\backslash\alpha}\right]\right) \prod_{i=1}^{N_d} \sum_{z_{i=1}}^{k} P\left(z_{i\backslash\theta}\right) P\left(w_{i\backslash z}, \varphi\right) d\theta d\varphi \quad (3\text{-}5)$$

Selon Wei et Croft (2006), LDA possède une sémantique générative cohérente en traitant la distribution de mélange de thème comme une variable aléatoire avec k- paramètre cachée après un large éventail de paramètres individuels qui sont explicitement liés à l'ensemble de formation; ainsi LDA surmonte le problème de su-rajustement et le problème de générer des nouveaux documents dans PLSI.

Ainsi le LDA permet à un document de contenir un mélange de thèmes, relaxant l'hypothèse faite dans le modèle clustering que chaque document est généré à partir d'un thème unique. Cette hypothèse peut être trop limitée pour modéliser d'une manière efficace une grande collection de documents, en revanche, le modèle LDA permet à un document d'exposer plusieurs thèmes à divers degrés.

5.2 A quoi sert le LDA?

Le LDA a un but essentiel de classement, il permet d'associer un contexte à un document à partir des mots contenus dans ce document, lesquels mots pris individuellement pourraient appartenir à des contextes différents.

L'expérience montre que les premiers liens dans les pages de résultats de recherche ont un contenu plus pertinent que ceux qui viennent après.

6 LDA pour la recherche ad-hoc

La faisabilité d'introduire la modélisation par thème dans la recherche ad-hoc n'est pas assez claire. Peu de travaux récents ont proposé certaines approches pour améliorer le calcul de la pertinence mais les résultats restent mitigés et suscitent de plus amples recherches [Wei et Croft, 2006] [Yi et Allan, 2009]. Néanmoins, nous pouvons introduire le modèle de thème dans une recherche ad-hoc en suivant trois approches principales :

6.1 Modèles de recherche combinés

Parmi les approches du modèle de langage pour l'estimation de la pertinence d'une requête par rapport à un modèle de document, il est possible d'utiliser le modèle de probabilité de requête. Sous l'hypothèse de sac de mots (indépendance des termes), et pour chaque

document D3 et une requête Q composée des termes {q}, un score de pertinence est estimé par :

$$P(Q|D) = \prod_{q \in Q} P(q|D) \tag{3-6}$$

Partant du principe de lissage de Dirichlet et intégrant le modèle LDA, une étude proposa de combiner linéairement trois modèles pour estimer la probabilité conditionnelle de générer un terme q sachant le document D [Wei et Croft, 2006] :

Le maximum de vraisemblance du terme q dans le document D : PMV(q|D),

Le maximum de vraisemblance du terme q dans le corpus C : PMV(q|C),

La probabilité du terme q selon les distributions estimées par LDA : PLDA(q|D).

Sachant que ND est la longueur (en mots) du document D, que μ est le paramètre à priori de lissage de Dirichlet et que λ (entre 0 et 1) est un paramètre d'ajustement de la contribution du modèle LDA, la pertinence d'un terme par rapport à un document est mesurée par :

$$P(q|D) = \lambda\left(\frac{N_D}{N_D + \mu} P_{MV}(q|D) + \frac{\mu}{N_D + \mu} P_{MV}(q|C)\right) + (1 - \lambda) P_{LDA}(q|D) \tag{3-7}$$

Les auteurs dans [Wei et Croft, 2006] confirmèrent avoir obtenu des résultats légèrement meilleurs que celles réalisés par le modèle de requête en choisissant (μ=1000, λ=0,7). Néanmoins, aucune comparaison, incluant d'autres modèles probabilistes ou vectoriels, n'a été appliquée. En testant sur le corpus AP4, les auteurs reportèrent une amélioration de la précision moyenne à 0,2651 par rapport au modèle de requête (0,2179) ou celui à base de cluster (0,2326).

Ce modèle de recherche souffre de certaines limites telles que la complexité de la formule de pertinence et la faible contribution du modèle LDA (1-λ=0,3). Par ailleurs, les expérimentations sur deux corpus moyens (CF et CACM) étaient infructueuses et les performances obtenues étaient très faibles par rapport à d'autres modèles de références.

[3] Rappelons qu'il s'agit du modèle de document M_D.
[4] Dépêches de presse de la période 1988-90 de l'*Associated Press*

6.2 Similarité dans l'espace des thèmes

Du moment où le modèle LDA permet de calculer, dans l'espace des thèmes, la distribution de tout document (même nouveau), nous pouvons donc estimer la distribution à posteriori d'une requête Q et mesurer la valeur de pertinence RSV de chaque document D.

La divergence de Kullback-Leibler peut être utilisée pour mesurer la dissemblance entre deux distributions p et q comme suit [Heinrich, 2008] :

$$D_{KL}(p,q) = \sum_{j=1}^{T} p_j \log_2 \frac{p_j}{q_j} \tag{3-8}$$

Selon l'équation (3-8), la divergence DKL est une fonction non-négative mais asymétrique. Il est possible de dériver une forme plus pratique d'une mesure symétrique :

$$SymKL(p,q) = \frac{1}{2}\left[D_{KL}(p,q) + D_{KL}(q,p)\right] \tag{3-9}$$

C'est la forme symétrique de la divergence de Kullback-Leibler qu'on utilise comme mesure de similarité entre les distributions des documents.

Bien que la divergence de Kullback-Leibler parait comme la plus appropriée pour mesurer la similarité entre ce genre de distributions, les fonctions de similarité vectorielle, telles que le cosinus, ont produit les meilleures performances [Wei et Croft, 2006]. Nos expérimentations ont confirmé le même constat.

6.3 Extension thématique de la requête

L'approche d'extension de la requête part du fait que le besoin d'information ne peut être satisfait complètement dans un système de recherche à base de mots clés (termes d'une requête). L'objectif consiste à enrichir la requête par de nouveaux termes exprimant le même sujet. Du moment où il permet de regrouper les mots les plus consistants dans le même thème, le modèle LDA paraît comme une alternative astucieuse pour découvrir les termes contextuellement rattachés à une requête donnée.

En considérant les relations associatives entre les termes, on peut mesurer une similarité basée sur les distributions conditionnelles. Pour chaque terme wi, on exprime le degré de dépendance par rapport au terme w comme suit [Steyvers et Griffiths, 2007] :

$$P(w_i|w) = \sum_{j=1}^{T} P(w_i|z = j)P(z = j|w) \quad (3\text{-}10)$$

Les termes w'i maximisant l'expression dans (3-10) interprètent une forte dépendance au terme w (w→w'i). En fixant un seuil δ de tolérance (pour accepter les termes les plus rattachées), il est possible de définir un cadre pour reconstruire la requête étendue QE à partir de la requête de base Q = {q}.

$$QE_{\delta}^{-} = \bigcup_{q \in Q} \{q', P(q'|q) \geq \delta \wedge P(q'|q) \geq P(q|q')\} \quad (3\text{-}11)$$

$$QE_{\delta}^{+} = \bigcup_{q \in Q} \{q', P(q|q') \geq \delta \wedge P(q|q') < P(q'|q)\} \quad (3\text{-}12)$$

$$QE_{\delta} = QE_{\delta}^{-} \cup QE_{\delta}^{+} \quad (3\text{-}13)$$

L'ensemble QE-, dans l'équation (3-11), représente l'extension de la requête Q par les hypo-termes dépendants (q→q'). Alors que QE+, dans l'équation (3-12), contient la requête étendue par les hyper-termes dont ceux de la requête Q en dépendent (q'→q). Afin d'estimer l'effet de l'approche d'extension de la requête, on calcule le nombre total des termes obtenus dans les requêtes de chaque corpus. Sur 704 termes des requêtes CF (635 termes dans CACM) nous reportons dans Tableau 3- le taux d'extension global (QE-$\cup$ QE+) en fonction du nombre de thèmes LDA et du seuil de toléranceδ.

Corpus	**CF**			**CACM**		
δ	10%	20%	30%	10%	20%	30%
LDA-100	1 566,3%	392,9%	130,3%	1 048,5%	155,1%	100,9%
LDA-200	1 596,6%	181,4%	120,3%	925,0%	254,5%	123,0%
LDA-300	1 164,1%	209,9%	110,8%	906,9	195,0%	104,7%

Tableau 3-1. Taux d'extension des requêtes dans les corpus CF et CACM.

Corpus	Terme	Extension
CF	Activity	cytoplasmic, esteras, extend, enzyme, chromatographic, proteolytic, arginine
	Bronchial	Asthmatic, aerosol, tracheostomy, asthma, endotrach, bronchoscopic
	Effect	serial, beneficial, dialyze, extraction, days, replic, adverse, intralipid, student
	Mucus	hypersecretion, variance, nonspecific, shed, responsive, production, viscid, technician
	Respiratory	pneumonia, colonization, escherichia, harbour, enterobacteriaceae, predominate, tract
	Role	issue, prospect, play
	Test	reliable, equipment, false, negative, bronchoconstrict, maximum, perform, northern, procedure, positive, strong, reasonably, reliably, collect
CACM	Approximation	cubic, spline, piecewis, quintic, ln, mesh, horowitz, minimax, periodic, ki, rational, reme, extrapolation
	Bit	register, shift
	Compiler	meta, translator, gri, assembler, compilation, compile, neliac, incremental
	Language	ability, familiar, assemble, advancement, macro, translation, deficiency, applicative, nonprogramm, calculus, meta, imperative, orient, grasp, assembly, crespi, lambda, semantics
	Mathematical	partly, apt, trigonometric, photocomposit, mathematics, pi, rodriguez, font, typeset
	Matrix	qr, column, band, row, eigen, multiplication, tridiagon, perturbation, inverse, latent, ergodic, subchain, reversible,

		diagonal, pentadiagon, pei, vector, eigenvalue, pencil, submatrix, f, eigenvector, symmetric, product, inversion
	Pattern	induce, chemical, shrink, waveform, electrocardiogram, match, carotid, arterial, peak, higgin, stark, pulse, recognition, structural, vault

Tableau 3-2. Exemples d'extension des termes dans les corpus CF et CACM (T=300, δ=10%).

La variation des taux d'extension et inversement proportionnelle au seuil de tolérance dans le même modèle de thème. Néanmoins, le seuil δ doit être soigneusement choisi afin d'obtenir une extension raisonnable pour chaque modèle LDA. En plus, nous dressons dans Tableau 3- l'extension de certains termes des requêtes dans les deux corpus selon modèle LDA à 300 thèmes latents et avec un seuil de tolérance δ=10%:

Nous proposons d'utiliser cette approche d'extension de requête avant d'appliquer un modèle de recherche dans l'espace de mots tel que TF, IDF ou Okapi.

7 Conclusion

Dans ce chapitre nous avons introduit les modèles probabiliste à thème avec plus de détail pour le LDA, qui devenu l'un des modèles les plus populaires dans les techniques de modélisation de texte par apprentissage automatique.

Le modèle LDA permet de construire des classifications qui sont réellement non-déterministes et qui recoupent, avec une bonne précision, les catégories d'origine. Cet effet est atteint par une modélisation toutefois peu naturelle, qui autorise le changement de thème à chaque mot. Le modèle LDA surmonte le problème de sur-ajustement et le problème de générer des nouveaux documents dans PLSI. Cependant, son intégration dans un système de recherche requiert davantage d'études et d'expérimentations afin d'apprécier son apport pour les SRI modernes. Nous décrivons dans les chapitres qui suivent notre implémentation pour évaluer différent modèles de recherche et appliquer la modélisation par thème sur des corpus de références.

CHAPITRE IV :
CONCEPTION & IMPLEMENTATION

1 Introduction

L'objectif de notre projet est de réaliser un système pour l'évaluation de la recherche Ad-Hoc pour une variété de modèles. Notre système doit offrir un environnement de test et de comparaison entre les performances de recherche aussi bien pour les modèles classiques que pour ceux à base de thèmes. Cependant, les approches d'intégration des modèles de thèmes dans les SRI restent controversée et pas assez solides. C'est pour cela, nous avons utilisé une plate-forme ouverte dans laquelle l'expert peut ajuster les entrées des tests au travers de fichiers XML pour chaque nouvelle collection. Nous présentons dans ce chapitre les éléments de conception et d'implémentation relatifs au système réalisé.

2 Architecture globale

Un itinéraire typique pour le développement de la recherche d'information ad-hoc est illustré à la figure 4.1. D'abord, il est nécessaire de préparer une collection de documents lisibles par machine. Deuxièmement, la transformation de la langue naturelle (NLP) est appliquée à chaque article de la collection. Les techniques NLP fréquemment utilisées comprennent l'atomisation (tokenization), la partie du discours (POS) de marquage, lemmatisation (lemmatization) et stemming, l'extraction des phrases, et l'extraction des concepts ontologiques. Troisièmement, tous les mots extraits, des phrases ou des concepts ontologiques sont enregistrées souvent dans les matrices creuses pour une utilisation future. Quatrièmement, les tâches d'exploration des données de texte telles que la recherche, le regroupement (clustering), la classification, sont exécutés sur des données indexées. Enfin, les résultats de recherche ou d'exploitation sont évalués avant d'être visualisés et analysés.

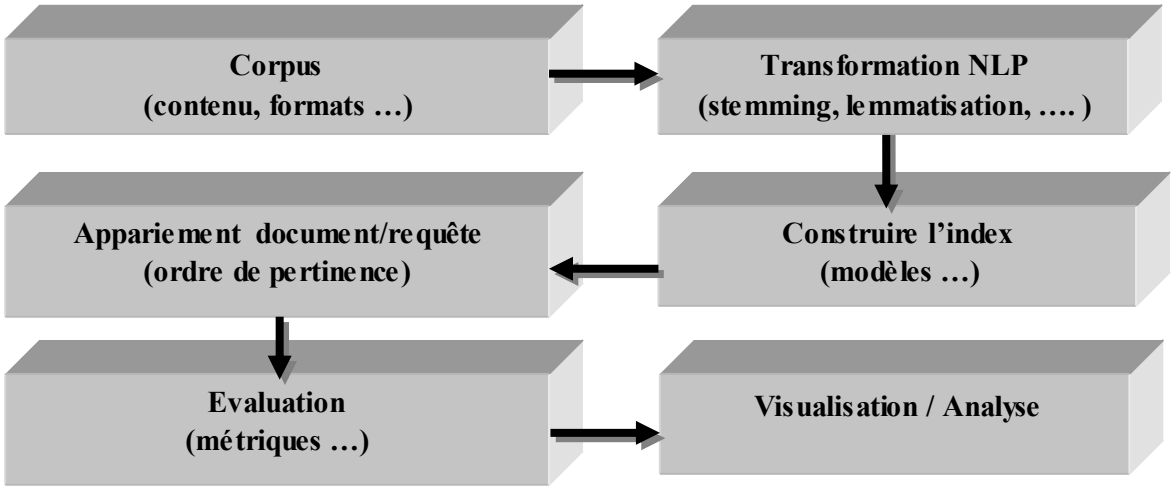


Figure 4.1 : Illustration du développement d'application

3 Modélisation par UML

3.1 Définition UML

UML (Unified Modeling Language en anglais, soit langage de modélisation objet unifié) est un langage graphique de modélisation des données et des traitements. C'est un langage à objet permettant de décrire l'analyse, la conception et l'implantation des systèmes.

3.2 Les diagrammes UML de notre application

3.2.1 Diagramme de cas d'utilisation

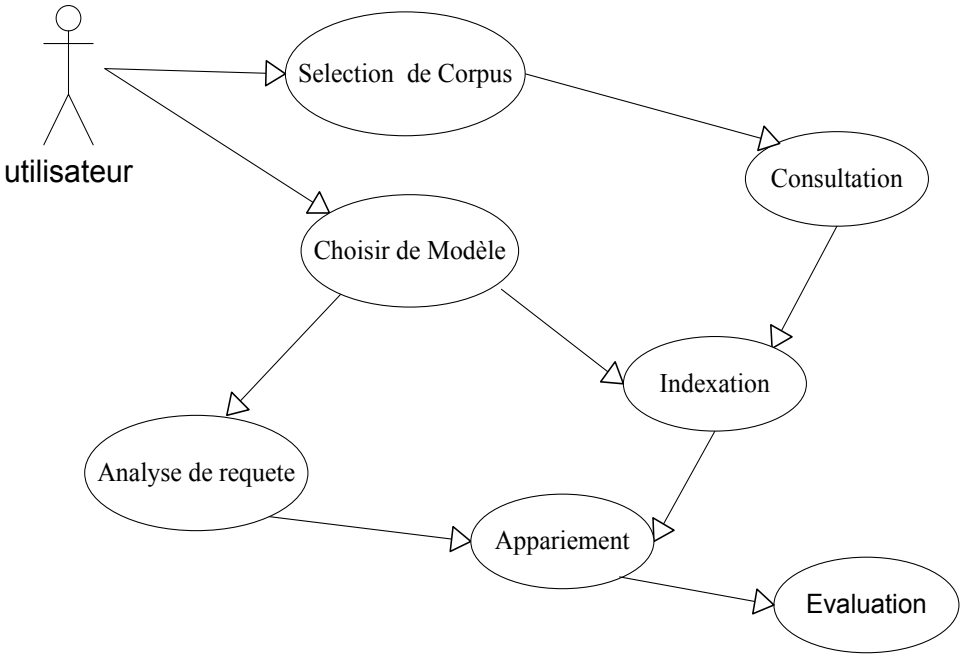


Figure 4.2 : le diagramme de cas d'utilisation

3.2.2 Diagramme de classes

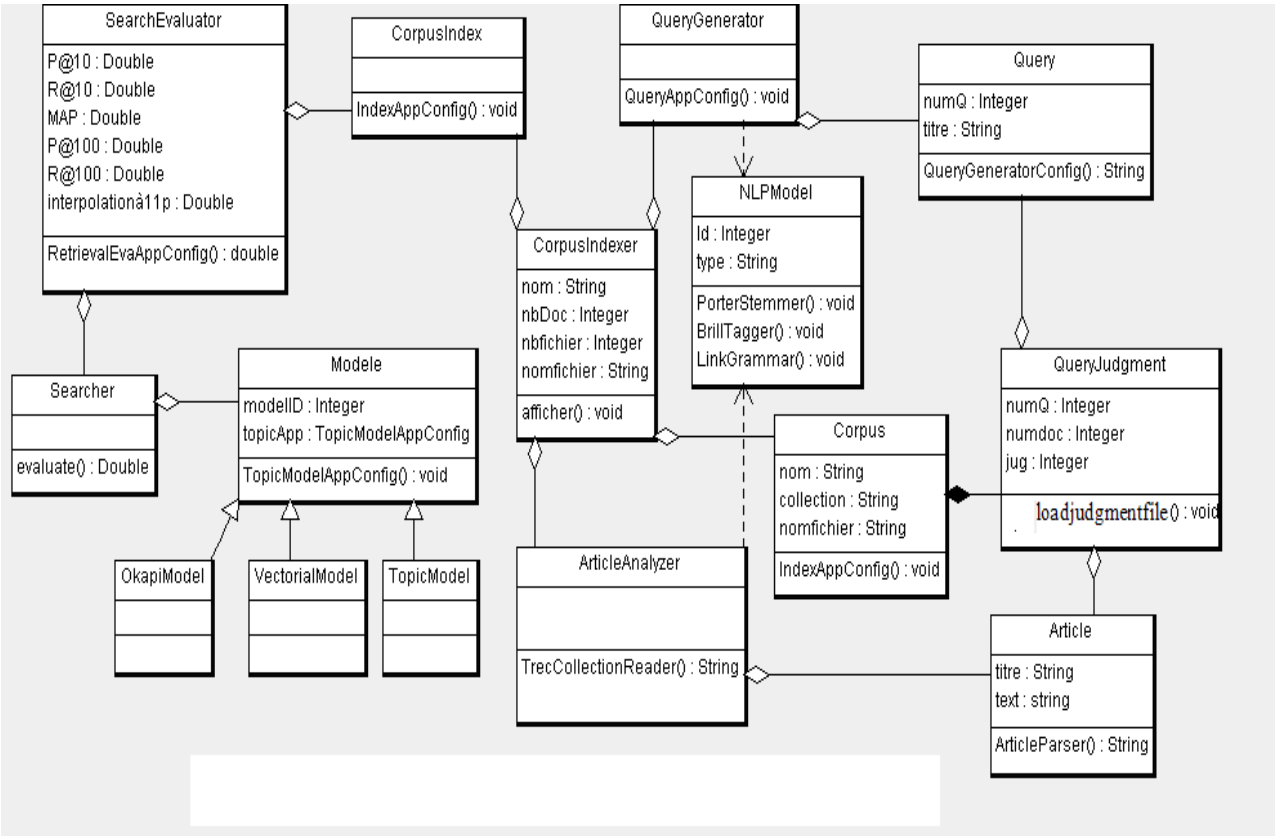


Figure 4.3 : le diagramme de classes

3.2.3 Diagramme d'activité

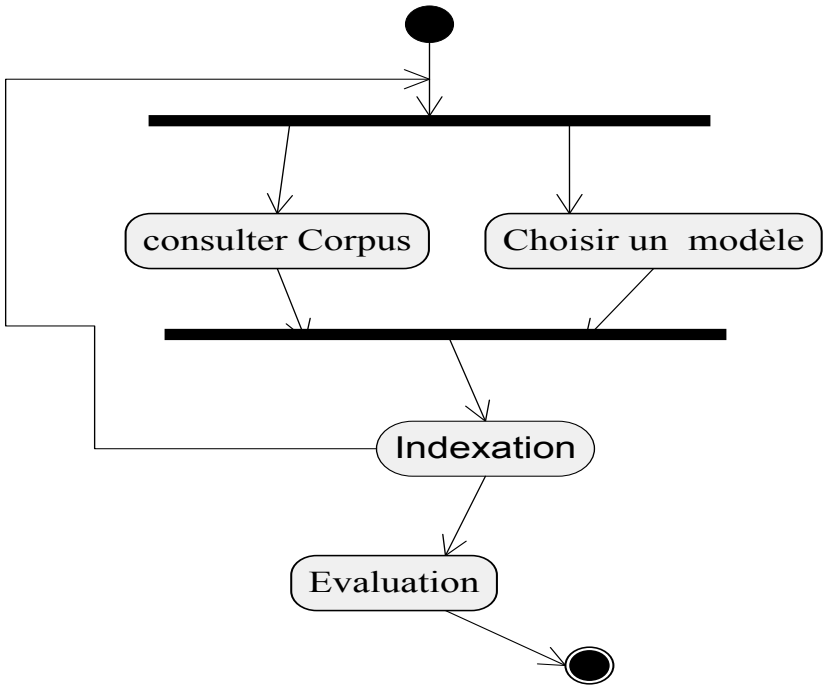


Figure 4.4 : le diagramme d'activité

3.2.4 Diagramme d'activité pour le choix d'un corpus

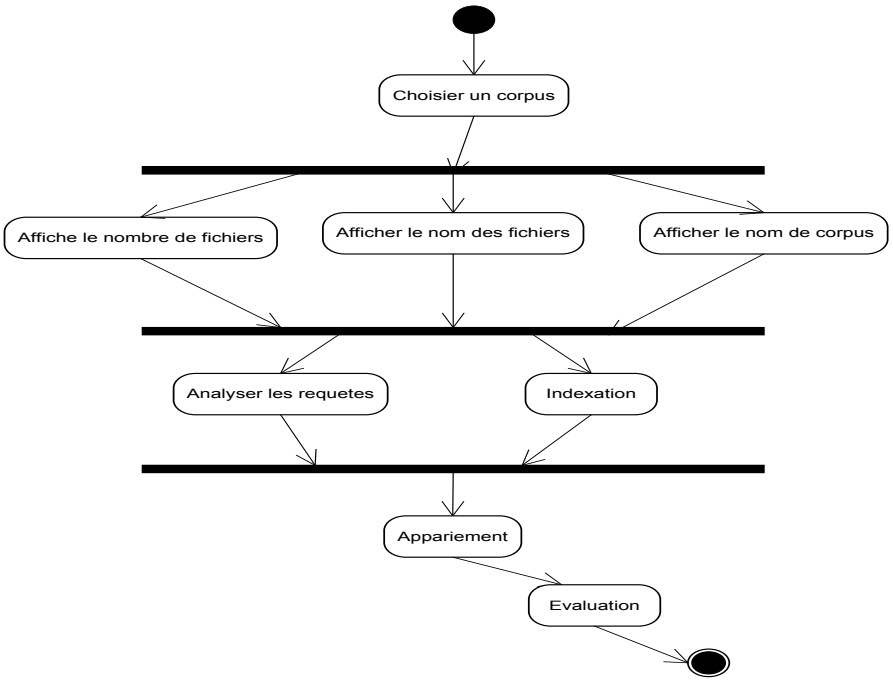


Figure 4.5 : le diagramme d'activité pour la sélection d'un corpus

3.2.5 Diagramme d'activité pour l'étape d'indexation

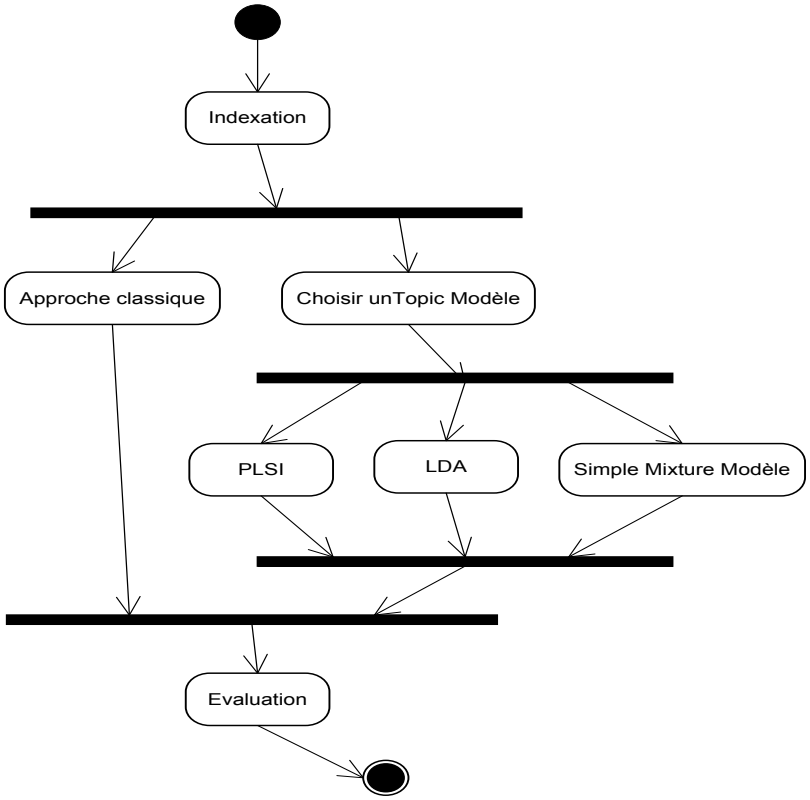


Figure 4.6 : le diagramme d'activité pour l'étape d'indexation

3.2.6 Diagramme d'activité pour l'étape d'évaluation

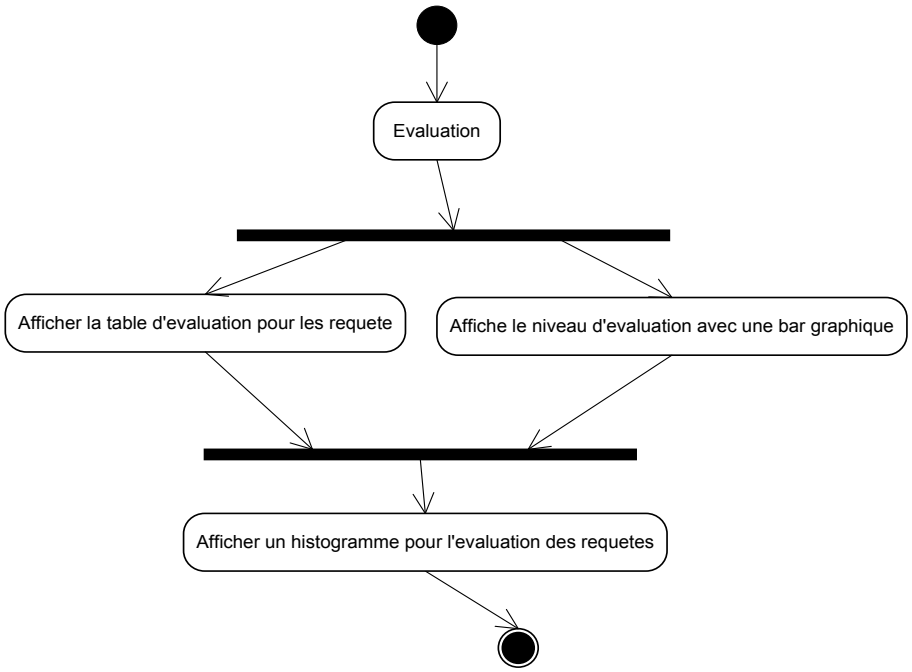


Figure 4.7 : le diagramme d'activité pour l'étape d'évaluation

3.2.7 Diagramme d'activité pour intégrer un corpus dans Dragon

N'importe quel corpus contient une collection, un fichier XML pour la configuration et un fichier de document texte pour les sujets de la collection peut l'intégré dans dragon.

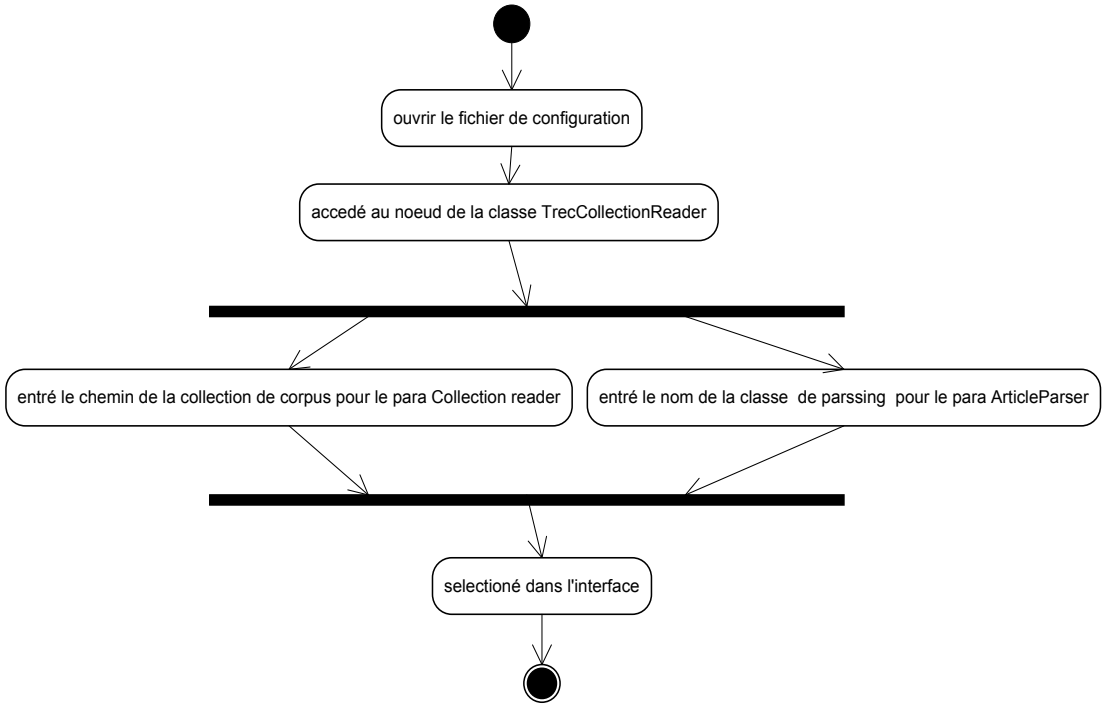


Figure 4.8 : le diagramme d'activité pour la configuration d'un corpus dans dragon.

4 Les outils NLP et de recherche

Le package NLP (natural language processing) fournit les fonctionnalités pour l'extraction des concepts différents (des mots, des phrases à plusieurs mots individuels, noms propres) ou des relations entre les articles lors de l'indexation.

Afin de faciliter l'extraction, dragon intègre divers outils de NLP comme stemmers, une partie des tagueurs de la parole (speech taggers), les analyseurs de phrases (sentence parsers), extracteus de phrase (phrase extractors), et nommé reconnaissance de l'entité. Pour la commodité du traitement du langage naturel, la boîte à outils fournit quatre structures de base de données, Document, paragraphe, une phrase, et la Parole, pour analyser et symboliser le contenu d'un article.

Les extracteurs et d'autres supportant les outils PNL utilisent ces quatre structures de base de données pour partager des données.

La boîte à outils définit trois types d'extracteurs de concept. La première est l'extracteur symbolique, qui extrait une séquence de mots individuels d'une phrase ou d'un document. Le second est l'extracteur de phrase, à savoir extraire plusieurs mots d'une phrase ou d'un document. L'extracteur phrase a besoin d'un Dictionnaire en entrée; le dictionnaire phrase pourrait être construit automatiquement par les outils de phrase comme Xtract. La troisième est l'extracteur de terme, qui extrait des termes d'une phrase ou un document.

Il existe des bibliothèques utilitaires pour le développement d'application relatives aux tâches de la recherche d'information telles que les packages ***Dragon***, ***LingPape***, ***Lemur***. De notre part, nous nous sommes basés dans la réalaisation de notre application sur la bibliothèque fournie par ***Dragon***.

4.1 LingPipe

LingPipe est une suite de bibliothèque Java pour l'analyse linguistique du langage humain. C'est une boite à outils pour le traitement de texte en utilisant la linguistique computationnelle. LingPipe est utilisé pour effectuer des tâches telles que:

Trouver les noms des personnes, des organisations ou des lieux dans les nouvelles.

- Classer automatiquement les résultats de recherche Twitter en catégories ;
- Suggérer l'orthographe correcte de requêtes ;

- Découvrir les relations entre les entités et les actions;
- Classer les passages de texte par langue, l'encodage, le genre, le sujet, ou le sentiment;
- Corriger l'orthographe par rapport à une collection de texte ;
- Découvrir les tendances importantes au fil du temps ;
- Fournir une partie du discours de marquage et de segmentation de phrase.

4.2 Dragon

Dragon est un package de développement basé sur le langage Java pour l'utilisation académique dans la recherche d'information (IR) et text mining (y compris la classification du texte, le regroupement de texte (text clustering,) et la modélisation des thémes).

Il est adapté pour les chercheurs qui travaillent dans IR et TM à grande échelle et préfèrent la programmation Java. En outre, il est différent de Lucene et Lemur, il fournit l'intégration des supports pour la IR sémantique et TM sémantique. La boîte à outils dragon intègre de façon transparente un ensemble d'outils NLP (naturel language processing), qui permettent à la boîte à outils d'indexer les collections de documents avec des systèmes de représentation différentes, y compris des mots, des phrases, l'ontologie à base de concepts. La boîte à outils n'a pas certaines fonctionnalités, y compris IR distribués et IR cross-language qui est une partie de Lemur.

Une autre caractéristique importante du Dragon est son évolutivité. Contrairement à de nombreux outils de text mining comme Weka, la boîte à outils dragon est spécialement conçue pour les applications à grande échelle. La boîte à outils utilise une matrice creuse pour faire des représentations des textes et ne charge pas toutes les données en mémoire dans le temps d'exécution. Par conséquent, il peut gérer des centaines de milliers de documents dotés d'une mémoire très limitée.

5 Les corpus

Notre application permet l'évaluation d'un système de recherche à base d'un thème latent LDA, elle intègre des corpus comme AP89, Genomics04, CACM et CF dans l'environnement de travail pour faire l'indexation et l'évaluation.

Le corpus AP89 contient 84.678 documents, 39.749.179 mots d'occurrence totale et 50 requêtes avec des jugements.

Le corpus Genomics TREC-2004, 50 requêtes (numérotées de 1 à 50) ont été crées sur la base de besoins d'information exprimés par des biologistes. Chaque requête est subdivisée en quatre champs soit le numéro de la requête (<ID>), un titre bref (<TITLE>), une description plus précise de la demande (<NEED>), et quelques informations permettant de mieux juger de la pertinence des articles dépistés (<CONTEXT>).

Le corpus CF (Cystic Fibrosis) contient 100 requêtes avec des jugements. Chaque jugement attribue un score de pertinence pour chaque document dont l'interprétation est comme suit (1 : pertinent, 2 : non pertinent). Pour notre évaluation, nous avons ignoré toute requête ayant un seul document pertinent.

Le corpus CACM inclut 63 requêtes dont 51 possèdent des jugements de référence (pertinent/non-pertinent). En ignorant celles qui donnent un seul document pertinent, nous nous sommes limités à 49 requêtes pour les évaluations.

6 L'environnement de programmation

Nous avons utilisés java comme langage de programmation, sous l'environnement NetBeans 6.9.1.

6.1 Le langage JAVA

Le langage Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy , présenté officiellement le 23 mai 1995 au SunWorld. Java est un langage de programmation orienté objets basé sur le langage C++ mais avec des fonctionnalités qui en rendent la programmation plus simple et plus sure (absence de pointeur, gestion automatique de la mémoire centrale, suppression des concepts complexes du C++, source de bugs (template, surcharge des opérateurs, opérateurs de conversion ...)).

6.2 L'IDE NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en Open Source par Sun. NetBeans permettant d'écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java mais peut supporter n'importe quel langage de programmation comme C, C++, JavaScript, PHP, HTML … Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages

Web).L'IDE NetBeans est un produit gratuit, téléchargeable sur son site officiel www.netbeans.org .

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris, Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Développement Kit (JDK) est requis pour les développements en Java. NetBeans en version 6.9.1 actuellement, est un IDE contenant la version complète de Java EE. Il contient GlassFish(l'implantation de référence d'un serveur d'applications Java), et génère les fichiers XML de configuration.

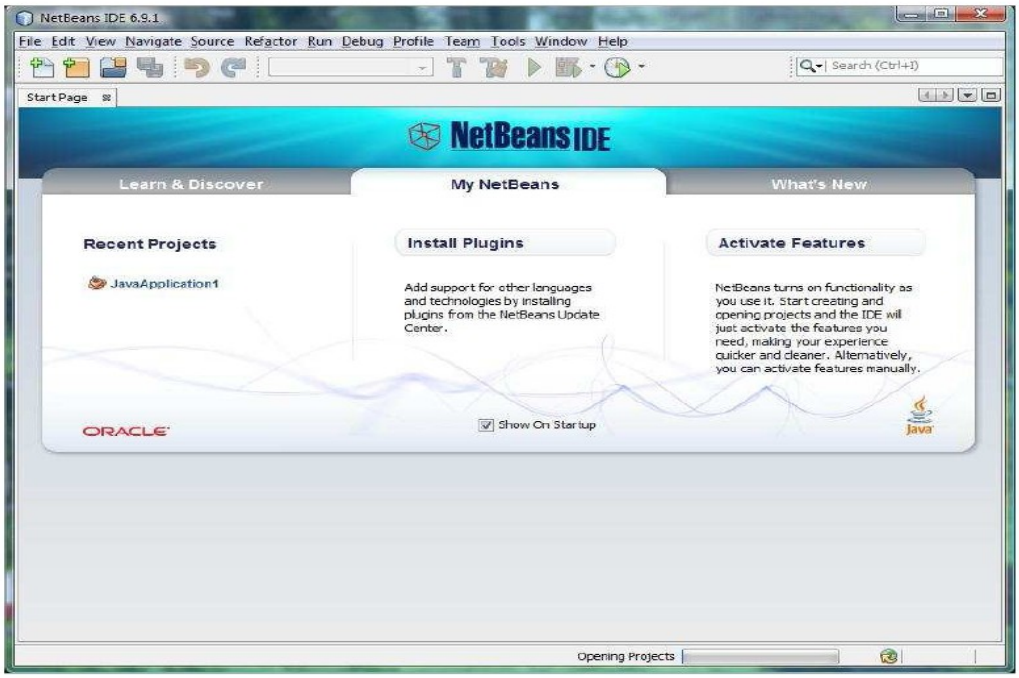


Figure 4.1 : l'environnement de développement intégré NetBeans.

7 Conclusion

Nous avons présenté dans ce chapitre la conception de notre application. L'étape essentielle est l'évaluation d'un système de recherche selon un modèle d'indexation et de représentation des documents et de requêtes. Ce modèle peut être choisi parmi une variété de choix ou bien ajusté par l'expert dans un fichier de configuration XML partant d'un corpus préalablement préparé.

CHAPITRE V :
MISE EN ŒUVRE & RESULTATS

1 Introduction

Nous décrivons dans ce chapitre les aspects pratiques relatifs à la mise en œuvre de notre système pour l'évaluation de la recherche Ad-Hoc. Nous dressons ensuite des échantillons des résultats préliminaires obtenus à partir de quatre corpus de références.

2 Choix du corpus

Pour effectuer une recherche il suffit d'entrer un corpus et ce dernier doit contenir une collection de document, les requêtes et les jugements. Dragon fournit divers outils pour collecter, préparer, et de lire des collections. Tous les packages et les classes avec le préfixe "dragon.onlinedb" sont liés à cette ligne de fonctionnalités.

L'unité d'une collection est l'article. Un article contient le titre, résumé, le corps, et ainsi de suite. Le lecteur de la collection (collection reader) est un agent qui lit les articles d'une collection. Souvent, les lecteurs de la collection travaillent avec l'analyseur d'article (article parser). Lecteurs de la collection généralement identifier la limite et extraire l'article de la collection et l'analyseur de l'article analyse le texte extrait brut en un article.

Dans notre application, deux types de lecteurs de collection ont été utilisés. La collection de base (basiccollectionreader) mis tout articles dans un seul fichier texte et chaque ligne représente un article dans le cas de la collection génomic04. Le style de collection TREC stocke toute la collection en plusieurs fichiers qui pourraient être dans les sous-dossiers et chaque fichier contient de multiples articles délimité par la balise "DOC" le cas de la collection ap89, CACM et CF. Pour les analyseurs d'articles on a utilisé l'analyseur d'article de base (basicArticleParser) et l'analyseur d'article SGM (SgmArticleParser) qui peut analyser la plupart des articles dans des collections TREC.

3 Fenêtres d'exécution

3.1 Corpus

La figure 5.1 montre comment on peut visualiser le contenu d'un corpus (ex : génomic04), l'élément sélectionné est la collection des documents qui est affiché dans la zone du texte. On peut aussi visualiser les requêtes correspond aux documents et les jugements en sélectionnant le fichier correspond à partir de la liste des fichiers. Comme on peut visualiser le contenu du corpus à partir du menu (File ⟶ Choisir un corpus).

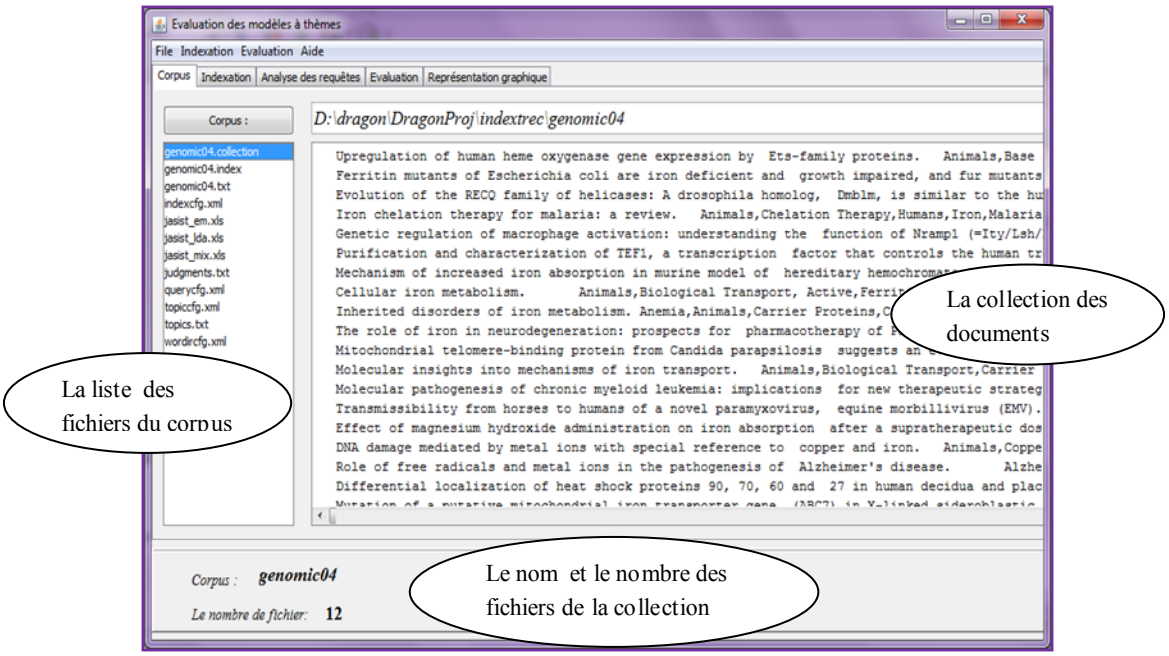


Figure 5.1 : Ouverture et lecture des différents fichiers du corpus «genomic04»

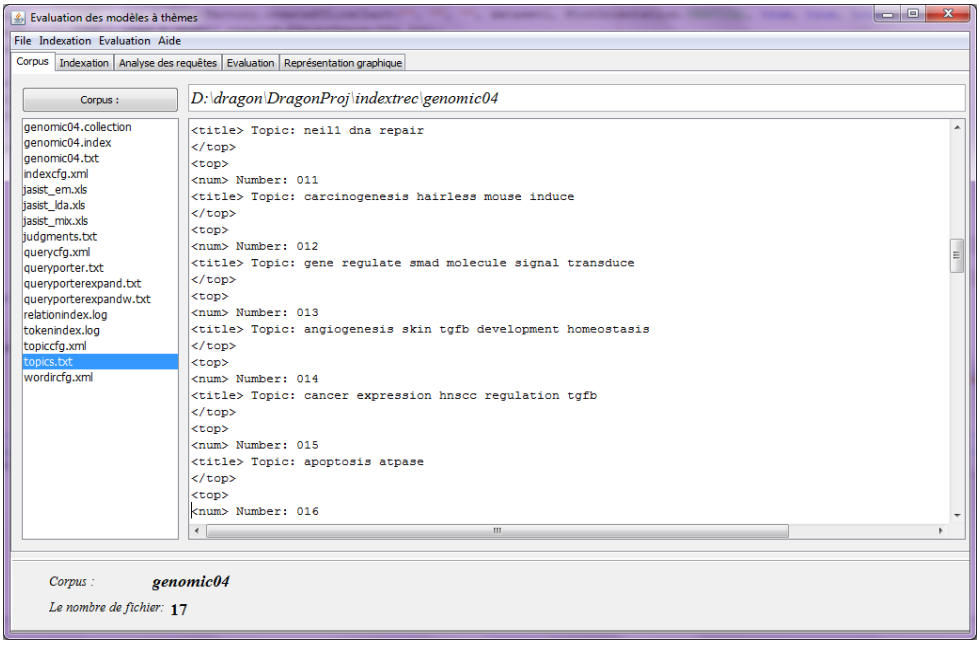


Figure 5.2 : l'ouverture et la lecture de requêtes du corpus «genomic04»

3.2 Indexation

L'indexation est une procédure de conversion des langues naturelles en format structurée et compactée mais avant d'indexer un document il faut appliquer le processus de Traitement Automatique du langage naturel (NLP). Le package NLP fournit les fonctionnalités pour l'extraction des concepts différents ou des relations entre les articles lors de l'indexation. Les concepts peuvent être des mots, des phrases à plusieurs mots individuels, noms propres. Afin de faciliter l'extraction, dragon intègre divers outils de la NLP comme stemmers, une partie des tagueurs de la parole (speech taggers), les analyseurs de phrases (sentence parsers), extracteus de phrase (phrase extractors), et nommé de reconnaissance de l'entité.

Pour faire l'indexation on a utilisé un fichier de configuration basé sur XML « indexcfg ». Ce fichier de configuration est placé dans le chemin corpus et comporte un nœud racine appelé configure. Le nœud racine peut contenir des nœuds d'objets multiples. Un nœud de l'objet correspond à un objet ou une application. Un nœud de l'objet à deux attributs obligatoires (type et id) et un attribut optionnel appelé la classe. Le type du nœud objet est souvent l'interface de l'objet outils et l'identifiant est un nombre entier pour distinguer des objets ganglions différents avec le même type. Ainsi, la combinaison de type et id doit être unique. Par exemple pour indexer le corpus ap89 on a spécifié le lecteur de la collection (SgmArticleParser).

```
<treccollectionreader type="collectionreader" id="1">
        <param name="collectionpath" value="indextrec/ap89"/>
        <paramname="articleparser" value="dragon.onlinedb.trec.SgmArticleParser"/>
</treccollectionreader>
```

Et aussi le type d'indexation qui est L'indexation de base (indexation par mot)

```
<basicindexer type="indexer" id="1">
```

Et le même travail est effectué pour les autres corpus CF et CACM sauf que nom de la collection est changé selon le corpus à indexer.

Il y a d'autre type d'indexation, l'indexation par phrase. L'indexation de base convertit les concepts extraits dans un indice basé sur des entiers et enregistrer les indices de concepts et de leur fréquence dans une matrice doc-terme; la fréquence du document inversé est enregistré dans une matrice terme-doc. L'indexation par phrase est presque la même que l'indexation de base, sauf qu'il sépare un article en phrases d'abord, puis traite chaque phrase comme un document à indexer. En d'autres termes, dans la matrice doc-term entraîné par l'indexation phrase, chaque le document désigne une phrase.

Les deux types d'indexation cités aux dessus sont appelés l'indexation classique, il y a autre type qui est l'indexation en utilisant les modèles à thème (topic model) qui sont le sujet de recherche ces dernière années. Trois modèle à thèmes sont implémenté (LDA lattent dirichlet allocation), modèle aspect et modèle (PLSI) d'unigramme(simple mixture) .

Le résultat de l'indexation des ces modèles sera des distributions des mots par 30 thème avec leurs probabilité c.-à-d. chaque mot a une probabilité d'appartenir à un thème.

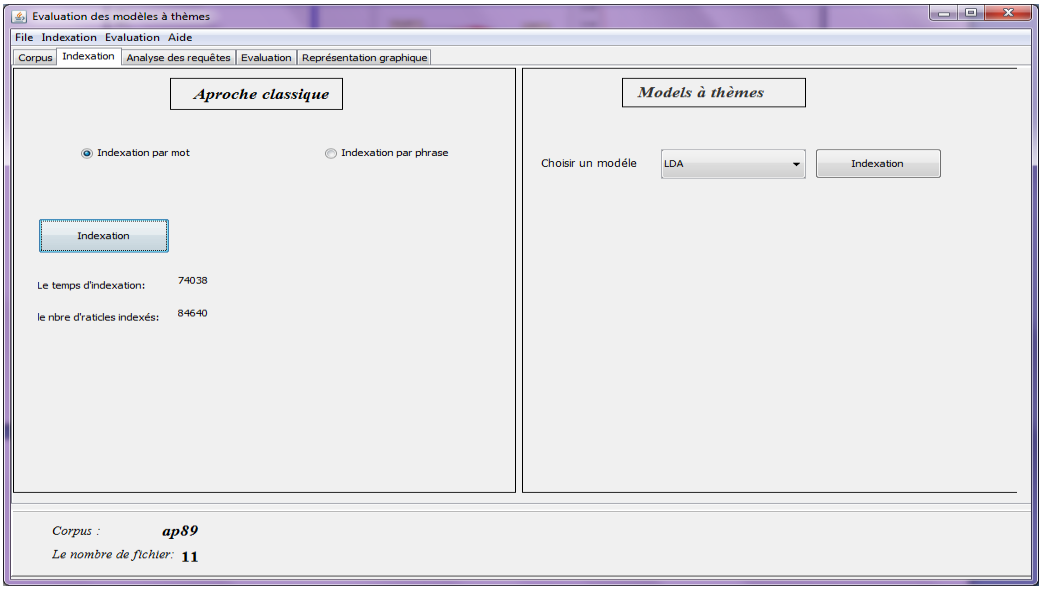


Figure 5.3 : L'indexation par mot du corpus ap89

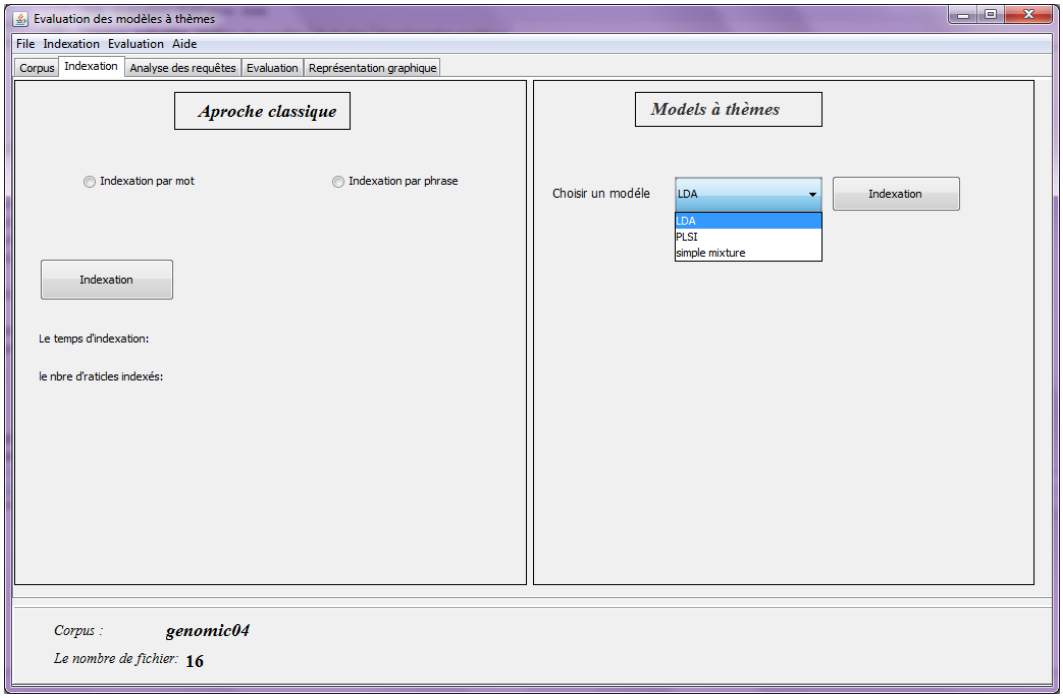
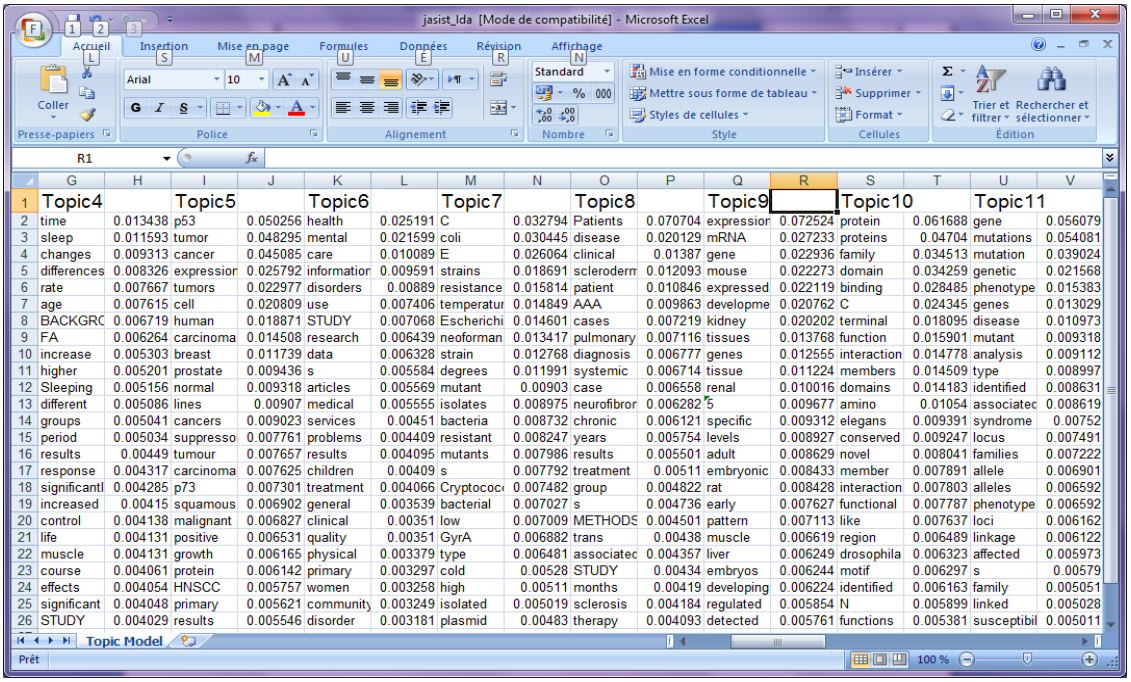


Figure 5.4 : l'indexation en utilisant modèle à thème LDA



Topic4		Topic5		Topic6		Topic7		Topic8		Topic9		Topic10		Topic11	
time	0.013438	p53	0.050256	health	0.025191	C	0.032794	Patients	0.070704	expression	0.072524	protein	0.061688	gene	0.056079
sleep	0.011593	tumor	0.048295	mental	0.021599	coli	0.030445	disease	0.020129	mRNA	0.027233	proteins	0.04704	mutations	0.054081
changes	0.009313	cancer	0.045085	care	0.010089	E	0.026064	clinical	0.01387	gene	0.022936	family	0.034513	mutation	0.039024
differences	0.008326	expressior	0.025792	informatior	0.009591	strains	0.018691	scleroderm	0.012093	mouse	0.022273	domain	0.034259	genetic	0.021568
rate	0.007667	tumors	0.022977	disorders	0.00889	resistance	0.015814	patient	0.010846	expressed	0.022119	binding	0.028485	phenotype	0.015383
age	0.007615	cell	0.020809	use	0.007406	temperatur	0.014849	AAA	0.009863	developme	0.020762	C	0.024345	genes	0.013029
BACKGRC	0.006719	human	0.018871	STUDY	0.007068	Escherichi	0.014601	cases	0.007219	kidney	0.020202	terminal	0.018095	disease	0.010973
FA	0.006264	carcinoma	0.014508	research	0.006439	neoforman	0.013417	pulmonary	0.007116	tissues	0.013768	function	0.015901	mutant	0.009318
increase	0.005303	breast	0.011739	data	0.006328	strain	0.012768	diagnosis	0.006777	genes	0.012555	interaction	0.014778	analysis	0.009112
higher	0.005201	prostate	0.009436	s	0.005584	degrees	0.011991	systemic	0.006714	tissue	0.011224	members	0.014509	type	0.008997
Sleeping	0.005156	normal	0.009318	articles	0.005569	mutant	0.00903	case	0.006558	renal	0.010016	domains	0.014183	identified	0.008631
different	0.005086	lines	0.00907	medical	0.005555	isolates	0.008975	neurofibror	0.006282	5	0.009677	amino	0.01054	associatec	0.008619
groups	0.005041	cancers	0.009023	services	0.00451	bacteria	0.008732	chronic	0.006121	specific	0.009312	elegans	0.009391	syndrome	0.00752
period	0.005034	suppresso	0.007761	problems	0.004409	resistant	0.008247	years	0.005754	levels	0.008927	conserved	0.009247	locus	0.007491
results	0.00449	tumour	0.007657	results	0.004095	mutants	0.007986	results	0.005501	adult	0.008629	novel	0.008041	families	0.007222
response	0.004317	carcinoma	0.007625	children	0.00409	s	0.007792	treatment	0.00511	embryonic	0.008433	member	0.007891	allele	0.006901
significantl	0.004285	p73	0.007301	treatment	0.004066	Cryptococ	0.007482	group	0.004822	rat	0.008428	interaction	0.007803	alleles	0.006592
increased	0.00415	squamous	0.006902	general	0.003539	bacterial	0.007027	s	0.004736	early	0.007627	functional	0.007787	phenotype	0.006592
control	0.004138	malignant	0.006827	clinical	0.00351	low	0.007009	METHODS	0.004501	pattern	0.007113	like	0.007637	loci	0.006162
life	0.004131	positive	0.006531	quality	0.00351	GyrA	0.006882	trans	0.00438	muscle	0.006619	region	0.006489	linkage	0.006122
muscle	0.004131	growth	0.006165	physical	0.003379	type	0.006481	associatec	0.004357	liver	0.006249	drosophila	0.006323	affected	0.005973
course	0.004061	protein	0.006142	primary	0.003297	cold	0.00528	STUDY	0.00434	embryos	0.006244	motif	0.006297	s	0.00579
effects	0.004054	HNSCC	0.005757	women	0.003258	high	0.00511	months	0.00419	developing	0.006224	identified	0.006163	family	0.005051
significant	0.004048	primary	0.005621	community	0.003249	isolated	0.005019	sclerosis	0.004184	regulated	0.005854	N	0.005899	linked	0.005028
STUDY	0.004029	results	0.005546	disorder	0.003181	plasmid	0.00483	therapy	0.004093	detected	0.005761	functions	0.005381	susceptibil	0.005011

Figure 5.5 : la liste des thèmes avec leur distribution des mots

3.3 La génération des requêtes

La deuxième partie nécessaire pour la recherche d'information ad-hoc est la génération des requêtes, cette étape consiste à convertir des requêtes dans un langage naturel à des requêtes structurées. Elles nécessitent aussi Traitement Automatique du langage naturel (NLP) pour l'extraction des concepts différents ou des relations entre les termes des requêtes.

Pour générer les requêtes du corpus ap89 par exemple on a utilisé le fichier de configuration Xml querycgf à partir de la description des requêtes (topic.txt)

```
<earlytrectopicreader type="collectionreader" id="1">

        <param name="topicfile" value="indextrec/ap89/topics.txt"/>

</earlytrectopicreader>

Et le type de la génération (génération de base)

<basicknowledgebase type="knowledgebase" id="1">
```

Il ya aussi d'autre type de génération des requêtes l'expansion de la requête(par phrase) qui génère les termes des requêtes avec un poids pour chaque terme et l'expansion de la requête (par mot) a le même principe avec l'expansion de la requête sauf que la requête est enrichit par d'autre termes avec leur poids.

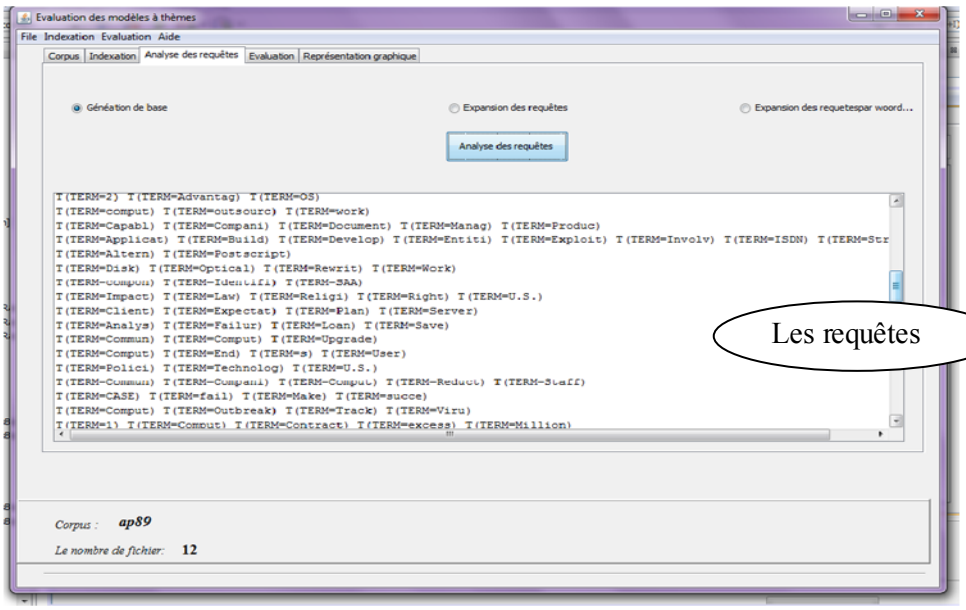


Figure 5.6 : la génération de base des requetes du corpus ap89

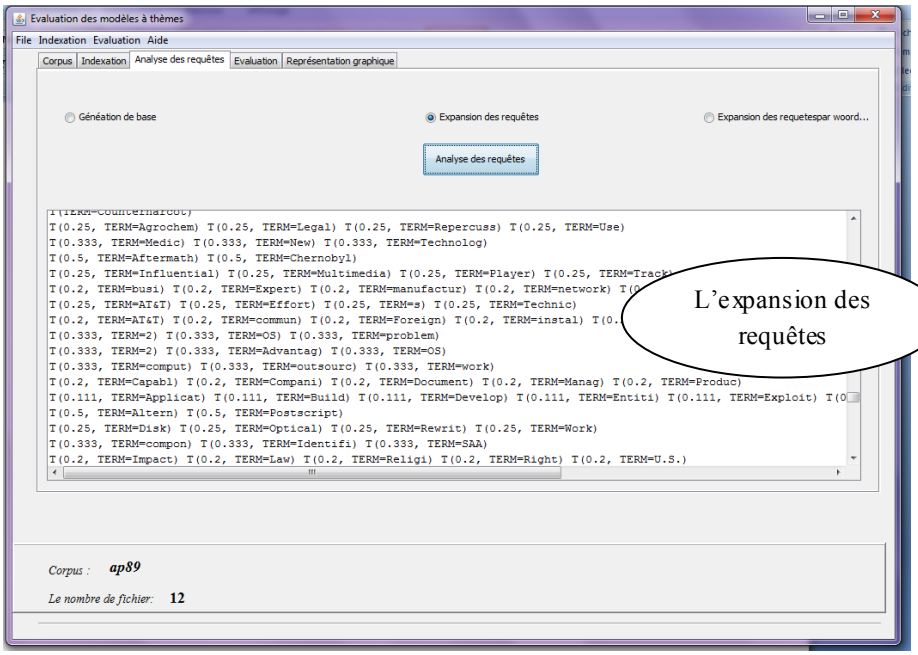


Figure5.7 : l'expansion des requêtes duc corpus ap89

3.4 L'évaluation

Après l'indexation des documents et la génération des requêtes, on peut tout simplement appeler le programme d'évaluation correspondant à évaluer les résultats de recherche. Pour évaluer la performance IR en utilisant le protocole TREC la classe utilisée est TrecEva et encore on a besoin des requêtes (après la génération) et les jugements de pertinence.il y a plusieurs modèle pour l'évaluation le modèle Okapi, le modèle d'extension de la requête, le

modèle d'extension de la requête par phrase et l'évaluation en utilisant la notion de rétroaction (feedbeak)

Les principales métriques d'évaluation en recherche d'information sont implémenté, la précision à 10 c.-à-d combien on a des documents pertinents dans 10 document retrouvés, la précision à 100 c.-à-d combien on a des documents pertinents dans 100 document retrouvés, le rappel à10, le rappel à100, la précision moyenne, interpolation à11point…

Ces mesures sont calculées pour chaque requête du corpus sélectionné et sont affichés dans un tableau. On peut représenter graphiquement les résultats d'évaluation par un histogramme, qui représente la précision à10, la précision à100, le rappel à10, le rappel à100 la précision globale, le rappel globale et la précision moyenne un segment qui représente les mêmes mesures et un graphe qui représente les valeurs de l'interpolation à 11 points.

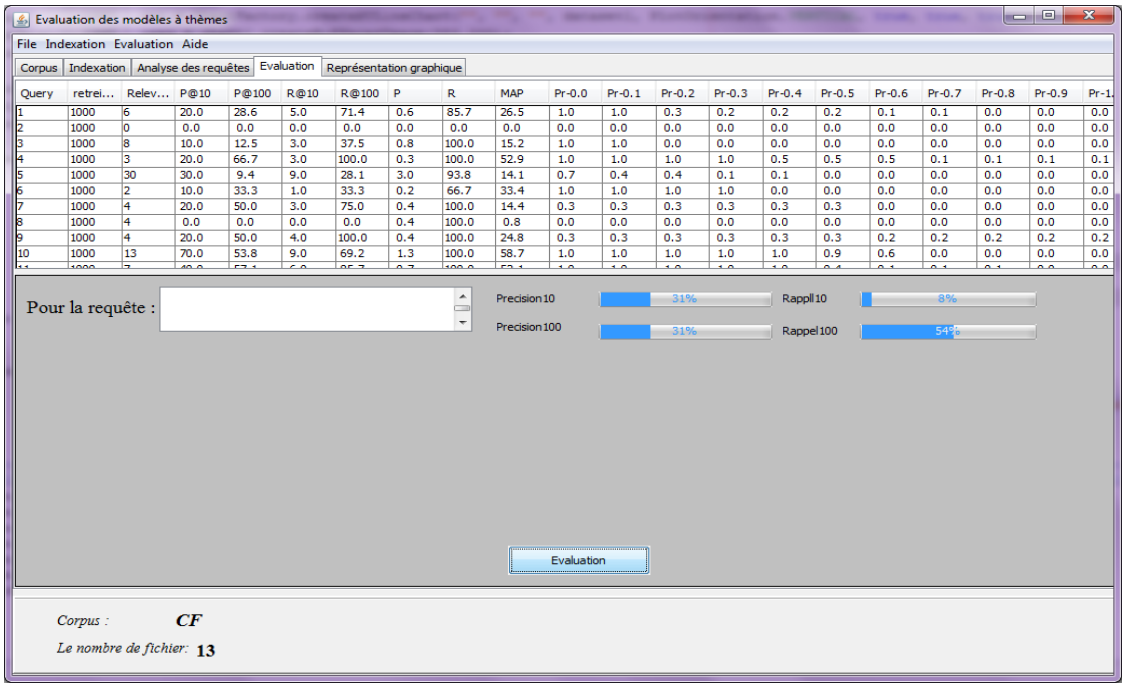


Figure 5.8: les résultats de l'évaluation illustrée dans un tableau

La sélection de n'importe quelle ligne du tableau donne la représentation graphique des mesures calculées ainsi la requête correspond. La figure 5.9 montre cette action.

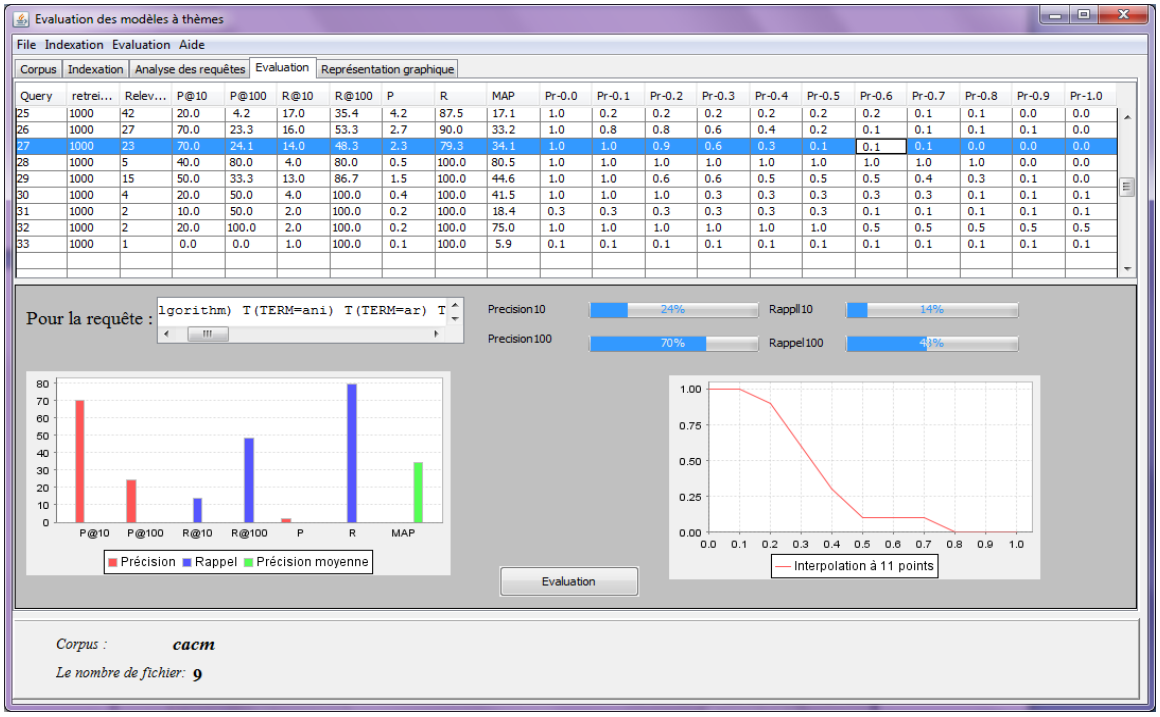


Figure 5.9 : les différentes mesures de l'évaluation du corpus cacm pour la requête 27

La figure 5.10 montre l'affichage graphique pour la requête globale.

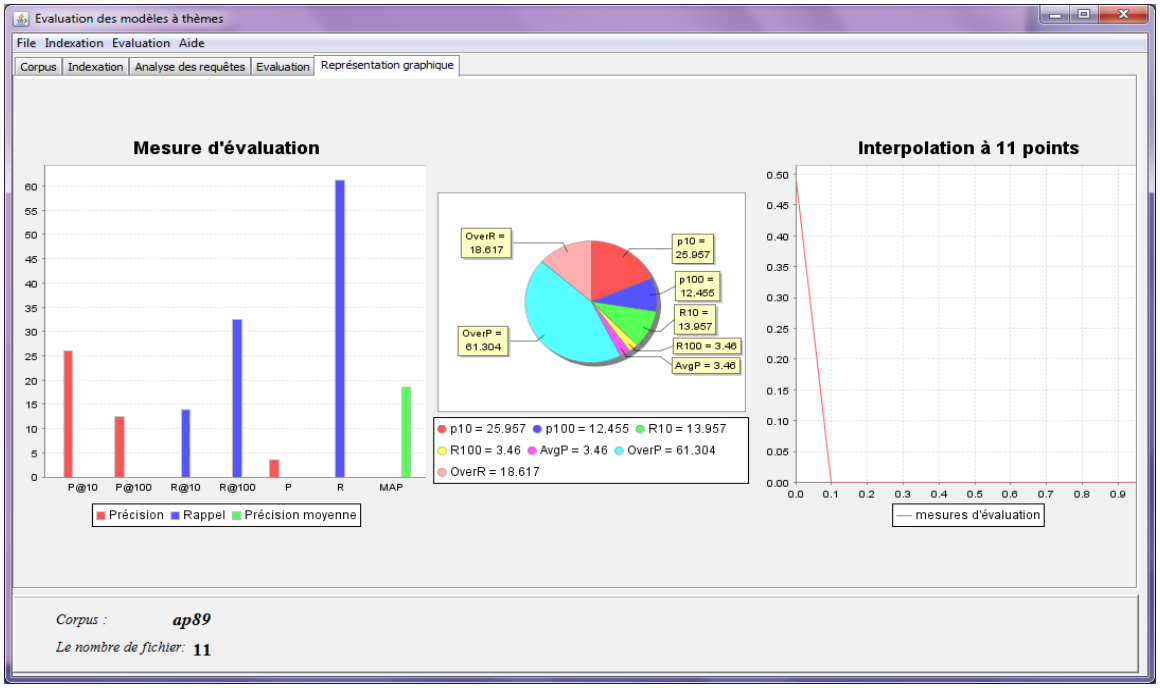


Figure 5.10: les mesures moyennes d'évaluation du corpus ap89

On peut aussi choisir un modèle pour l'évaluation à partir du menu

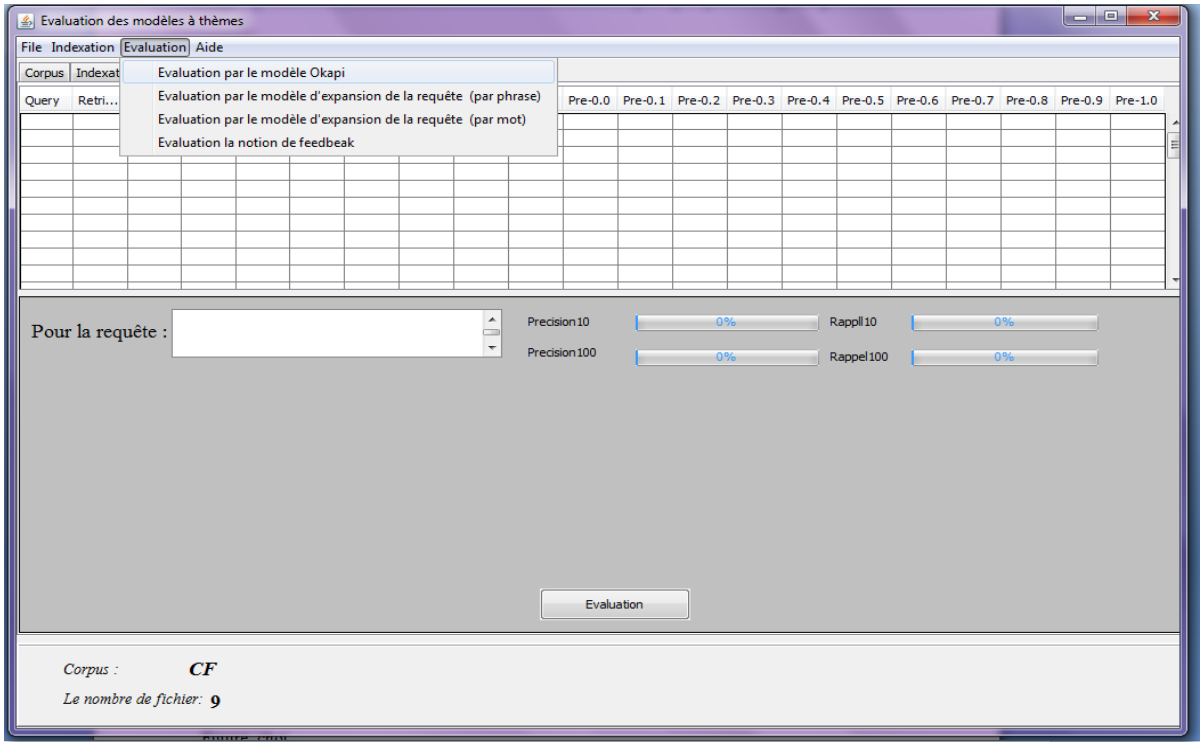


Figure 5.11: choix du modèle pour l'évaluation à partir du menu

4 Résultats

A partir de l'exécution de l'évaluation selon plusieurs modèles d'appariement okapi, feedbeak et le modèle d'expansion de la requête on peut extraire les résultats illustrés dans les tableaux 1, 2,3et 4 et les figures 5.14, 5.15 ,5.16, 5.17 .sur les quatre corpus ***AP89, CF, CACM, Genomic04***.

Corpus	Pertinents	P@10	P@100	R@10	R@100	P	R	MAP
Genomic04	105	36.7	5.5	26.4	22.2	10.5	62.8	24.4
Ap89	35	26.0	12.5	14.0	32.5	3.5	61.3	18.6
Cacm	13	31.4	31.2	8.4	66.6	1.3	91.4	30.2
CF	16	30.7	31.4	7.8	54.1	1.6	91.6	30.2

Tableau5.1: les résultats de l'évaluation par le modèle okapi pour les corpus genomic04, ap89, cacm et CF

Corpus	Pertinents	P@10	P@100	R@10	R@100	P	R	MAP
Genomic04	100	35.4	5.5	26.6	22.2	10.6	0.1	24.3
Ap89	35	24.5	12.1	14.0	32.4	3.5	61.4	18.7
Cacm	12	30.6	32.9	8.8	69.5	1.2	85.3	33.2
CF	17	28.7	29.9	7.8	54.4	1.7	96.5	39.1

Tableau 5.2: résultats d'évaluation des corpus genomic04, ap89, cacm et CFen utilisant l'indexation par phrase et le modèle d'expansion de la requête (par phrase)

Corpus	Pertinents	P@10	P@100	R@10	R@100	P	R	MAP
Genomic04	96	34.4	9.3	25.5	25.4	9.6	0.1	26
Ap89	36	21.7	11.3	13.6	31.5	3.6	62.9	17.4
Cacm	12	30.6	32.9	8.8	69.5	1.2	85.3	33.2
CF	17	28.7	29.9	7.8	54.4	1.7	96.5	39.1

Tableau 5.3: résultats d'évaluation des corpus genomic04, ap89, cacm et CF en utilisant le modèle d'expansion de la requête (par mot)

Corpus	Pertinents	P@10	P@100	R@10	R@100	P	R	MAP
Ap89	40	29.1	11.8	16.4	35.3	4.0	67.3	22.5
Cacm	12	30.2	29.5	9.7	73.9	1.2	88.6	34.7
CF	17	31.1	31.1	8.4	57.4	1.7	97.8	29.3
genomic04	109	38.8	8.8	29.8	28.3	10.9	70.7	32.2

Tableau 5.4: résultats de l'évaluation des corpus en utilisant la notion de feedbeak.

A partir de ces tableaux on peut conclure les résultats suivants :

- le corpus genomic04 a attient plus d'amélioration en appliquant le modèle feedbeak .
- Le corpus ap89 a attient plus d'amélioration en appliquant le modèle feedbeak.
- Le corpus CF a attient plus d'amélioration en appliquant le modèle d'expansion de la requête.
- Le corpus cacm a attient plus d'amélioration en appliquant le modèle d'expansion de la requête.

5 Conclusion

Ce chapitre a été consacré à la mise en œuvre de notre application et à la visualisation des résultats d'évaluation d'un système de recherche en appliquant la recherche sur plusieurs corpus. Les expérimentations menées dans ce projet nous ont permises de valider notre application pour intégrer un corpus, l'indexer (par modèles classiques ou bien par un modèle à thème latent) et enfin d'afficher les résultats d'évaluation. Les représentations graphiques offertes par notre système permettent de mieux comparer et analyser différent modèles de recherche.

CONCLUSION GENERALE

Les travaux développés dans ce mémoire s'inscrivent dans le cadre de l'évaluation des systèmes de recherche d'information sur des collections réelles. Rappelons que le but d'un SRI est de trouver les documents similaires à une requête formulée par un utilisateur. Le SRI, pour répondre au besoin en information d'un utilisateur, doit être en mesure de comparer les documents disponibles avec la requête. Cette comparaison se fait le plus souvent sur la base des attributs communs aux documents et à la requête : les termes.

Nous pensons que le présent travail permet de mieux apprécier les modèles de recherche d'information dans une approche empirique et sur des corpus de références ou même réels. L'intégration du modèle de thème dans un SRI peut être réalisée selon plusieurs approches non encore approuvées. Notre système offre un cadre flexible pour applique telle ou telle approche et ce en permettant d'indexer tout nouveau corpus après un paramétrage standard via un fichier XML.

Chaque corpus sélectionné, peut être par la suite indexé selon un modèle de recherche avant d'être évalué. Cette évaluation est se basée sur quatre métriques principales (précision, rappel, précision moyenne(MAP) et interpolation à 11 points). Les outils d'analyse offerts par notre système permettent d'apprécier les performances d'un modèle sur différents corpus de tests.

REFERENCES BIBLIOGRAPHIQUES

1. Belkin N. J, Croft W. B. 1992. Information retrieval and information filtering: Two sides of the same coin?. CACM, pages : 29-38.
2. Benzécri Jean-Paul et al(1981). Pratique de l'analyse des données, tome 3. Linguistique et lexicologie.Dunod, Paris.
3. Berry. M and P. Young (1995).Using latent semantic indexing for multilanguage information retrieval, Computers and the humanities, Decembre pages : 413-429.
4. Blei, David M. and Ng (2003). Andrew and Jordan, Michael. Latent Dirichlet allocation. Journal of Machine Learning Research.
5. Boughanem. M, C. Chrisment, L. Tamine, (1999).Query space exploration based on genetic algorithms. Information Retrieval Journal. pages : 125-136.
6. Brin Sergey and Lawrence Page, April 1998 . The anatomy of a large-scale hypertextual Web search engine. Source, Computer Networks and ISDN Systems archive, vol. 30, Issue 1-7.
7. Buckley. C, G. Salton(1995). Optimization of relevance feedback weights. Proceedings of ACM SIGIR 95, pages : 351-357.
8. Cho, Garcia-Molina, et al(1998). « Efficient crawling through URL ordering ».
9. Cooper, W, 1988. Getting Beyond Boole, Information Processing & Management, p. 24:3.
10. Deerwester S., Dumais S., Furnas G., Landauer T., Harshman R. (1990), Indexing by Latent Semantic Analysis, Journal of the American Society for Information Science, Vol. 41, Numb. 6, 391-407.
11. Dervin, B. and Nilan M(1983). Information needs and uses in M. Williams(Ed.), Annual Review of Information Science & Technology, vol. 21, p. 1-25. White Plains, NY: Knowledge Industry.
12. Dumais.S(1995). Latent Semantic Indexing (LSI), TREC-3 Report In proceedings of TREC-3, pages : 319-230.
13. Evans.P.(2007). « Scaling and assessment of data quality ».Acta crystallographica. Section D, Biological crystallography, Vol. 62, No. Pt 1. (01 January 2006), pp. 72-82.

14. Fluhr. N, Buckley C(1990). Probabilistic document indexing from relevance feedback data. Proceedings of ACM SIGIR 90, pages : 45-61.
15. Foltz 1990 P. W(1990). Using Latent Semantic Indexing for information filtering. CACM, pages : 40-47.
16. Fox.E(1983). Extending the boolean and vector space models of informationretrieval with p-Norm queries and multiple concept types, PhD Thesis Cornell University
17. Fox, E. & Koll, M, (1988). Partial Enhanced Boolean Retrieval: Experiments with the SMART and SIRE Systems, Information Processing & Management.
18. Frakes W. B (1992). Information retrieval, data structures and algorithms. Prentice Hall, Englewood Cliffs, NJ.
19. Furnas, G. S. Deerwester, S. Dumais, T. Landauer, R. Harshman, L. Streeter, and K. Lochbaum (1988). Information retrieval using a singular value decomposition model of latent semantic structure. In Proceedings of ACM SIGIR 88, p. 465–480.
20. Girolami. Mark , Kaban. Ata(2003). « On an equivalence between PLSI and LDA » Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 433-434.
21. Green, A(1990). What do we mean by users need? British Journal Academic Librarianship 5, p. 65-78.
22. Grossman D, Fieder O.1998. Information retrieval: Algorithms and heuristics. Kluwer Academic Publishers.
23. Harman. D (1992) Relevance feedback revisited. Proceedings ofACM SIGIR 92, pages : 125-132.
24. Heinrich**.** Reinhart**,** Stefan Schuster (2008). **«** The Regulation Of Cellular Systems **»**. posted to book heinrich systems-biology by pmendes .
25. Henzinger, Monika. R(2000). « Improved Data Structures for Fully Dynamic Biconnectivity » SIAM J. Comput., Vol. 29, No. 6. pp. 1761-1815.
26. Hofmann, T(1999) . Probabilistic latent semantic indexing. In Proceedings of SIGIR '99, Berkeley, CA, USA.
27. Hofmann T. (2001), Unsupervised Learning by Probabilistic Latent Semantic Analysis, Machine Learning Journal, Vol. 42, Numb. 1, 177-196.
28. Hoffman. Eric P, Dustin S. Hittel, William E. Kraus(2003), « Skeletal muscle dictates the fibrinolytic state after exercise training in overweight men with characteristics of metabolic syndrome. » .The Journal of physiology, Vol. 548, No. Pt 2.

29. Holscher C., Strube G (2000). « Web Search Behavior of Internet Experts and Newbies», Proceedings of the 9th International Conference on the World Wide Web (www9) - Computer Networks, vol. 33, no 1-6, 2000, p. 337-346, North-Holland Publishing Co.
30. Kuhlthau, Carol (1993). Seeking Meaning: A process approach to Library and information services. Norwood, NJ: Ablex.
31. Kuhlthau, C.; Turock, B.; George, M. & Belvin, R(1990). Validating a model of the search process: a comparison of academic, public and school library users, Library & Information Science Research, p. 12:1.
32. Lawrence Steve, C Lee Giles, Gary William Flake(2000) « Efficient identification of web communities » Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Pages 150-160, Publisher ACM.
33. Madsen P.T, Johnson M, Aguilar Soto N, Zimmer W.M.X, Tyack P.L(2005) Biosonar performance of foraging beaked whales (Mesoplodon densirostris). J. Exp. Biol. 208, page 181–194.
34. Mencser A , K. Gyani, J. I. Kiss(2003) « Telescopic esophageal anastomosis: operative technique, clinical experiences » Article first published online. Volume 16, Issue 4, pages 315–322.
35. Morita. M, Shinoda.Y(1994), Information filtering based on user behavior analysis and best match text retrieval. Proceedings of ACM SIGIR 94, http://www.jaist.ac.jp/jaist/is/labs/shinodalab/ papers/1994/sigir-94.ps, pages : 272-281.
36. Najork. M, and Wiener.J. L. (2001). Breadth-First Crawling Yields High-Quality Pages. In Proceedings of the 10th International World Wide Web Conference, pages 114 118, Hong Kong. Elsevier Science.
37. Nigam K., McCallum A., Thrun S., Mitchell T. (2000), Text Classification from Labeled and Unlabeled Documents using EM, Machine Learning, Vol. 39, Numb. 2/3, 103-134.
38. Pochet B. et Thirion P. (2005). Méthodologie documentaire et formation à l'information. Bulletin d'informations pédagogiques **57**, 15-25. En ligne, http://www.restode.cfwb.be/download/infoped/info57b.pdf.
39. Robertson. Channing R, Robert W. Watkins(1977). « A total internal-reflection technique for the examination of protein adsorption ».Journal of Biomedical Materials Research, Vol. 11, No. 6. , pp. 915-938.
40. Robertson. Channing R, T. D. Braun, H. J. Siegal, N. Beck, et al(1999). « A comparison study of static mapping heuristics for a class of meta-tasks on heterogeneous computing

systems ».Heterogeneous Computing Workshop. (HCW '99) Proceedings. Eighth In Heterogeneous Computing Workshop. pp. 15-29, doi:10.1109/HCW.

41. Salton. G (1970). The SMART retrieval system: Experiments in automatic document processing. Prentic pages : 50-57.
42. Salton.G(1971). A comparison between manual and automatic indexing methods Journal of American Documentation, 20(1) pages : 61-71.
43. Salton. G (1971). The SMART Retrieval System. Prentice Hall.
44. Salton.G(1989). Automatic text processing: The transfromation, analysis and retrieval of information by computer. Addison-Wesley publishing, pages : 85-92.
45. Schutz.A, and Luckmann. T(1973). Structures of the Life World. Northwestern University Press, Evanston, Ill., Ed. ACM, New York, Sept. 1990, p. 45-61.
46. SHNeiderman. B(1998). Designing the User Interface, Addison-Wesley, 3e édition.
47. Sparck Jones, K(1971). Automatic keyword classification for information retrieval. Butterworths, London.
48. Spousta.J, M. Urbánek T. Béhounek, and T. ikola(2007). "Imaging reflectometry in situ," Appl. Opt. **46**, 6309-6313 http://www.opticsinfobase.org/ao/abstract.cfm?URI=ao-46-25 6309.
49. Steyvers. Mark , Griffiths. Tom (2007). « Probabilistic Topic Models ».In Handbook of Latent Semantic Analysis .
50. Tamine. L, Boughanem. M (2000). An Improved genetic algorithm to query optimization. Proceedings of ACM-CIKM 2000, pages : 45-52.
51. Tricot A. (2003). Apprentissage et recherche d'information avec des documents électroniques. Mémoire en vue de l'habilitation à diriger des recherches Université deToulouse. 128 p. En ligne,http://perso.wanadoo.fr/andre.tricot/Tricot_HDR.pdf, consulté le 16 fèvrier 2005.
52. Vignaux Georges (2003). Essai de définition d'un document, document de travail du RTPDOC.
53. Wong .GG , JS Witek, PA Temple, KM Wilkens, AC Leary, DP Luxenberg, SS Jones, EL Brown, RM Kay(1985). « Human GM-CSF: molecular cloning of the complementary DNA and purification of the natural and recombinant proteins ». Vol. 228 no. 4701 pp. 810-815.
54. Xing Wei and W. Bruce Croft, (2006). «LDA-Based Document Models for Ad-hoc Retrieval «University de Massachusetts Amherst 140 Governors Drive Amherst, MA 01003.

55. Yates R. B. Neto , R,1999.Modern Information Retrieval. ACM Press, Addison Wesley, pages : 70-77.
56. Yi Xing , Allan James(2009). « A Comparative Study of Utilizing Topic Models for Information Retrieval » .In Advances in Information Retrieval, Vol. 5478 (2009), pp. 29-41.