UNIVERSITE BLIDA I

Faculté des Sciences
Département d'Informatique

MEMOIRE DE MAGISTER

Spécialité : Systèmes d'Informations et de Connaissances

OLAP ET RECHERCHE D'INFORMATION : OPERATEURS DE SUMMARIZATION DE DOCUMENTS ET DE REQUETES OLAP SUR LES TEXTES

Par

BOUSLAH Meriem

Devant le jury composé de

S. OUKID KHOUAS	Maître de Conférences A, U. de Blida	Présidente
Z. ALIMAZIGHI	Professeur, U.S.T.H.B, Alger	Examinateur
D. ZEGOUR	Professeur, ESI, Alger	Examinateur
N. BENBLIDIA	Maître de Conférences A, U. de Blida	Promotrice
O.BOUSSAID	Professeur, U. Lyon 2	Co-promoteur
W. HIDOUCI	Maître de Conférences A, ESI, Alger	Invité

Blida, Janvier 2014

RESUME

De nos jours, les technologies entrepôt de données et OLAP (On Line Analytical Processing) permettent d'analyser et d'interroger d'importantes masses de données circulant au niveau des entreprises afin d'améliorer le processus de décision. Par ailleurs, l'agrégation ou la summarization est l'une des plus importantes opérations d'analyse qu'offrent ces technologies. De par leurs caractéristiques, les entrepôts de données et les opérateurs OLAP sont capables de manipuler et d'agréger seulement les données numériques.

L'objectif de ce travail est de permettre la summarization de données textuelles au sein d'un environnement OLAP en se basant sur des techniques de la Recherche d'Information. Pour ce faire, nous définissons d'abord un modèle de données multidimensionnel en constellation basé sur des mesures textuelles pour abriter les données textuelles. Ensuite, nous proposons deux opérateurs de summarization de données textuelles, *Term_Up* et *Term_Down*, basés sur une forme adaptée de la formule de pondération *TF-IDF*. Une mesure pour l'évaluation de la qualité de la summarization *T_Mesure* est également proposée. Finalement, des expérimentations réalisées sur un entrepôt d'articles de presse confirment l'intérêt de l'approche proposée.

ملخص

تحتل تقنيات مخازن المعطيات و عمليات التحليل الآني أهمية كبيرة على مستوى المؤسسات لما لها من قدرة على تحليل و مساءلة كميات كبيرة من المعطيات في آن واحد، حيث تعتبر عملية التلخيص من أهم عمليات التحليل التي تضمنها هذه التقنيات. من جهة ثانية, تطبيق هذه التقنيات محصور حاليا في المعطيات الرقمية .

هدف هذا العمل, هو تمكين تقنيات مخازن المعطيات و أدوات التحليل من معالجة المعطيات النصية كمعالجتها للمعطيات الرقمية. في هذا الإطار، مقترحنا يتمثل في نموذج للمعطيات متعدد الأبعاد قادر على دمج المعطيات النصية. بالموازاة مع ذلك، نحن نقترح أداتين جديدتين للتحليل الآني قادرتين على تلخيص المعطيات النصية. من جهة أخرى، نقترح مقياس لتقييم عملية التلخيص. في النهاية، تمت الاستعانة بمخزن لمقالات صحفية لتقييم المقترحات المقدمة خلال هذا العمل.

ABSTRACT

As large amount of unstructured text becomes available in multidimensional databases, it is increasingly important to support efficient online analysis of text data. A fundamental problem for analysis of multidimensional text databases is the absence of suitable operators. The objective of this work is to allow the summarization of textual data in an OLAP environment using Information Retrieval techniques. To do this, we first define a multidimensional data model based on textual measures. Then, we propose two operators for the summarization of textual data : *Term_Up* and *Term_Down*. These two operators are based on an adapted form of the weighting formula TF-IDF. A measure for evaluating the quality of the summarization *T_Mesure* is also proposed. Finally, experiments performed on a newspaper warehouse confirm the interest of the proposed approach.

REMERCIEMENTS

Tout d'abord, je tiens à exprimer mes plus vifs remerciements et ma gratitude à Madame BENBLIDIA Nadjia et à Monsieur BOUSSAID Omar pour leur encadrement continu, et pour les orientations et les remarques constructives qu'ils m'ont fournies durant toute la période de mon travail. Je les remercie également pour la confiance qu'ils m'ont accordée et pour la grande liberté d'idées et de travail qu'ils m'ont donnée.

Je tiens également à remercier profondément Monsieur ASABAT Ali pour tous ses soutiens et encouragements tout au long de ma formation.

Mes remerciements et mon profond respect vont aux membres du jury pour le temps et l'attention consacrés à mon travail.

Je remercie aussi tous ceux qui m'ont apporté leur savoir et ont contribué à ma formation de magister. Qu'ils trouvent ici l'expression de ma gratitude et de mon profond respect.

Je remercie mes parents, mon mari, mes frères et sœurs pour leur contribution, leur soutien et leur patience.

Enfin, je tiens à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail, plus particulièrement : Foued et Moulida.

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1	Processus de la Recherche d'Information.....	16
Figure 1.2	Processus d'indexation.....	19
Figure 1.3	Structure du fichier inverse.....	26
Figure 1.4	Partitionnement de la collection de documents pour une requête	32
Figure 1.5	Courbe de rappel et de précision pour l'exemple du tableau 1.1	34
Figure 1.6	Courbe simplifiée de rappel et de précision pour une requête.....	35
Figure 2.1	Analyse de documents gouvernementaux	42
Figure 2.2	Exemple d'une requête d'interrogation de l'entrepôt de données	43
Figure 2.3	Architecture de l'entrepôt de données *R-Cube* .....……..	46
Figure 2.4	Base multidimensionnelle de texte .....…….....	47
Figure 2.5	Hiérarchie des termes du domaine .....…….....	47
Figure 2.6	Analyse d'articles scientifiques en utilisant des mesures textuelles…….....	48
Figure 4.1	Processus de summarization.....	61
Figure 4.2	La Stop liste proposée par Fox	67
Figure 5.1	Exemple de requête-type sur un entrepôt d'articles de presse représentée par une table multidimensionnelle.....	92
Figure 5.2	Modèle de données pour l'entrepôt d'articles de presse.....	94
Figure 5.3	Déroulement du processus de summarization.....	98
Figure 6.1	Architecture de l'application « *Summarize* ».....	104
Figure 6.2	Modèle multidimensionnel de données pour l'entrepôt d'articles de presse.....	105
Figure 6.3	Représentation des résultats de pondération de la collection 1 par un nuage de points.....	108
Figure 6.4	Représentation des résultats de pondération de la collection 2 par un nuage de points.....	109
Figure 6.5	Représentation graphique des résultats obtenus pour QS(k) des collections 1 et 2.....	112

TABLE DES MATIERES

RESUME……..... 2

REMERCIEMENTS…….……..... 5

TABLE DES MATIERES……..…….... 6

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX……..... 8

INTRODUCTION ……..…….... .. 10

1. INTRODUCTION A LA RECHERCHE D'INFORMATION ……..……..... 15

1.1 Introduction…….... 15
1.2 Concepts de base de la recherche d'information ……..…..... 15
1.3 Modèles de recherche d'information..... 26
1.4 Evaluation des systèmes de recherche d'information ……. 30
1.5 Conclusion ……..... 39

2. LES APPROCHES DE COUPLAGE OLAP-RI……..…….... 40

2.1 Introduction……..... 40
2.2 les approches de couplage OLAP-RI……..…….... 40
2.2.1 Systèmes de Recherche d'Informations multidimensionnels……..... 41
2.2.2 Entrepôt de données textuel..... 44
2.3 Conclusion..... 49

3. RESSOURCES ET SIMILARITE SEMANTIQUES……..... 50

3.1Introduction……..... 50
3.2 Ressources sémantiques……... 50
3.3 Similarité sémantique…….. 55
3.4 Conclusion……..... 58

4. SUMMARIZATION PROPOSEE..... 60

4.1 Introduction..... 60
4.2 Processus de summarization..... 60
4.3 Algorithme de summarization..... 81
4.4 Conclusion..... 87

5. SUMMARIZATION AU SEIN DE L'ENTREPOT DE DONNEES..... 88

5.1 Introduction..... 88
5.2 Entrepôt de données classiques..... 88
5.3 La summarization au sein d'un entrepôt de données..... 90
5.4 Cas d'étude : Entrepôt d'articles de presse..... 92
5.5 Formalisation de l'entrepôt de données textuel..... 94
5.6 Déroulement du processus de summarization..... 97
5.7 Conclusion..... 102

6. IMPLEMENTATION : RESULTATS ET DISCUSSION..... 103

6.1 Introduction..... 103
6.2 Approche retenue pour l'implémentation..... 103
6.3 Expériences et résultats de summarization..... 105
6.4 Conclusion..... 113

7. CONCLUSION..... 114

REFERENCES..... 117

INTRODUCTION

- ***Contexte général du travail***

Le développement rapide des technologies de l'information et l'avènement d'Internet et des réseaux, ont accru le volume d'informations disponibles au sein des entreprises de manière considérable. Une bonne prise de décision nécessite l'exploitation de toutes ces informations qui transitent au niveau de l'entreprise, mais le volume considérable de ces informations et les limites des bases de données traditionnelles rendent leur exploitation une tâche délicate et peu rentable. L'entrepôt de données constitue une bonne alternative dans la mesure où il dispose des outils nécessaires au traitement de grandes masses de données [Sullivan, 2001].

Les entrepôts de données ont été introduits au début des années 1990 dans le but d'améliorer le processus de décision au sein de l'entreprise [Codd et al, 1993]. Un entrepôt de données est un système pour le stockage et l'analyse de données. Il se base sur deux concepts principaux, le fait et la dimension. Le fait représente le sujet que nous voulons analyser ; il est mesuré à l'aide d'un ensemble d'indicateurs numériques que nous appelons mesures. Quant à la dimension, elle représente l'axe d'analyse selon lequel nous voulons analyser les données. Une dimension se caractérise par un ensemble de descripteurs ; elle peut être hiérarchisée (ayant plusieurs niveaux de détail) ou non.

De nos jours, les technologies entrepôt de données et OLAP (On Line Analytical Processing) permettent d'analyser et d'interroger d'importantes masses de données circulant au niveau des entreprises [Colliat et al, 1996]. En effet, les données d'un entrepôt de données classique proviennent essentiellement des bases de données opérationnelles de l'entreprise ; elles sont extraites à partir des bases de données et graduellement intégrées au sein de l'entrepôt de données. L'extraction et l'intégration des données dans un entrepôt de données passent par plusieurs étapes. D'abord, les données extraites des bases de données doivent être normalisées et homogénéisées. Ensuite, ces données sont stockées dans l'entrepôt de données en respectant l'organisation multidimensionnelle de l'entrepôt. Finalement, plusieurs types de traitement des données sont réalisés, entre autre, par la technologie OLAP. Cette dernière permet l'exploitation et l'analyse des données de l'entrepôt à travers un ensemble d'opérateurs.

En effet, différents types d'analyses peuvent être réalisés sur les données d'un entrepôt de données, notamment l'analyse en ligne OLAP. Cette dernière offre la possibilité de naviguer au travers les données afin de mieux les observer et ainsi les comprendre. L'analyse des données de l'entrepôt se base sur une navigation exploratrice qui permet de décrire, d'expliquer ou de justifier certains faits en offrant plusieurs vues de la même donnée et en permettant de la synthétiser selon les besoins de l'utilisateur.

Ainsi, les opérateurs OLAP permettent différentes opérations d'analyses au sein d'un entrepôt de données telles que le groupement ou l'agrégation. L'agrégation constitue une des plus intéressantes opérations d'analyse qu'offre la technologie OLAP. Elle permet un traitement synthétique des données à l'aide des deux opérateurs d'agrégation : *Roll-Up* et *Drill-Down*. Il s'agit de deux opérateurs qui agissent à travers la granularité au sein d'un entrepôt de données ; ils permettent de naviguer au sein de l'entrepôt en passant des données les plus détaillées aux données plus synthétisées(le forage vers le haut) et inversement (le forage vers le bas).

- ***Problématique*** :

La robustesse et la fiabilité de l'entrepôt de données placent actuellement ce dernier au cœur des systèmes d'informations décisionnels [Chaudhuri, 2011]. En effet, les entrepôts de données viennent de connaitre un essor important au sein des entreprises. Par ailleurs, leur prolifération a permet de mieux les concevoir et de les exploiter. L'exploitation d'entrepôts de données basés sur une architecture multidimensionnelle et des mesures numériques est une tâche bien maitrisée actuellement [Sullivan, 2001]. En revanche, seules les données numériques peuvent être exploitées par ces entrepôts de données, tout autre type de données (texte, image, .etc.) est non exploitable. Selon Tseng et Chou, les données numériques exploitables par l'entrepôt de données ne représentent que 20% des données circulant au niveau d'une entreprise [Tseng et Chou, 2006]. Les 80% restantes, généralement contenues dans des documents électroniques et constituées principalement par des données textuelles, restent hors de la portée de l'entrepôt de données et par conséquent du processus de prise de décision.

Pour que la donnée textuelle soit exploitable au niveau de l'entrepôt de données, il faut absolument étendre les approches classiques de construction et d'exploitation des entrepôts de données. Cette adaptation concernera deux volets :

- Le stockage des données textuelles au sein de l'entrepôt de données : l'architecture multidimensionnelle classique d'un entrepôt de données n'est adéquate qu'aux données

numériques, l'intégration de données textuelles au sein de l'entrepôt nécessite une approche de modélisation multidimensionnelle adaptée aux données textuelles.

- L'analyse OLAP de données textuelles : la technologie OLAP permet différents types d'analyse au sein de l'entrepôt grâce à un ensemble d'opérateurs. Ces opérateurs, basés pour la plupart sur des fonctions arithmétiques, ne sont appropriés qu'aux données numériques. Effectuer des analyses OLAP sur des données textuelles nécessite la mise en place de nouveaux opérateurs OLAP adaptés aux données textuelles.

Par ailleurs, la technologie OLAP nous offre deux opérateurs pour l'agrégation au sein d'un entrepôt de données : *Roll-Up* et *Drill-Down*. Ces deux opérateurs nous permettent de naviguer au travers des hiérarchies et de synthétiser les données numériques en fonction des requêtes utilisateurs. L'agrégation (forage vers le haut) consiste à représenter les données à un niveau de granularité supérieur selon la hiérarchie définie pour la dimension. Une fonction d'agrégation (somme, moyenne,…) indique comment sont calculées les valeurs du niveau supérieur de la hiérarchie à partir de celles du niveau inférieur. Les fonctions d'agrégation OLAP sont des fonctions arithmétiques (somme, moyenne,..) en adéquation avec la nature numérique des données (mesures) traitées, elles ne permettent pas l'agrégation de données textuelles.

Dans ce travail, nous nous intéressons à étendre la modélisation multidimensionnelle et les opérateurs OLAP aux données textuelles en se basant sur des techniques de la Recherche d'Information. Plus précisément, notre objectif est de permettre la summarization (l'agrégation) de données textuelles par des opérateurs OLAP au sein de l'entrepôt de données. Cette summarization est réalisée en permettant l'intégration de données textuelles à travers une approche de modélisation adaptée et en créant de nouveaux opérateurs basés sur des fonctions capables de résumer du texte.

- ***Approche proposée :***

Pour que la donnée textuelle soit exploitable notamment agrégée au niveau de l'entrepôt de données, il faut absolument étendre les approches classiques d'entrepôts de données et les opérateurs d'agrégation OLAP adaptés jusqu'à présent qu'aux données numériques. L'adaptation des entrepôts de données aux données textuelles concernera deux volets:

- L'intégration des données textuelles au sein de l'entrepôt de données : elle nécessite une représentation conceptuelle multidimensionnelle adaptée aux données textuelles.
- L'agrégation de données textuelles : elle nécessite des opérateurs d'agréation OLAP adaptés aux données textuelles.

Ainsi, nos objectifs dans ce travail se résument à :

- La proposition d'un processus de summarization sémantique de données textuelles permettant l'agrégation d'un contenu textuel par ses *k* termes les plus représentatifs. Ce processus se base sur une forme adaptée de la formule de pondération *TF-IDF*, ainsi qu'un ajustement sémantique des poids des termes. Une mesure de la qualité de la summarization effectuée est également proposée.
- La définition d'un modèle de données multidimensionnel adapté aux données textuelles en s'inspirant du modèle de données proposé par [Olivier, 2009]. En effet, le modèle proposé sera doté de mesures textuelles qui vont nous permettre d'intégrer, d'analyser et de manipuler les données textuelles.
- La proposition de deux opérateurs de summarization : *Term_Up* et *Term_Down* basé sur le processus de summarization proposé. *Term_Up* et *Term_Down* sont deux opérateurs de forage à travers la granularité. *Term_Up* permet d'agréger un contenu textuel en *k* termes, tandis que *Term_Down* permet d'avoir la vue détaillée.

D'autre part, afin de mieux illustrer l'approche proposée, nous allons nous intéresser à un cas d'étude bien précis : l'analyse des articles de presse. Des expériences sur des collections tirées à partir d'un entrepôt d'articles de presses seront menées. Ces expériences vont permettre l'évaluation du processus de summarization proposé.

- ***Organisation du mémoire :***

Ce mémoire s'articule autour de trois parties : un état de l'art global, une description de l'approche proposée et une présentation des expérimentations effectuées. D'abord, le premier chapitre d'état de l'art sera consacré à la présentation des concepts de base de la Recherche d'Information. Le deuxième chapitre présentera une revue sur les travaux combinant la Recherche d'Information et la technologie OLAP afin de permettre l'intégration et la manipulation de données textuelles dans un environnement OLAP. Comme il s'agit d'une summarization sémantique, le troisième chapitre de l'état de l'art consistera à une brève présentation des ressources sémantiques existantes ainsi qu'aux mesures de calcul de la similarité sémantique les plus répandues.

La deuxième partie de ce mémoire, constituée du quatrième et du cinquième chapitre, sera consacrée à la présentation de l'approche de summarization proposée. Ainsi, le quatrième chapitre consistera à la présentation des différentes étapes du processus de summarization proposé, et à sa formalisation à l'aide de pseudos-algorithmes. Le cinquième chapitre sera

consacré à la présentation de la summarization au sein de l'entrepôt de données avec la proposition d'un modèle de données adapté aux données textuelles et de deux nouveaux opérateurs pour le forage en haut et en bas.

Finalement, la troisième partie de ce mémoire consistera à la présentation des expérimentations effectuées afin d'évaluer la qualité de la summarization proposée. Nous finirons ce mémoire par une conclusion.

CHAPITRE 1
INTRODUCTION A LA RECHERCHE D'INFORMATION

1.1 Introduction

Historiquement, la Recherche d'Information est née d'un besoin de gestion de documents au niveau des bibliothèques. Les spécialistes du domaine : bibliothécaires, documentalistes …etc. veillaient d'une part au stock des documents et leur pérennité, et d'autre part à assurer l'accès à ces documents. Avec l'avènement de l'Internet, et l'explosion de la quantité de documents mis à la disposition du grand public, le besoin de systèmes automatiques pour le stockage et la gestion automatique des documents s'est fait sentir. La Recherche d'Information va apporter les méthodes et les outils nécessaires à la réalisation de tels systèmes.

Nous pouvons définir la Recherche d'Information comme la branche de l'informatique qui s'intéresse à l'organisation, au stockage et à la sélection d'informations répondant aux besoins des utilisateurs [Salton, 1970], [Salton, 1984]. Ce domaine manipule différents concepts : le besoin de l'utilisateur en information, la requête, le document, les modèles de Recherche d'Information, la pertinence, etc.

L'objectif de ce chapitre est de présenter les concepts de base de la Recherche d'Information ainsi que l'évaluation des SRI (Système de Recherche d'Information). Nous allons commencer par la présentation du processus de la Recherche d'Information et celui de l'indexation. Ensuite nous allons passer en revue les principaux modèles de la Recherche d'Information. Finalement, nous présenterons les principales mesures d'évaluation des SRI.

1.2 Concepts de base de la recherche d'information

1.2.1 Processus de la recherche d'information

Le processus de Recherche d'Information peut être décrit comme suit (Figure1.1) : un utilisateur formule une requête exprimant son besoin en information et la soumet à un SRI. Ce dernier doit répondre à cette requête par une liste de documents en fouinant dans sa base documentaire. D'abord, le système procède à l'indexation de la requête afin que cette dernière ait la même représentation que les documents de sa base d'information déjà indexés, puis un appariement entre la requête indexée et l'ensemble des documents du SRI est réalisé. Les

documents les plus similaires à la requête de l'utilisateur vont être sélectionnés, et présentés à l'utilisateur en guise de réponse à sa requête.

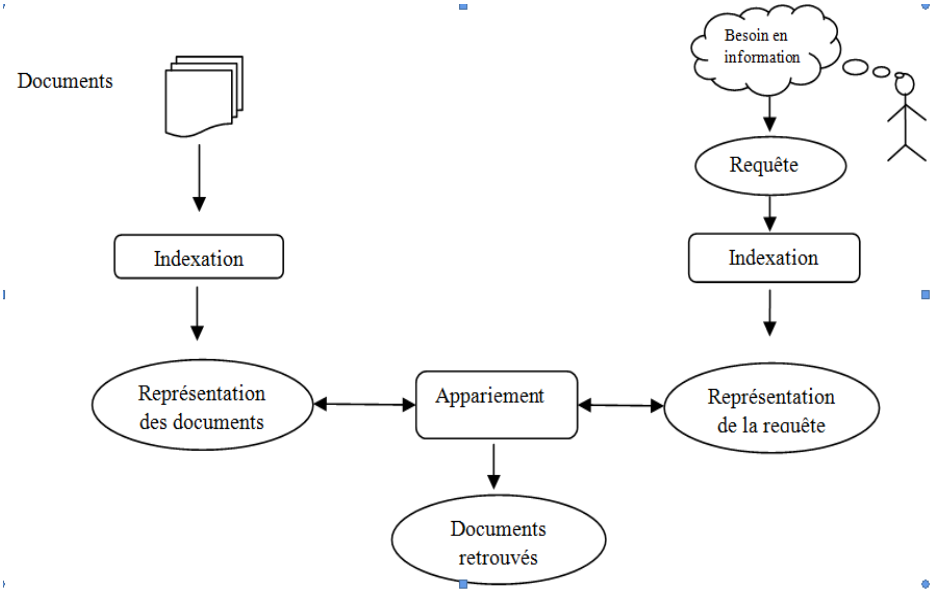

Figure1.1 : Processus de la Recherche d'Information.

Plus généralement, nous pouvons définir le processus de Recherche d'Information comme le processus qui permet de mettre en correspondance l'ensemble des informations disponibles dans la base des documents du SRI et le besoin de l'utilisateur en information formulé à l'aide d'une requête. Cette correspondance est réalisée à l'aide d'un ensemble de mécanismes de recherche et d'appariement qui permettent de sélectionner les documents pertinents (pour le SRI) à la requête formulée.

Le processus de Recherche d'Information comprend plusieurs concepts : le document, la requête, l'indexation, les modèles de Recherche d'Information. Nous allons définir les données d'entrée du processus de la Recherche d'Information, à savoir : les documents et la requête. Par la suite, nous présenterons les autres concepts.

Document : c'est l'élément élémentaire d'une base de documents. Il peut être un texte, un fragment de texte, une image, un extrait audio ou vidéo, etc. Globalement, on appelle document tout élément qui peut représenter une réponse à une requête utilisateur. Selon la structure, nous pouvons distinguer les documents structurés, documents semi-structurés et documents non structurés.

Requête : elle exprime un besoin en information d'un utilisateur. Formulée par ce dernier, c'est elle qui déclenche le processus de la Recherche d'Information. La littérature propose divers types de langage d'interrogation pour la formulation d'une requête, les plus répandus sont [Nassr, 2002] :

- ✓ Interrogation en langage naturel ou quasi naturel : l'utilisateur exprime sa requête en langage libre (langage naturel) sous forme de mots clés. Le système se charge de traduire ces mots clés en une requête compréhensible par le système.

- ✓ Interrogation en langage booléen : l'utilisateur exprime sa requête sous forme d'un ensemble de termes reliés entre eux par des opérateurs booléens (ET, OU, NON, etc.). Plusieurs moteurs de recherche, se basent sur ce mode d'interrogation, parmi les plus connus nous pouvons citer : Altavista, Google, etc.

- ✓ Interrogation en langage graphique : afin d'aider l'utilisateur dans la formulation de sa requête, une vue d'ensemble de la base d'information constituée de l'ensemble des termes de cette base est proposée à l'utilisateur à travers une interface graphique.

1.2.2 Indexation

Les documents sont des objets complexes qui ne sont pas simples à manipuler. Afin de simplifier la manipulation du contenu de ces documents et de diminuer leur complexité, nous utilisons l'opération d'indexation qui vise à créer une représentation interne du document. Cette représentation reflète le contenu du document ; elle va servir à comparer ce contenu à celui d'une requête. En effet, la finalité de l'indexation est de permettre de repérer les documents dont le contenu correspond au besoin formulé par une requête donnée.

Le processus d'indexation consiste à extraire un ensemble de termes représentatifs appelés descripteurs du document. Les descripteurs peuvent être des termes simples (composés d'un seul mot) ou des termes multi-mots ; ils constituent le langage d'indexation. Ces termes descripteurs doivent refléter le plus fidèlement possible le contenu du document ; en effet, la qualité de l'opération de restitution des documents réponse à une requête dépend fortement de choix des descripteurs. Par ailleurs, l'indexation ne sert pas seulement à extraire les descripteurs mais aussi à évaluer leur représentativité au sein du document. L'indexation tend à répondre à deux problèmes essentiels : le choix des termes descripteurs d'un document et l'évaluation de leurs degrés de représentativité au sein de ce document. Cette évaluation de représentativité ou de discrimination du descripteur est réalisée en utilisant des techniques de pondération. Il existe deux façons pour réaliser l'indexation : manuelle ou automatique.

1.2.2.1 Indexation manuelle

L'indexation manuelle est réalisée par un spécialiste du domaine ou par un expert documentaliste, ce dernier doit analyser tout le texte afin de choisir les termes les plus représentatifs comme des descripteurs du texte. Par sa nature humaine, l'indexation manuelle est pertinente, cohérente et profonde. Néanmoins, elle possède de nombreux inconvénients : elle pose le problème du vocabulaire utilisé, elle dépend fortement des connaissances de l'indexeur au sujet que traite le texte, et enfin elle devient impraticable si le nombre de documents à indexer est important car elle nécessite la lecture de l'intégralité des documents.

1.2.2.2 Indexation automatique

L'indexation automatique regroupe un ensemble de traitements automatisés qui ont pour objectif l'extraction d'une liste de descripteurs textuels avec l'évaluation de leur pouvoir de discrimination. Nous avons deux types de traitements pour l'indexation automatique : les traitements statistiques et les traitements linguistiques. Le traitement statistique [Salton et Yang, 1973] [Rijsbergen, 1979] est un traitement qui se base sur la fréquence des termes dans le document, quant au traitement linguistique [Sheridan et Smeaton, 1992], il se base sur des techniques de traitement de langage naturel comme la tokenisation et l'analyse syntaxique.

Le choix entre une indexation automatique et une indexation manuelle dépend d'un certain nombre de paramètres dont le plus important est le volume de la collection de documents à indexer. Une étude comparative entre les deux types d'indexation [Anderson et Pérez-Carballo, 2001] montre que les avantages et les inconvénients des deux types d'indexation tendent à s'équilibrer ; par ailleurs, le domaine et le volume de la collection déterminent le type d'indexation à privilégier.

D'une manière générale, l'indexation automatique consiste aux étapes suivantes : analyse lexicale, élimination de mots vides, lemmatisation, pondération, et création de l'index (Figure 1.2).

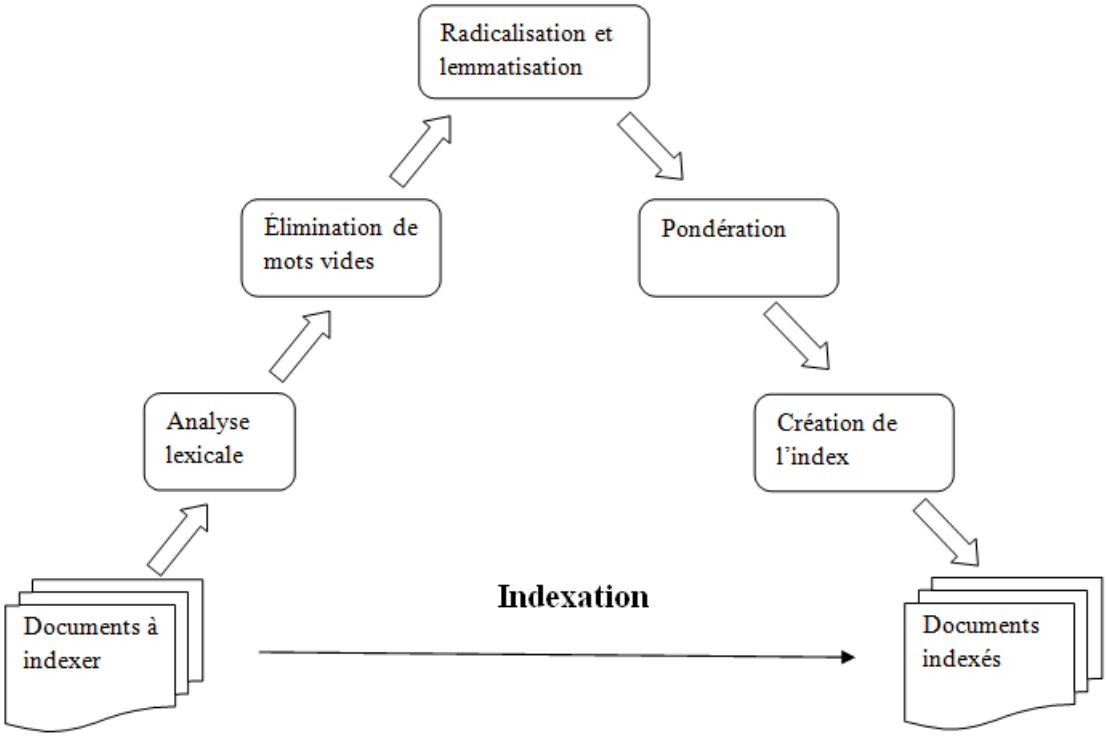

Figure 1.2 : Processus d'indexation.

1.2.2.2.1 Tokenisation

La tokenisation est le processus qui permet de transformer le contenu textuel d'un document en un ensemble de termes en reconnaissant les espaces de séparation des termes, les chiffres, les ponctuations, etc.

1.2.2.2.2 Elimination de termes vides

Le but de l'indexation automatique est de pouvoir extraire les termes les plus significatifs d'un texte. Or, dans chaque texte il existe des termes vides non porteurs de sens qui ne peuvent pas faire partie des termes les plus significatifs ([Fox, 1990], [Fox et al, 1992], [Rijsbergen, 1979]). Les termes non porteurs de sens comprennent : les articles, les pronoms, les prépositions et les mots outils. Les termes athématiques font aussi partie des termes vides. Les termes athématiques sont les termes qui exposent le sujet du texte mais ne le traitent pas comme par exemple : contenir, appartenir, etc. L'élimination des mots vides est relativement un traitement simple ; il s'agit de vérifier tous les mots d'un texte mot par mot et éliminer les mots vides qui s'y trouvent. Pour cela, on utilise une liste où on recense tous les termes vides. Cette liste est habituellement appelée : anti-dictionnaire ou stop-list.

L'élimination de termes vides doit être contrôlée car elle influence la qualité des descripteurs choisis à la fin de l'indexation. Il est évident que le rôle d'un terme peut changer selon le contexte où il est utilisé ; pour cela la stop-list utilisée doit être adaptée à la collection de documents à indexer.

D'autre part, l'élimination des termes vides est importante car elle permet d'augmenter la précision en diminuant le bruit résultant de la présence de termes vides, mais l'élimination de ces derniers peut dans certain cas réduire le taux de rappel c'est-à-dire la proportion des documents pertinents renvoyés.

1.2.2.2.3 Radicalisation et Lemmatisation

Un terme donné peut avoir plusieurs formes dans un texte ; en revanche, le sens véhiculé par le terme reste généralement le même. Par exemple, nous pouvons constater que les termes : émettre, émis, émission, émettra, émetteur partagent tous le même contenu sémantique que véhicule le terme émettre même si ces termes se présentent sous une forme morphologique différente de celle de terme émettre. Si un utilisateur soumis une requête avec le terme émettre, alors vraisemblablement il serait intéressé par les documents qui ne comportent pas le terme émettre mais comportent d'autres termes de sa famille comme ceux cités précédemment. De là, nous constatons qu'il serait intéressant de ramener les termes qui font partie de la même famille à une certaine forme commune. Pour cela, il existe deux alternatives essentielles : La radicalisation et la lemmatisation.

- Radicalisation **:** La radicalisation ou le Stemming consiste à regrouper les variantes d'un même radical, par exemple, la radicalisation transforment les termes « collaborateur, *collaboration, collaborerai, collabora* » en le même radical « *collabor* ». Le radical est généralement un terme non réel qui ne fait pas partie du vocabulaire de la langue. Les algorithmes de radicalisation se différencient les uns des autres par les méthodes utilisées. Il y a ceux qui se basent sur des règles linguistiques tels que celui de Porter [Porter, 1980] ; d'autres se basent sur des méthodes statistiques, nous citons l'algorithme des n-grammes [Adamson, 1974]. Ils peuvent également se baser sur des règles lexiques [Savoy, 1993].

Parmi les algorithmes les plus répandus, nous trouvons l'algorithme de Porter. C'est un algorithme linguistique qui fait partie des algorithmes de suppression des affixes [Frakes, 1992]. L'algorithme de Porter se base sur l'idée de ramener un terme à son radical en supprimant sa terminaison. Pour ce faire, cet algorithme applique successivement plusieurs règles de

transformation visant à supprimer le pluriel, les formes conjugués et les différentes dérivations utilisées afin d'obtenir des adjectifs, des adverbes, etc.

Dans certains cas, la radicalisation d'un terme peut changer la sémantique originale du terme. Par exemple, le verbe « porter » et le nom « porte » ont pour radical « port » alors qu'ils sont sémantiquement très différents. Néanmoins, selon les études de comparaison réalisées, l'utilisation de la radicalisation est bénéfique pour la Recherche d'Information ; elle peut apporter davantage de pertinence au résultat retourné [Frakes, 1992].

- Lemmatisation : Une autre façon de regrouper les termes qui font partie de la même famille est la lemmatisation. Il s'agit de transformer les flexions des termes liées aux formes conjugués et accordées d'un terme à leur lemme. Par exemple, le terme« émettras » aura « émettre » comme lemme une fois ramené à une forme non conjuguée, également, le terme« réalisées » aura pour lemme le terme « réalisé » après suppression de la flexion liée à l'accord en genre (féminin) et en nombre (pluriel). Comme nous le remarquons dans les deux exemples précédents, le lemme, et contrairement au radical, est un terme réel qui appartient au vocabulaire de la langue. De ce point de vue, la lemmatisation est plus cohérente linguistiquement. Contrairement à la radicalisation, la lemmatisation ne peut pas être réalisée à l'aide de troncatures automatiques ou à l'aide d'un ensemble de règles ; elle nécessite l'utilisation de dictionnaires et un traitement singulier des termes un par un.

Par ailleurs, nous avons deux types de lemmatisation : lemmatisation hors contexte et lemmatisation en contexte. La lemmatisation hors contexte est une lemmatisation qui n'a pas besoin du contexte (la phrase) dans lequel le terme apparait ; elle se base plutôt sur la forme fléchie qu'épouse le terme. A partir de cette forme fléchie, le lemmatiseur retourne directement l'ensemble des lemmes possibles pour cette flexion dans chaque contexte grammaticale où elle peut être résolue. En d'autres termes, ce type de lemmatiseur ne renvoie pas un lemme unique mais une collection de lemmes possibles en fonction de la catégorie grammaticale du mot et c'est à l'utilisateur de choisir le lemme le plus approprié. Afin de simplifier la tâche à l'utilisateur, la lemmatisation en contexte consiste à identifier la fonction grammaticale d'un terme fléchi pour pouvoir en déduire automatiquement le lemme approprié parmi la liste des lemmes possibles et ne présenter à l'utilisateur qu'un seul lemme associée à la catégorie grammaticale.

1.2.3 Pondération

Le but de l'indexation est d'associer à un document un ensemble de descripteurs qui représentent le mieux son contenu. Afin de choisir ces descripteurs, nous avons besoin d'une stratégie qui nous permet de quantifier la représentativité de chacun des termes d'un document et de choisir les meilleurs. La pondération est la stratégie qui nous permet de réaliser cette quantification.

Différentes approches de pondération ont été proposées dans la littérature [Salton, 1987]. La majorité des approches de pondération se basent sur la notion de fréquence. En effet, nous considérons que plus un terme est fréquent dans un document plus il est important. Cette considération est justifiée par la loi de distribution des termes de Zipf [Zipf, 1949]. Cette loi stipule que si on dresse une liste de l'ensemble des termes d'un texte par ordre décroissant du nombre d'apparition, la fréquence d'un terme serait inversement proportionnelle à son rang dans la liste, ou encore, que le produit de la fréquence de n'importe quel terme par son rang dans la liste dressée est constant. Cette loi est écrite sous la forme :

$$fréquence * rang = constante \qquad (1.1)$$

Zipf explique cela par le faite qu'il est plus facile pour un auteur d'un document de répéter certains termes qu'il a utilisé auparavant au sein du même texte que d'utiliser de nouveaux termes. Ainsi, la fréquence d'un terme dans un document peut être un bon indicateur de son importance au sein de ce document, et un moyen efficace pour quantifier cette importance sur lequel repose la majorité des formules de pondération existantes[Robertson, 2004].

D'autre part, la plupart des formules de pondération des termes sont construites par combinaison de deux facteurs. Un facteur de pondération locale (*tf : Term Frequency*), quantifiant la représentativité locale d'un terme dans le document, et un second facteur de pondération globale (*idf : Inverse of Document Frequency)* mesurant la représentativité globale du terme vis-à-vis de la collection des documents. Ces deux facteurs, de pondération locale et globale, doivent être pris en compte ensemble pour une bonne évaluation de la représentativité d'un terme dans un document appartenant à une collection donnée. En effet, la fréquence locale nous indique la représentativité du terme seulement par rapport au document, en même temps, la fréquence globale nous informe sur la représentativité d'un terme pour toute la collection des documents. Ainsi, et grâce à la fréquence globale *idf,* un terme qui est très présent dans tous les

documents ne doit pas être considéré comme terme représentatif pour un document même s'il apparaisse fréquemment dans ce document.

***tf* (Term Frequency):** Ce facteur indique l'importance du terme dans le document où il se trouve. Cette importance est proportionnelle à la fréquence (ou à la présence) du terme au sein du document. Elle peut être calculée de plusieurs façons [Salton, 1987] :

- fonction brute de tf_{ij} (term frequency) : correspond au nombre d'apparitions du terme t_i dans le document d_j ;
- fonction binaire : elle vaut 1 si le terme est présent dans le document, 0 sinon.
- fonction logarithmique : combine tf_{ij} avec un logarithme, donné par : α+log (tf_{ij}), où α est une constante. Cette fonction permet d'atténuer les effets de larges différences entre les fréquences d'apparitions des termes dans le document.
- fonction normalisée : permet de réduire les différences entre les valeurs associées aux termes du document en donnant des valeurs entre 0.5 et 1. Cette fonction est donnée comme suit :

$$tf_{ij} = 0.5 + 0.5 * \frac{tf_{ij}}{max_{t_i \in D_j} tf_{ij}} \qquad (1.2)$$

Où $max_{t_i \in D_j} tf_{ij}$ est la plus grande valeur tf_{ij} des termes du document D_j.

***Idf* (Inverse of Document Frequency):** mesure l'importance d'un terme dans toute la collection. Un terme trop fréquent dans la collection ne doit pas avoir le même impact sur la collection qu'un terme moins fréquent. Ce facteur est mesuré à l'aide de différentes formules, parmi lesquelles, nous pouvons citer :

$$Idf_i = \log\left(\frac{N}{n_i}\right) \qquad (1.3)$$

$$Idf_i = \log\left(\frac{N - n_i}{N}\right) \qquad (1.4)$$

$$Idf_i = \log(1 + \frac{N}{n_i}) \qquad (1.5)$$

Où :

N : est le nombre des documents de la collection.

n_i : est le nombre de documents où le terme t_i apparait.

Avec la pondération globale *Idf*, on s'intéresse plutôt à la distribution d'un terme dans le corpus, en négligeant la fréquence d'apparition d'un terme dans un document. Ainsi, on peut éliminer les termes fonctionnels (tels que : de, et, pour, etc.) qui sont présents dans la majorité des documents de la collection sans qu'ils soient représentatifs du contenu de ces documents.

Comme nous l'avons mentionné précédemment, la pondération des termes se base sur la combinaison des deux facteurs de pondération locale et globale (*tf* et *idf)* qui nous donne la formule de pondération *TF-IDF*. Avec cette formule, plus un terme est présent dans un document et absent dans le reste des documents de la collection, plus il est considéré comme représentatif et il lui est attribué un poids important.

La mesure de pondération *TF-IDF* donne une bonne approximation de l'importance du terme dans le document, particulièrement dans les corpus de documents ayant des tailles similaires. Toutefois, elle ne tient pas compte d'un aspect assez important du document qui est sa longueur. En effet, dans un document long, on utilise plus de termes pour décrire le sujet et on les répète encore plus. Ainsi, ces termes seront plus fréquents et auront des poids encore plus importants ce qui favorise les documents longs lors de l'appariement document/requête. Afin de pallier à ce problème, de nouvelles formules pour la mesure *TF-IDF* qui intègrent la taille du document ont été proposées [Salton et Buckley, 1987]. Parmi ces mesures, nous citons la formule *TF-IDF* normalisée de Robertson et Sparck-Jones [Robertson et Sparck-Jones, 1997] qui est définie comme suit :

$$tf - idf_{ij} = \frac{tf_{ij} * (k_1 + 1) * \log(\frac{N}{n_i})}{k_1 * \left((1-b) + b * \frac{dl_j}{\Delta l}\right) + tf_{ij}} \quad (1.6)$$

Où :

$tf - idf_{ij}$: est le poids du terme t_i dans le document D_j ;

k_1 : est un paramètre qui contrôle l'influence de la fréquence du terme t_i, sa valeur optimale dépend de la longueur et de l'hétérogénéité des documents dans la collection de documents (dans la collection de test TREC k_1= 2) ;

b : appartient à l'intervalle [0, 1], il contrôle l'effet de la longueur du document ;

dl_j : est la longueur du document D_j , elle est mesurée soit par rapport au nombre de termes, soit par rapport à la taille en octet ;

Δl : est la taille moyenne d'un document par rapport à la collection de documents, elle est aussi mesurée soit en nombre de termes, soit en octet.

1.2.4 Structure des fichiers index

Afin de répondre plus rapidement à une requête, des structures de stockage particulières sont nécessaires pour stocker les descripteurs des documents sélectionnés lors du processus d'indexation. Différentes structures ont vu le jour [Rijsbergen, 1979], nous citons :

- Fichier séquentiel : Le fichier séquentiel est la structure utilisée pour le stockage de données dans les bandes magnétiques. Ces fichiers insèrent les enregistrements l'un après l'autre d'une façon séquentielle. La recherche d'un enregistrement consiste alors à parcourir tous les enregistrements du fichier jusqu'à trouver l'enregistrement recherché. Ainsi, ces fichiers offrent une grande facilité pour l'insertion de nouveaux enregistrements et une optimisation dans l'espace de stockage mais ils sont couteux en temps de recherche.

- Fichier de signature : Le fichier de signature est un fichier ou le document textuel est remplacé par sa signature. La signature d'un document est une chaine binaire de *n* bits, elle résulte d'une opération de hachage sur l'ensemble des termes que contient le document. Une fois toutes les signatures d'une collection de documents sont élaborées, elles sont stockées dans le fichier de signatures. Le fichier de signature a une taille beaucoup plus faible que la collection de documents originale, la recherche dans ce fichier est beaucoup plus rapide que dans la collection. Avec ce fichier de signature, la réponse à une requête consiste à calculer pour la requête sa signature et la comparer aux signatures de la collection.

Avec ce type de fichier d'index, on dispose de peu d'informations sur les termes. En effet, on ne peut connaitre que la présence ou non d'un terme, sa fréquence d'apparition et son emplacement sont inconnus. Néanmoins, les résultats obtenus par ce type de fichier sont relativement bons, tant du point de vue mise à jour, charge de la machine, stockage de données ou réponses aux requêtes.

- Fichier inverse : Il est le type de fichier utilisé par la majorité des Systèmes de Recherche d'information. Son principe est le suivant : Pour chaque document de la collection, on choisit une liste de mots clés qui le décrivent. L'ensemble des mots clés, qui décrivent la collection des documents, sont stockés généralement par ordre alphabétiques. Pour chaque mot clé, un pointeur va vers chacun des documents que ce mot décrit.

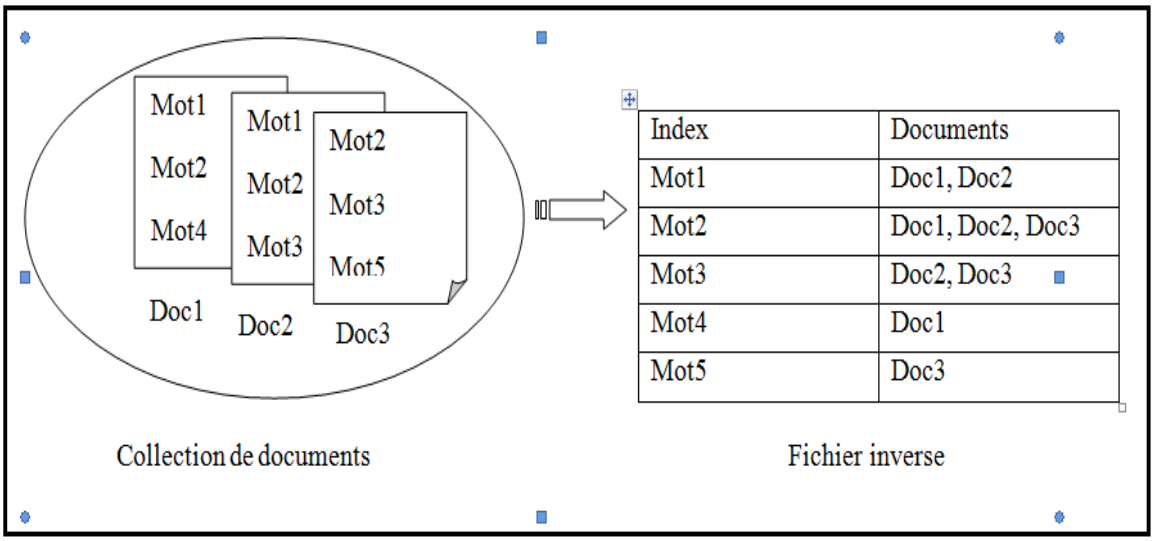

Figure 1.3 : Structure du fichier inverse.

Outre les documents où le terme apparait, un fichier inverse contient habituellement la fréquence locale *tf* et globale *idf* du terme. Pour chaque terme t_i, nous avons une fréquence globale idf_i et une série de couples (d_j, tf_{ij}) pour les fréquences locales tf_{ij} du terme selon les documents d_j.

Du point de vue temps de réponse, les fichiers inverses sont plus performants que les fichiers de signature. En effet, l'utilisation de la structure de fichier inverse rend la recherche d'un résultat plus rapide ce qui fait d'eux la structure de fichier d'index la plus utilisée par les Systèmes de Recherche d'Information actuels. Néanmoins, ce type de structure consomme énormément de place de stockage, les fichiers index étant parfois aussi gros que les fichiers de données, surtout dans le cas où les positions où apparaissent les mots clés dans les fichiers sont stockées. Les mises à jour sont aussi coûteuses puisqu'il faut refaire l'index à chaque nouvelle insertion.

1.3 Modèles de Recherche d'Information

Le modèle de Recherche d'Information a pour rôle de modéliser le processus de Recherche d'Information sur lequel se base le SRI. Ce modèle a deux fonctions essentielles :

- La création d'une représentation interne du document et de la requête en utilisant les termes d'indexation ;
- La définition d'une méthode pour la mesure de similarité d'un document par rapport à une requête ;

On peut classer les modèles de recherche d'information en trois catégories :

- les modèles ensemblistes ;
- les modèles algébriques ;
- les modèles probabilistes.

1.3.1 Modèles ensemblistes

Les modèles ensemblistes se basent sur la théorie des ensembles et l'algèbre de Boole ; ainsi une requête est représentée par un ensemble de termes séparés par des opérateurs logiques : conjonction (ET), disjonction(OU) et négation (NON). Le document est, quant à lui, représenté par une liste de mots-clés. Ces modèles permettent d'effectuer des opérations d'union, d'intersection et de différence lors de l'interrogation. Le modèle le plus connu et le plus utilisé de cette catégorie est le modèle booléen.

Modèle booléen : Le modèle booléen est un modèle ensembliste qui se base sur l'utilisation des opérateurs logiques. Dans ce modèle, la requête est exprimée sous forme d'une expression logique composée de mots clés et d'opérateurs booléens. Le mot clé est un terme qui définit un besoin en information de l'utilisateur. L'opérateur booléen est un opérateur logique qui permet de lier les termes de la requête, nous citons :

ET (∧) : sert à indiquer la présence simultanée de plusieurs mots clés ;

OU (∨) : indique que l'on cherche des documents ayant pour mot clé l'un des mots donnés ;

NON (¬) : sert à éliminer des documents incluant un mot clé.

Le processus de recherche mis en œuvre via ce type de modèle, consiste à effectuer des opérations logiques sur les ensembles de documents qui sont définis par la présence ou l'absence de termes d'indexation. Afin de trouver les documents réponses, un appariement exact entre le document et l'équation de la requête est réalisé. Finalement, il ya que les documents répondant exactement aux termes de la requête qui sont restitués. Prenons par exemple la requête q= t1 ^ t2 ¬ t3. Dans ce cas, le SRI doit retourner en réponse tous les documents qui ont t1 et t2 comme descripteurs et n'ont pas t3 comme descripteur.

Le modèle booléen est considéré comme le modèle le plus simple à comprendre et à mettre en œuvre, néanmoins il présente quelques inconvénients : d'abord la similarité entre un document et une requête est de 1 ou de 0, ainsi nous ne pouvons pas classer les documents retournés par ordre de pertinence, ils sont considérés tous comme égaux. D'autre part, la formulation d'un besoin en une expression logique n'est pas une tâche accessible à tous les utilisateurs. Pour certains besoins, la requête peut devenir très compliquée.

1.3.2 Modèles algébriques

Les modèles algébriques sont des modèles qui se basent sur la statistique. Ils utilisent une représentation vectorielle des documents et des requêtes [Piwowarski, 2003]. Dans cette représentation, la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance algébrique dans un espace multidimensionnel dont les dimensions sont les termes issus de l'indexation [Salton et al, 1983]. Le représentant le plus connu et le plus utilisé de cette catégorie est le modèle vectoriel. Par la suite, plusieurs modèles s'inspirant du modèle vectoriel ont été proposés dans le domaine de la RI : le modèle vectoriel généralisé [Wong et al, 1985], le modèle connexionniste [Boughanem, 1992] [Mothe, 1994] [Kwok, 1989] et le modèle LSI (Latent Semantic Indexing) [Deerwester et al, 1990].

Modèle vectoriel : Luhn [Luhn, 1957] a été le premier à suggérer une représentation vectorielle des requêtes et des documents. Par la suite, Gérard Salton et son équipe [Salton, 1971] [Salton et al., 1983] ont mis en place cette représentation à travers leur projet SMART (Salton's Magical Automatic Retriever of Text).

Dans ce modèle, un document ainsi qu'une requête sont représentés comme un vecteur de poids. Formellement, un document d'une collection *DOC* est représenté dans un espace vectoriel de dimension m par un vecteur $\vec{d}$= (d_1,d_2 , d_3, … , d_m) dont chaque composante d_k(avec k ϵ(1,2,3,…,m)) est liée à un des termes d'indexation. La requête est également représentée par un vecteur $\vec{q}$= (q_1,q_2,q_3,… , q_m) dont chaque composante q_k(avec k ϵ (1,2,3,…, m)) est liée à un des termes d'indexation.

Avec ce modèle, le calcul de la pertinence d'un document par rapport à une requête revient au calcul de la similarité entre les vecteurs liés à la requête et au document. De nombreuses mesures de similarité pour le modèle vectoriel ont été proposées, la plus connue d'entre elles est le produit scalaire. Cette mesure correspond au produit scalaire classique entre le vecteur du document $\overrightarrow{d_i}$ et le vecteur de la requête $\overrightarrow{q}$, elle est définie comme suit :

$$Sim(\vec{d}\,,\vec{q}) = \sum_{k=1}^{m} d_k . q_k \quad (1.7)$$

Avec cette mesure, les documents longs sont avantagés par rapport aux documents courts du fait que le produit scalaire ne tient pas compte de la longueur du document. En effet, les documents longs ont plus de descripteurs et ont plus de chance d'être considérés comme pertinents. D'autres mesures normalisées, qui prennent en considération la longueur du document, ont été proposées. Il s'agit de :

Cosinus: $Sim(\overrightarrow{d_i}, \vec{q}) = \frac{\sum_{i=1}^{n}(d_i * q_i)}{\sqrt{\sum_{i=1}^{n} d_i^2 * \sum_{i=1}^{n} q_i^2}}$ (1.8)

Jacard: $(\overrightarrow{d_i}, \vec{q}) = \frac{\sum_{i=1}^{n}(d_i * q_i)}{\sum_{i=1}^{n} d_i^2 + \sum_{i=1}^{n} q_i^2 - \sum_{i=1}^{n} d_i\, q_i}$ (1.9)

Dice: $Sim(\overrightarrow{d_i}, \vec{q}) = 2 * \frac{\sum_{i=1}^{n}(d_i * q_i)}{\sum_{i=1}^{n}(d_i^2 + q_i^2)}$ (1.10)

Grâce à la pondération des termes et du calcul de similarité d'un document par rapport à une requête, le modèle vectoriel permet de classer les documents selon leurs pertinences et de les présenter à l'utilisateur sous forme d'une liste ordonnée de documents du plus pertinent au moins pertinent. Néanmoins, le modèle vectoriel présente un inconvénient majeur qui est l'indépendance mutuelle des termes d'indexation [Wong et al, 1985]. En effet, le modèle vectoriel ne permet pas de représenter des phrases ou des mots multi-termes [Yates et al, 1999].

1.3.3 Modèles probabilistes

Le modèle probabiliste résout le problème de la Recherche d'Information en utilisant un modèle mathématique basé sur la théorie de probabilité [Robertson et Sparck Jones, 1976]. L'idée de faire appel à la théorie de probabilité a été d'abord proposée par Maron et Kuhns [Maron et Kuhns, 1960] au début des années 1960. Par la suite, Robertson a introduit le principe de classement probabiliste PRP (Probability Ranking Principle). Ce principe consiste à classer les documents selon la probabilité de pertinence vis-à-vis de la requête. En effet, la similarité entre un document et une requête est mesurée par le rapport entre la probabilité qu'un document *d* soit pertinent pour une requête *q*, notée *p*(*d*/*q*), et la probabilité qu'il soit non pertinent notée $p(\bar{d}/q)$. A leur tour, ces deux probabilités *p*(*d*/*q*) et $p(\bar{d}/q)$ sont calculées sur la base de deux autres probabilités qui se basent sur la présence ou l'absence d'un terme dans un document. Ces probabilités sont : $p(t/p)\ et\ p(t/np)$.

$p(t,p)$ est la probabilité qu'un terme *t* de la requête apparaisse dans un document sachant que ce document est pertinent et $p(t,np)$ est la probabilité qu'un terme *t* de la requête apparaisse dans un document sachant que ce document est non pertinent. Ainsi, la similarité d'un document vis à vis d'une requête, en se basant sur la présence d'un terme dans un document, est définie comme suit :

$$Sim(q,d) = \frac{p(d/q)}{p(\bar{d},q)} = \sum_{i=1}^{t} log \frac{p(t_i/p)(1-p(t_i,np))}{p(t_i,np)(1-p(t_i/p))} \quad (1.11)$$

D'autres auteurs intègrent la fréquence d'apparition des termes dans le document. Ainsi, Croft et Harper [Croft et Harper, 1979] proposent de calculer la similarité d'un document vis-à-vis d'une requête en prenant en compte la probabilité d'apparition d'un terme t_i dans un document d notée $P(d_i)$. $P(d_i)$ peut simplement correspondre à la fréquence d'apparition du terme t_i dans le document d ; elle est calculée selon la formule :

$$P(d_i) = \frac{\mathrm{tf_i}}{\mathrm{Max\ tf}} \qquad (1.12)$$

Ainsi, la similarité entre un document et une requête est calculée comme suit :

$$Sim(d,q) = C \sum_{i=1}^{n} d_i\, q_i + \sum_{i=1}^{n} p(d_i)\, d_i q_i \log(\frac{N-n_i}{n_i}) \qquad (1.13)$$

Où :

C : constante ;

N : nombre de documents de la collection ;

n_i : nombre de documents contenant le terme *ti.*

De leur coté, Robertson et Walker [Robertson et Walker, 1994] ont utilisé la loi de poisson pour calculer la probabilité de pertinence en intégrant la fréquence du terme et la longueur du document. Les auteurs ont aussi implémenté le modèle probabiliste dans le système Okapi.

D'une manière générale, et selon Jacques Savoy [Savoy, 1993], si nous comparons les trois modèles de recherche d'information que nous venons de voir dans cette section, nous trouvons que le modèle de recherche probabiliste est plus efficace que le modèle de recherche booléen, mais moins performant que le modèle de recherche vectoriel.

1.4 Evaluation des systèmes de Recherche d'information

1.4.1 Notions de base

Un SRI a pour fonction de répondre à un besoin d'information d'un utilisateur par un ensemble de documents pertinents. L'évaluation d'un SRI peut concerner de nombreux aspects : la pertinence des résultats retournés, le mode de présentation des résultats à l'utilisateur, la satisfaction de l'utilisateur des résultats retournés, les performances du SRI en matière de temps de calcul et d'espace de stockage, …etc. Ainsi, nous pouvons distinguer deux volets essentiels pour l'évaluation d'un SRI : l'efficience et l'efficacité. L'efficience regroupe les critères de performance du système qui sont le temps de calcul et l'espace de stockage. Plus un système est rapide et peu exigeant en matière d'espace de stockage plus il est considéré comme meilleur. Par

ailleurs, l'efficacité concerne un aspect plus compliqué du SRI qui est la pertinence des résultats retournés et leur degré de satisfecit par rapport à la requête soumise.

Dans cette section d'évaluation des SRI, nous allons nous focaliser sur la capacité d'un SRI à sélectionner les documents pertinents pour une requête à partir d'une base documentaire. Par ailleurs, nous pouvons différencier deux types de pertinence : une pertinence objective et une pertinence subjective. La pertinence objective (pertinence système) se base sur le score de pertinence calculé par le SRI et attribué à un document par rapport à une requête donnée. La pertinence subjective (pertinence utilisateur) quant à elle, se base sur le jugement que porte l'utilisateur sur un document selon son besoin en information.

L'évaluation d'un SRI se base sur un ensemble de mesures qui permettent de juger le degré de pertinence des résultats retournés, ces mesures permettent également de comparer les SRI entre eux. L'élaboration de ces mesures a établie un nombre d'hypothèses relatives à l'évaluation des SRI dont les plus importantes sont :

- *Présentation :* Les documents sont ordonnés par score décroisant ;
- *Ordre du parcours :* L'utilisateur parcourt la liste des documents retournés en respectant l'ordre dans lequel ils sont présentés. Il ne procède jamais d'une façon aléatoire ;
- *Jugement absolu* : Un document reste pertinent même s'il contient exactement la même information qu'un autre document déjà présenté à l'utilisateur ;
- *Non additivité :* Deux documents non pertinents ne pourront jamais former une unité d'information pertinente (même s'ils se complètent).

1.4.2 Mesures d'évaluation

1.4.2.1 Mesures de rappel et de précision

Le principe du processus de Recherche d'Information d'un SRI consiste à choisir un ensemble de documents à restituer considérés comme pertinents pour la requête soumise. Selon ce principe, nous pouvons distinguer deux façons pour partitionner la collection de documents (Figure 1.4) :

La pertinence : nous partitionnons la collection de documents en deux ensembles, les documents pertinents et les documents non pertinents à une requête donnée.

La restitution des résultats : de même, la collection de documents est partitionnée en deux ensembles, l'ensemble des documents restitués et l'ensemble des documents non restitués à une requête donnée.

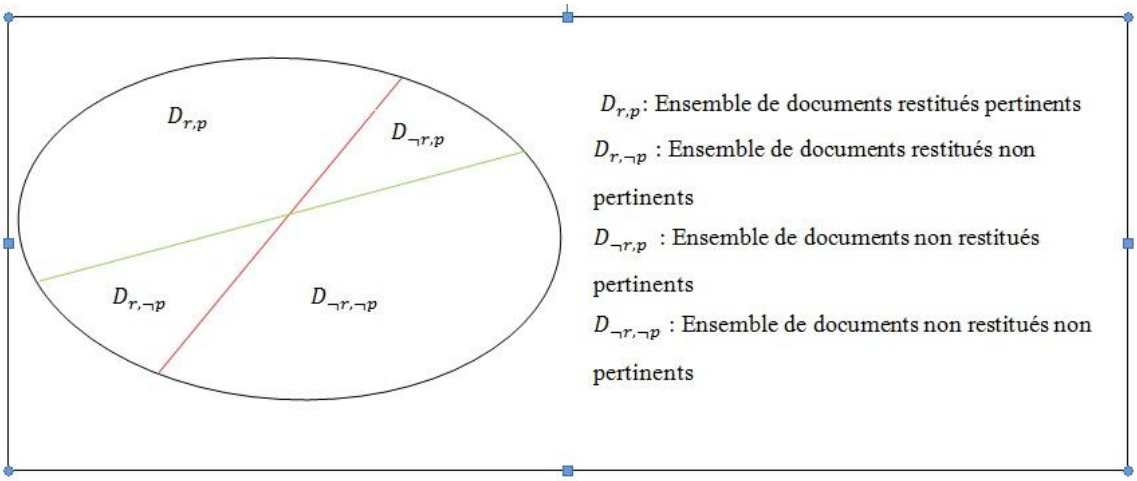

Figure 1.4 : Partitionnement de la collection de documents pour une requête

Afin de mesurer la qualité des documents restitués par le SRI, nous allons comparer les ensembles de documents résultats des deux partitionnements cités dans le paragraphe précédent. Cette comparaison est réalisée à travers deux mesures de base qui sont : le rappel et la précision. Ces mesures ont été introduites par Van Rijsbergen [Rijsbergen, 1979].

Rappel : mesure la proportion des documents pertinents restitués comme résultat à la requête par rapport aux documents pertinents existants dans la collection de documents pour la même requête. Si le rappel est égal à 1 cela veut dire que tous les documents pertinents qui existent dans la collection des documents ont été restitués par le SRI. Cette mesure permet aussi de déterminer le silence qui est la proportion de documents pertinents non restitués.

$$Rappel = \frac{D_{r,p}}{D_{r,p} + D_{\neg r,p}} \quad (1.14)$$

Précision : mesure la proportion des documents pertinents restitués par rapport à l'ensemble des documents restitués. La précision vaut 1 quand tous les documents restitués sont pertinents et elle est nulle quand aucun document pertinent n'est restitué. Cette mesure permet également de déterminer la proportion des documents non pertinents restitués par le SRI appelé bruit.

$$Précision = \frac{D_{r,p}}{D_{r,p} + D_{r,\neg p}} \quad (1.15)$$

D’une façon générale, l’évaluation de la qualité des résultats retournés se base sur la combinaison des deux mesures : rappel et précision. Un SRI qui aurait un rappel et une précision qui valent 100% simultanément est un SRI qui a restitué tous les documents pertinents et aucun document non pertinent. En réalité, cette situation n'arrive pas. Le plus souvent, il est possible d’obtenir un taux de précision et de rappel aux alentours de 30 %.

Par ailleurs, la mesure de la précision indépendamment du rappel et inversement est peu significative car les deux mesures rappel et précision ne sont pas indépendantes l’une de l’autre ; quand l’une augmente l’autre diminue. En effet, si nous voulons avoir un rappel de 100%, il suffit de restituer tous les documents de la collection mais dans ce cas nous aurons aussi une faible précision et un important bruit (documents non pertinents). En revanche, si nous voulons avoir une importante précision, il faut qu’il y ait peu de documents restitués ce qui affaiblit le rappel. Ainsi, avoir un bon SRI revient à maximiser la précision sans trop sacrifier le rappel, nous pouvons gérer ce compromis entre rappel et précision en utilisant la courbe rappel-précision.

Courbe rappel-précision : L’évaluation d’un SRI sur la base des deux mesures : rappel et précision passe par la construction de la courbe rappel-précision. Cette courbe concerne une collection de documents et une requête donnée. Pour construire la courbe, il faut calculer pour chaque document restitué la paire des valeurs (précision, rappel). Un exemple d’une collection de dix documents triés par ordre de pertinence système est présenté dans le tableau 1.1. Cette collection contient dix documents dont cinq documents pertinents pour la requête soumise.

Ordre de restitution	Pertinent	Rappel	Précision
1	✓	0.20	1
2	✓	0.40	1
3		0.40	0.66
4	✓	0.60	0.75
5		0.60	0.60
6	✓	0.80	0.67
7		0.80	0.57
8		0.80	0.50
9		0.80	0.44
10	✓	1	0.50

Tableau 1.1: Exemple du calcul de rappel et de précision pour une requête.

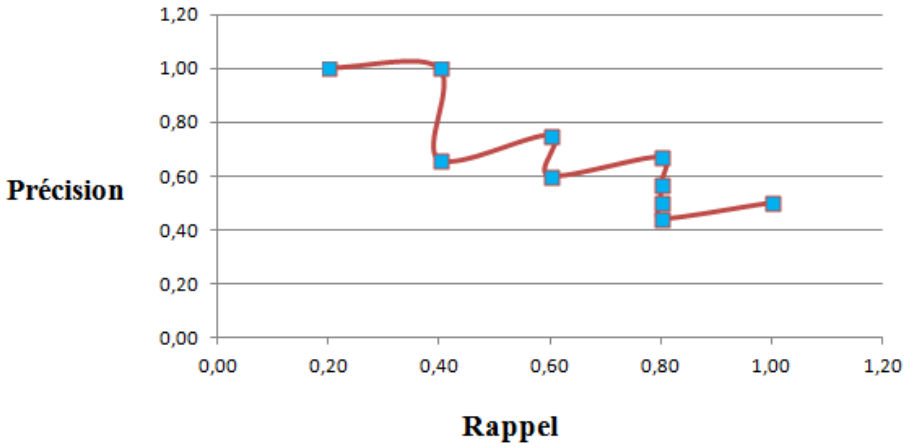

Figure 1.5 : Courbe de rappel et de précision pour l'exemple du tableau 1.1.

En observant la courbe rappel-précision (Figure1.5), nous pouvons constater que plusieurs valeurs de précision peuvent correspondre au même point de rappel. Afin d'obtenir des courbes plus aisées à lire, on ne représente généralement que la précision calculée à chaque point de rappel (Figure 1.6). Un point de rappel est le rappel calculé à la restitution d'un document pertinent. Par exemple, dans l'exemple précédent, nous avons cinq points de rappel qui sont les points de restitution des documents 1, 2, 4, 6, 10.

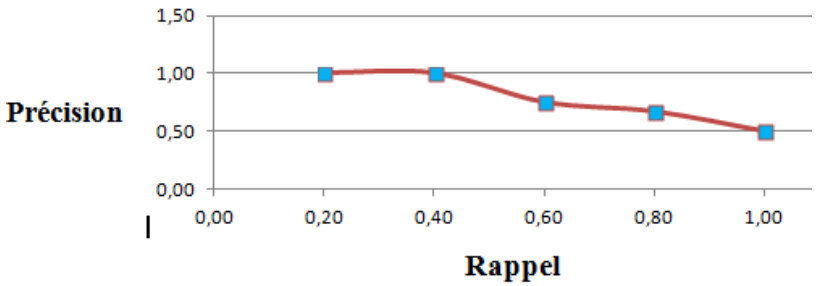

Figure 1.6 : Courbe simplifiée de rappel et de précision pour une requête.

.

Le calcul de la précision par rapport à l'ensemble des documents restitués, en ignorant l'ordre dans lequel les documents sont restitués, peut être insuffisant. En effet, quand le nombre de documents restitués est important, nous ne pouvons pas s'attendre à ce que l'utilisateur les consulte tous. L'utilisateur peut facilement se décourager des premiers documents s'il trouve qu'ils ne sont pas pertinents. Ainsi, même s'il existe des documents pertinents au bas de la liste de restitution, ils ne seront pas consultés par l'utilisateur. De là, nous constatons bien que l'ordre avec lequel les documents sont présentés est important et influe l'évaluation de l'utilisateur de la réponse retournée.

Afin d'évaluer cet ordre, nous avons la précision P_n et le rappel R_n qui sont respectivement la précision et le rappel aux *n* premiers documents restitués. Ainsi, si nous avons 30 documents réponses restitués à une requête donnée, nous calculons P_n , R_n avec *n=5, 10, 15, 20, 25, 30.* Cela aide à évaluer la capacité du SRI à restituer des documents pertinents en tête de liste. Un cas particulier de la précision à *n* documents est la R-précision ou la précision exacte, elle représente la précision calculée à *n* documents restitués en ayant une requête qui admet *n* documents pertinents. C'est le niveau où le rappel vaut la précision. Dans l'exemple de la figure1.5, la R-Précision est la précision calculée à la restitution du $5^{ème}$ document, elle vaut 0.60.

Jusqu'à là, nous calculons les valeurs de précision et de rappel par rapport à une seule requête. Pour évaluer un SRI pour un ensemble de requêtes, il faut calculer ce qu'on appelle la *précision moyenne*. Une *précision moyenne* est une précision qui utilise les résultats restitués à plusieurs requêtes. Le calcul de la précision moyenne, pose le problème des valeurs de rappel. En effet, nous avons des niveaux de rappel qui diffèrent d'une requête à une autre. Pour calculer la précision moyenne, nous devons d'abord normaliser les niveaux de rappel entre requêtes, généralement, nous retenons 10 niveaux de rappel à pas de 0.1. La valeur de précision qui n'existe pas pour un niveau de rappel retenu est calculée par interpolation linéaire comme suit.

Pour deux points de rappel, *i* et *j*, *i* < *j*, si la précision au point *i* est inférieure à celle au point *j*, on dit que la précision interpolée à *i* est égale la précision à *j*.

1.4.2.2 Mesures alternatives

Le rappel et la précision sont deux mesures intéressantes pour l'évaluation d'un SRI mais leur efficacité est relative à leur utilisation ensemble. Pour cela, plusieurs mesures combinant le rappel et la précision sont apparues, nous citons les plus importantes :

Moyenne harmonique F : La précision et le rappel permettent d'évaluer la qualité des résultats restitués par le SRI mais leur considération l'un séparément de l'autre peut diminuer de leur efficacité. La mesure F combine les deux mesures *P* et *R* afin d'avoir une meilleure évaluation du SRI, elle est calculée comme suit :

$$F = \frac{2 * R * P}{R + P} \qquad (1.16)$$

La mesure F a des valeurs élevées seulement lorsque la précision et le rappel ont les deux des valeurs élevées. Ainsi, plus la valeur de la moyenne harmonique est élevée et meilleur sera le SRI.

D'autre part, La précision et le rappel à *n* documents restitués P_n et R_n permettent d'évaluer la qualité de l'ordre avec lequel les documents sont restitués, nous pouvons calculer une moyenne harmonique F_n qui combine les deux mesures P_n et R_n afin d'avoir une meilleure évaluation de l'ordonnancement des documents restitués. Elle est calculée comme suit :

$$F_n = \frac{2 * R_n * P_n}{R_n + P_n} \qquad (1.17)$$

E-Mesure :

Dans la moyenne harmonique, le rappel et la précision ont la même importance. La E-Mesure est une extension de la moyenne harmonique proposée par Rijsbergen [Rijsbergen, 1979]. Elle permet de donner plus d'influence à la précision ou au rappel à travers un paramètre b. La E-Mesure est calculée comme suit :

$$E - Mesure(b) = 1 - \frac{1+b^2}{\frac{b^2}{rappel} + \frac{1}{précision}} \qquad (1.18)$$

Cette mesure est paramétrée par *b* : plus *b* est grand, plus l'importance donnée au rappel est grande et inversement.

1.4.2.3 Mesures orientées utilisateurs

L'objectif d'un SRI est de satisfaire un besoin en information d'un utilisateur. Ainsi, le jugement réel de la pertinence des résultats dépend de la satisfaction de l'utilisateur de ces résultats. Pour cela, plusieurs mesures d'évaluation orientées utilisateurs ont été proposées [Korfhage, 1997] ; nous citons ici les plus importantes :

- Ratio de couverture : c'est la proportion des documents restitués par le système et connus comme pertinents par l'utilisateur par rapport à la totalité des documents pertinents connus par l'utilisateur.
- Ratio de nouveauté : c'est la proportion des documents restitués pertinents et non connus par l'utilisateur par rapport à la totalité des documents pertinents.
- Rappel relatif: le rapport entre le nombre de documents pertinents trouvés par le système et le nombre de documents pertinents que l'utilisateur espérait trouver.
- Effort de rappel: le rapport entre le nombre de documents pertinents que l'utilisateur espérait trouver et le nombre de documents restitués consultés dans l'espoir de trouver des documents pertinents.

1.4.3 Collections de tests

Les mesures d'évaluation que nous avons présentées dans les sections précédentes ont l'objectif commun d'évaluer un SRI. L'évaluation d'un SRI peut être réalisée autrement en comparant le SRI à d'autres SRI. Pour ce faire, nous avons besoin de collection de documents qu'on appelle collection de test. Ces collections de documents doivent représenter une référence fiable et sûre pour permettre la comparaison entre les différentes stratégies de recherche des SRI. En effet, une collection de test doit satisfaire deux critères importants : d'une part, elle doit tenir compte de la pertinence subjective de l'utilisateur, d'autre part, elle doit contenir une masse d'information importante et variée. Ainsi, une collection de test est composée de :

- Une collection de documents ;
- Un ensemble de requêtes ;
- La liste des documents pertinents pour chaque requête.

La liste des documents pertinents pour chaque requête est établie par des experts ayant les connaissances nécessaires sur le domaine de la collection.

L'évaluation d'un SRI consiste d'abord à lui soumettre la liste de requêtes de la collection de test. Pour chaque requête, nous comparons la liste de documents restitués par le SRI par rapport à la liste de documents pertinents que contient la collection de test pour évaluer la capacité de ce SRI à restituer les documents pertinents. Par la suite, nous comparons les résultats d'évaluation obtenus pour tous les SRI afin d'identifier le meilleur système de recherche.

Depuis l'avènement de la RI, plusieurs compagnes de test se basant sur les collections de test ont vu le jour. L'objectif principal de ces compagnes est de stimuler la recherche d'information afin qu'il y ait des outils RI et des stratégies de recherche plus fiable et plus efficaces. Parmi les plus importantes compagnes de test, nous pouvons citer : la collection Cranfield [Cleverdon, 1962], la collection CISI et la collection TREC que nous allons présenter dans la section suivante.

Les collections TREC : Text Retrieval Conference ou TREC est un projet international initié au début des années quatre-vingt-dix par le NIST5 dans le but de proposer des collections standards pour l'évaluation de SRI sur des collections de documents conséquentes. Il est aujourd'hui co-sponsorisé par le NIST, l'ITL6 et l'IARPA7 (ex-DARPA8). Actuellement, TREC est organisée annuellement. Pour chaque session de la compagne, une collection de documents et de requêtes est fournie aux participants. Ces derniers exploitent leurs propres SRI sur les données de la compagne et renvoient une liste ordonnée de documents réponse pour chaque requête. Finalement, pour chaque requête on compare la liste de documents restitués par rapport à la liste de documents pertinents retenue.

Outre, la tâche standard de recherche de documents pertinents pour une requête, TREC propose de nombreuses autres tâches à évaluer selon les tendances du moment. Nous pouvons citer : le filtrage d'information, la recherche ad hoc (soumettre des requêtes sur une collection statique), la recherche multilingue, la fusion des bases de données et de nombreuses autres tâches secondaires. Depuis son apparition, la compagne TREC a grandement contribué à l'évolution des outils et des tâches de la Recherche d'Information en fournissant des collections de test fiables et efficaces.

1.5 Conclusion

Ce premier chapitre a été consacré à la présentation des concepts de base de la Recherche d'Information. Plus particulièrement, nous nous sommes intéressés au processus d'indexation et aux modèles principaux de la Recherche d'Information ainsi qu'aux différentes mesures d'évaluation des Systèmes de Recherches d'Information existantes. La Recherche d'Information est un domaine en pleine expansion et qui vise essentiellement de répondre à un besoin d'utilisateur en information d'une façon rapide et efficace.

CHAPITRE 2
SURVEY SUR LES APPROCHES DE COUPLAGE OLAP-RI

1.1 Introduction

Les entrepôts de données ont été introduits au début des années 1990 [Codd et al, 1993], dans le but d'améliorer le processus de décision au sein de l'entreprise. En effet, les entrepôts de données facilitent la consultation et l'analyse des données agrégées et historisées généralement au sein des bases de données multidimensionnelles [Colliat et al, 1996]. De nos jours, la construction des bases multidimensionnelles à base de données numériques à des fins d'analyse est une tâche bien maitrisée par les spécialistes du domaine. En revanche, l'intégration des données textuelles au sein des entrepôts de données représente toujours un verrou scientifique.

Dans ce cadre, quelques travaux récents se sont intéressés à intégrer les données textuelles dans les entrepôts de données en se basant sur les différentes techniques de traitement de texte existantes. Par exemple, Keith et al [Keith et al, 2005] proposent l'utilisation des approches TALN (Traitement Automatique du Langage Naturel) pour l'agrégation des mots dans un environnement OLAP selon leur morphologie. D'autres travaux, utilisent des méthodes statistiques issues du domaine de la Recherche d'Information (RI) pour agréger les données textuelles telles que les travaux de Pujolle et al. (2008), Lin et al. (2008, 2011), Zhang et al.(2012) et Pérez-Martínez et al. (2008).

Dans ce chapitre, nous nous intéressons aux approches proposées pour l'intégration des données textuelles au sein des entrepôts de données en se basant sur des techniques de la Recherche d'Information. Dans cette optique, nous allons présenter dans ce chapitre une revue sur les travaux combinant la Recherche d'Information et la technologie OLAP. Ces travaux ont pour objectif principal l'intégration du contenu textuel au niveau des entrepôts de données et sa manipulation par des opérateurs OLAP. Pour ce faire, les auteurs de ces travaux essayent de coupler des techniques et des outils robustes de la Recherche d'Information, très utilisés pour l'interrogation de corpus de texte, avec des mécanismes OLAP.

2. Revue sur les approches de couplage OLAP-RI

Récemment, plusieurs approches essayant de combiner la Recherche d'Information et la technologie OLAP ont vu le jour. Ces approches visent de coupler la fiabilité et la robustesse de la Recherche d'Informations dans l'indexation, l'interrogation et la manipulation de grands corpus textuels et les possibilités d'analyse qu'offre la technologie OLAP telles que le groupement ou l'agrégation.

Selon Pérez et al, nous pouvons classer ces approches selon deux catégories [Pérez et al, 2006] : Les approches utilisant les bases multidimensionnelles pour l'implémentation des SRI et les approches utilisant la Recherche d'Information pour la manipulation de contenu textuel au sein des bases de données multidimensionnelles.

2.1 Systèmes de Recherche d'Informations multidimensionnels

L'implémentation des SRI en utilisant les bases multidimensionnelles vise à améliorer la fiabilité des SRI et la pertinence des réponses offertes aux utilisateurs. Le fait de stocker les documents dans des entrepôts de données en étant classifiés selon des thématiques (sujets) bien déterminés peut énormément faciliter la navigation au sein du corpus des documents et par conséquent améliorer la qualité des documents retournés en réponse à une requête donnée. Toutefois, les travaux qui utilisent les bases de données multidimensionnelles de texte pour l'implémentation des SRI, ne vont pas jusqu'à la réalisation effective des opérations OLAP sur les documents textuels. C'est plutôt les avantages de l'organisation multidimensionnelle des documents qui justifie leur recours à des bases multidimensionnelles pour l'implémentation des SRI.

Parmi ces travaux, nous citons :

2.1.1 Les travaux de McCabe et al

Dans ce travail [McCabe et al, 2000], les auteurs proposent une approche pour la manipulation du contenu textuel des documents en combinant des opérations d'analyse OLAP à des techniques de Recherche d'Information. Selon les auteurs, la donnée textuelle est souvent convertible en une donnée multidimensionnelle ce qui permet son intégration dans une base de données multidimensionnelle. Pour illustrer leur approche, les auteurs ont travaillé sur une collection de documents d'archives gouvernementales en adoptant une modélisation multidimensionnelle en étoile (Figure 2.1).

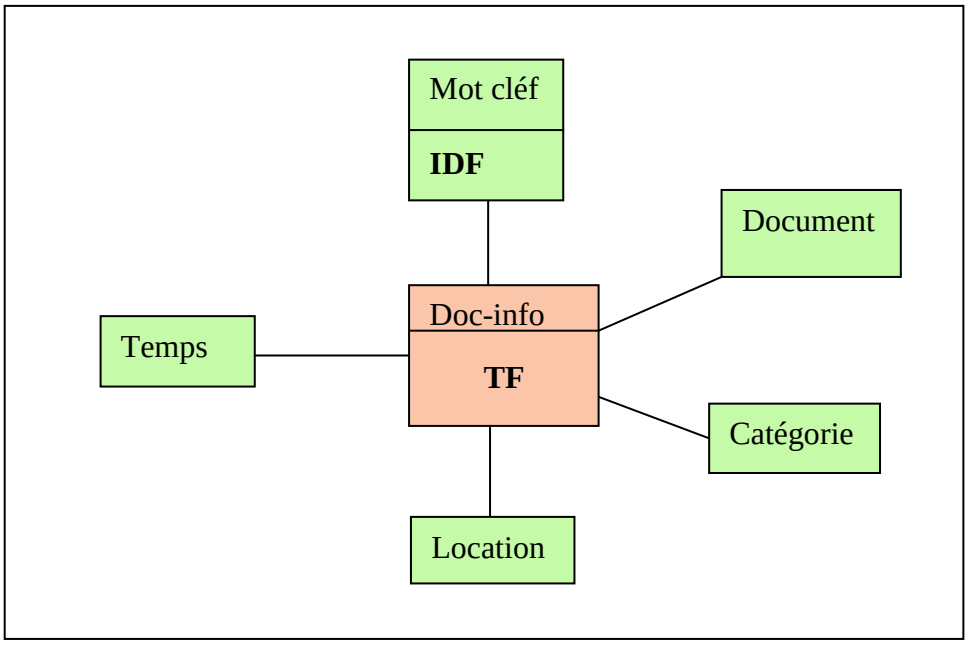

Figure 2.1 : Analyse de documents gouvernementaux [McCabe et al. 2000].

Dans l'exemple illustré dans la figure 2.1, la table de fait *Doc-info* comporte une mesure numérique qui représente le nombre d'apparition d'un mot-clé i dans un document j appelée communément en Recherche d'Information TF (Term Frequency). Un autre poids est affecté au mot-clé mesurant l'importance de ce dernier relativement à toute la collection, le facteur IDF (inverse document frequency).

La manipulation des documents au sein de la base multidimensionnelle construite, se fait par la combinaison du modèle de Recherche d'Information vectoriel et du langage d'interrogation des bases multidimensionnelles MDX. L'alimentation de l'entrepôt de données se fait partiellement grâce à un outil basé sur des techniques de traitement de langage naturel pour l'extraction des métas données des documents tel que : L'auteur, le nom de publication, la date de publication.etc.

L'interrogation de l'entrepôt de document s'effectue par la soumission d'une requête composée de mots clés. Une nouvelle dimension pour abriter cette requête est créée au niveau de l'entrepôt de données et appelée *queryterm. Queryterm* est intégrée au niveau de la requête MDX afin de pouvoir sélectionner l'ensemble des documents satisfaisants à la requête *queryterm* et la requête MDX (Figure 2.2). La sélection des documents satisfaisants se base sur le calcul de la pertinence d'un document en utilisant les deux mesures TF et IDF. Un travail de validation en utilisant la collection TREC constituée à partir de documents d'archives gouvernementales a été présenté.

```
SELECT NON EMPTY { [measures].weight } on columns,
NON EMPTY TopCount (Order([doc_Id].members,
[measures].weight, DESC), 10) on rows
FROM docInfo
WHERE [queryterm]
```

Figure 2.2 : Exemple d'une requête d'interrogation de l'entrepôt de données [McCabe et al, 2000].

Ce travail a été parmi les premiers travaux à avoir traité l'intégration des documents textuels non structurés au niveau des entrepôts des données notamment en utilisant des techniques de la Recherche d'Information. Néanmoins, les possibilités d'analyse proposées sont limitées et se basent sur un comptage de mots clés ; ce qui ne permet pas une analyse OLAP réelle du contenu textuel des documents comme l'agrégation du texte par exemple.

2.1.2 Les travaux de Lee et al

Lee et al proposent dans leurs travaux un Framework de recherche d'information qui permet d'intégrer des dimensions hiérarchiques. En effet, ce Framework permet d'organiser les données selon des dimensions multidimensionnelles tout en utilisant la structure de données « index inversé » [Lee et al, 2002].

Ainsi, les auteurs proposent un nouveau moteur de recherche multidimensionnel « MIRE » qui prend en charge les données structurées et le texte. Plus précisément, il s'agit de la conception d'un moteur de recherche qui puisse prendre en charge la nature hiérarchique des données structurées et qui permette de naviguer selon ces hiérarchies. Pour ce faire, ils adoptent une modélisation multidimensionnelle des données structurées en utilisant un schéma en étoile. Quant au texte, qui est une donnée non structurée, il est intégré en utilisant la structure de donnée « index inversé » bien connue en Recherche d'Information.

Par l'adoption d'une modélisation multidimensionnelle des données au niveau du moteur de recherche, Lee et al estiment ainsi permettre la possibilité d'effectuer des opérations OLAP telle que la navigation à travers la granularité (*Roll-up* et *Drill-down*). Cette navigation va permettre aux utilisateurs de mieux connaitre les sujets traités par la collection des documents ce qui va, selon les auteurs, influer la formulation de la requête en la rendant plus précise et plus expressive par rapport au besoin de l'utilisateur. En effet, la bonne formulation de la requête améliore énormément la qualité des résultats fournis par le moteur de recherche.

Afin de montrer l'intérêt de leur approche, les auteurs ont procédé à une comparaison entre le moteur de recherche proposé « MIRE » et des moteurs de recherches basés sur les différentes stratégies d'indexation qui existent déjà telles que : B-TREE.

Un autre travail de Lee et al [Lee et al, 2003] traite l'intégration de la modélisation multidimensionnelle au sein d'un moteur de recherche d'information. Les auteurs proposent une nouvelle méthode pour l'intégration de dimension hiérarchique au niveau de moteur de recherche. Cette méthode consiste à utiliser les ensembles de synonymies et les relations hyponymes (is-a) définies au niveau du thésaurus WordNet afin d'émaner de nouvelles dimensions hiérarchiques.

2.2 Entrepôt de données textuel

Les approches que nous allons présenter dans cette section ont l'objectif commun de permettre l'intégration et la manipulation du contenu textuel non structuré au sein d'un environnement OLAP. Un critère important qui peut différencier ces approches est le type de l'intégration du contenu textuel au niveau de l'entrepôt de donnée. Ainsi, nous pouvons distinguer deux types d'intégration de données textuelles : l'intégration logique et l'intégration physique des données textuelles.

2.2.1 Intégration logique du contenu textuel au niveau des entrepôts de données

Le but de l'intégration logique du contenu textuel au sein des entrepôts de données est d'essayer d'exploiter les données textuelles non structurées contenues dans des documents textuels qui ne sont pas prises en charge par les entrepôts de données classiques. Cette intégration des données textuelles va enrichir l'entrepôt de données et va fournir au décideur une information supplémentaire en vue de l'aider à mieux comprendre et analyser les faits observés au niveau de ces entrepôts de données. Dans ce cadre, nous allons présenter les travaux de Pérez et al.

2.2.1.1 Les travaux de Pérez et al

Pérez et al proposent un Framework pour l'exploitation des données textuelles documentaires non structurées situées au niveau des documents XML en considérant ces documents comme une source complémentaire d'information [Pérez et al, 2005]. En effet, le travail présenté consiste à construire un entrepôt de données appelé *R-Cube* [Figure 2.3]. Cet entrepôt est composé d'un entrepôt de données classique construit à partir des données structurées et d'un autre entrepôt qui contient les documents décrivant les circonstances des faits observés. De cette façon, Pérez et al espèrent permettre au décideur d'analyser l'ensemble de

l'information documentaire disponible notamment celle sous forme textuelle, qu'elle soit exploitable par un entrepôt de données classique ou non. En d'autres termes, l'entrepôt *R-Cube* proposé permet d'exploiter l'ensemble des données numériques et textuelles structurées et non structurées disponibles.

L'entrepôt de données *R-cube* possède deux nouvelles dimensions : *contexte* et *pertinence*. Chaque fait de l'entrepôt de données a une valeur de pertinence par rapport à un contexte d'analyse donné (un ensemble de contraintes sur les dimensions). La construction du *R-Cube* est effectuée de la façon suivante : une requête RI (un ensemble de mots clés) exprimant le contexte d'analyse est soumise par le décideur. Les documents satisfaisants à la requête soumise sont retrouvés par un moteur de recherche d'information et utilisé par un outil appelé *Fact extractor* pour l'extraction des faits inclus dans ces documents et qui correspondent au contexte d'analyse choisi. Les documents et les faits retournés sont ordonnés relativement à leurs pertinences au contexte d'analyse, cette dernière est évaluée grâce au modèle présenté dans [Pérez et al, 2004].

Ainsi pour chaque scénario d'analyse, l'entrepôt de données *R-cube* mets à la disposition du décideur, pour chaque fait, un ensemble de documents apportant une information complémentaire à l'information offerte par l'entrepôt de données classique pour aider le décideur dans la compréhension des faits observés. De plus, le *R-cube* est supposé capable d'extraire de nouveaux faits à partir des documents. Néanmoins, le *R-Cube* ne permet pas une interrogation du contenu textuel des documents ou sa manipulation à travers des opérations d'analyse OLAP tels que le groupement ou l'agrégation.

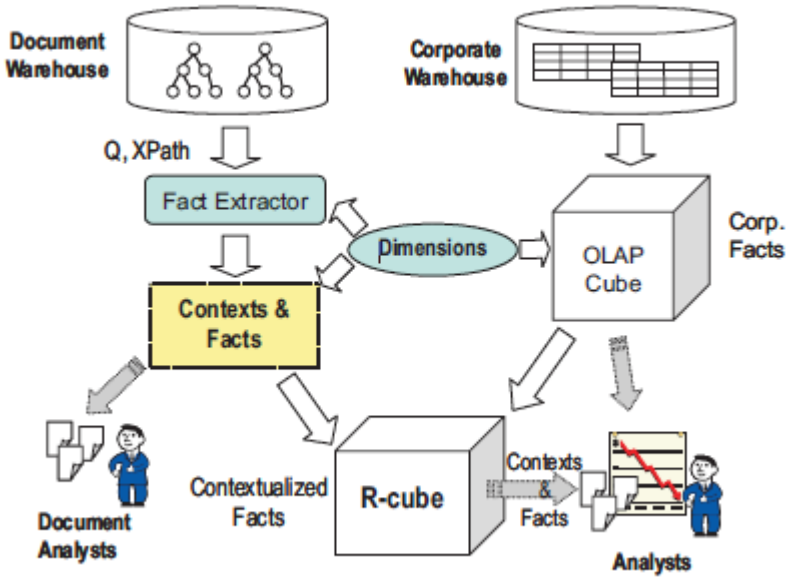

Figure 2.3 : Architecture de l'entrepôt de données *R-Cube* [Pérez et al, 2004].

2.2.2 Intégration physique du contenu textuel au niveau des entrepôts de données

L'objectif de l'intégration physique du contenu textuel au niveau des entrepôts de données est de permettre la manipulation du contenu textuel non structuré avec des opérateurs OLAP. Comme les opérateurs OLAP classiques sont basés sur des fonctions arithmétiques qui ne permettent pas de manipuler la donnée textuelle, de nouveaux opérateurs doivent être proposés en se basant sur des techniques issues du domaine de la Recherche d'Information. Dans ce cadre nous citons :

2.2.2.1 Les travaux de Lin et al

Lin et al proposent un nouveau modèle de cube de données appelé « *Text Cube* » [Lin et al, 2008]. Il s'agit d'une base multidimensionnelle de documents d'opinions des clients sur les produits de la firme DELL. Le modèle *Text cube* supporte deux mesures :

- Une mesure textuelle qui représente l'ensemble de mots clés qui indexent le document (Figure 2.4).
- Une mesure numérique élaborée (vecteur de valeurs) calculée sur la base des deux fonctions de pondération : TF et IDF.

Dimensions				Text Data
M (Model)	P (Price)	T (Y/M/D) (Time)	S (Score)	$\mathcal{DOC}$ (Documents)
m1	p1	2007/07/01	s1	$d_1 = \{w1, w1, w2, w6, w8\}$
m1	p1	2007/07/01	s2	$d_2 = \{w1, w3, w6, w6, w7\}$
m1	p2	2007/08/01	s2	$d_3 = \{w2, w3, w6, w6\}$
m2	p2	2007/08/01	s2	$d_4 = \{w4, w5, w6, w7\}$
m2	p3	2008/06/01	s1	$d_6 = \{w4, w4, w5, w8\}$

Figure 2.4 : Base multidimensionnelle de texte [Lin et al, 2008]

Ce modèle comporte aussi une nouvelle dimension correspondant à la hiérarchie des termes (Figure 2.5). Cette hiérarchie est construite sur la base des liens sémantiques qui relient tous les mots clés utilisés pour indexer les documents du *text cube*. Selon les auteurs, cette hiérarchie est ajoutée afin de permettre de réaliser une analyse sémantique du contenu de la base de documents. Cette hiérarchie est aussi dotée de deux opérateurs permettant le déplacement entre les niveaux de la hiérarchie: pull-up et push-down.

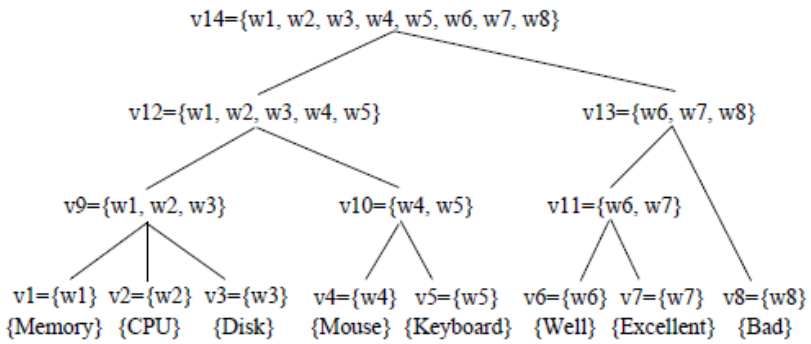

Figure 2.5 : Hiérarchie des termes du domaine [Lin et al, 2008].

L'interrogation du *text cube* se fait de la manière suivante : une requête composée de mots clés et des contraintes sur les dimensions est soumise. Une reformulation de la requête ainsi qu'une agrégation des documents sont effectuées. La reformulation consiste à remplacer chaque mot clé de la requête par ses descendants dans la hiérarchie des termes en utilisant les deux opérateurs proposés : pull-up et push-down. L'évaluation de la requête reformulée par le *text cube* nous donne un ensemble de documents agrégés en un ensemble de mots clés combinés à des valeurs de TF et IDF mises à jour.

2.2.2.2 Les travaux de Ravat et al

Dans ce travail, les auteurs se sont intéressés à la problématique de l'agrégation du contenu textuel au sein d'un environnement OLAP [Ravat et al, 2008]. Plus particulièrement, ils

s'intéressent à la proposition de nouvelles fonctions qui permettent d'agréger les données textuelles comme le permettent les fonctions d'agrégation arithmétiques sur les données numériques. Les auteurs travaillent essentiellement sur des données textuelles issues de documents XML. En effet, ce travail s'articule autour de la proposition d'une fonction d'agréation appelée TOP-KEYWORD. Cette fonction agrège ou résume un ensemble de mots clés en *m* mots clés les plus représentatifs en se basant sur la fonction de pondération *TF-IDF*.

Pour répondre aux spécificités des données textuelles, deux nouveaux types de mesures ont été introduits par les auteurs : la mesure textuelle brute et la mesure textuelle élaborée. La première est une donnée textuelle non formatée par exemple le contenu textuel d'un document. La deuxième est une donnée textuelle résultant d'un prétraitement effectué sur la mesure brute, par exemple l'ensemble de mots clés qui indexent un document.

Afin d'illustrer le fonctionnement de la fonction TOP-KEYWORD, un exemple d'application a été adopté : l'analyse d'une collection d'articles scientifiques [Figure 2.6]. Le sujet « articles » dispose de trois mesures : une mesure numérique *Tx_Accept* qui représente le taux d'acceptation de l'article, une mesure textuelle brute *Texte* qui représente le contenu textuel de l'article scientifique lui-même et finalement une mesure textuelle élaborée *Mots_Clefs* qui représente l'ensemble des mots clés représentatifs de l'article scientifique.

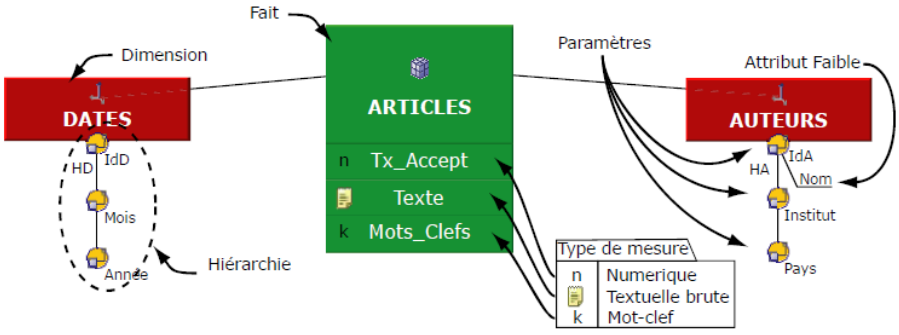

Figure 2.6 : Analyse d'articles scientifiques en utilisant des mesures textuelles [Ravat et al, 2008].

L'opération d'agrégation, proposée dans ce travail, en utilisant la fonction TOP-KEYWORD consiste en trois étapes:

- Prétraitement :

La première étape consiste à formater le contenu du document (mesure brute) à travers quelques opérations de prétraitement tel que : le retrait de mots vides de sens (articles, prépositions, pronoms), utilisation d'ontologie pour la suppression de termes hors domaine, la lemmatisation. Ce formatage du contenu brut du document a pour but d'éliminer des éléments susceptibles de biaiser les calculs et de parasiter les résultats.

- Ordonnancement des termes du document :

Le prétraitement effectué sur le contenu brut du document nous donne un contenu formaté et qui ne contient que des termes influents. L'étape suivante consiste à ordonnancer l'ensemble des termes du document formaté. Cet ordonnancement se fait sur la base d'un poids qui évalue la représentativité du terme au sein du document qui le contient. Ce poids est calculé en utilisant la fonction de pondération bien connue en Recherche d'Information TF-IDF. Cette fonction est le produit entre la représentativité d'un terme dans un document (TF : term frequency) avec l'inverse de sa représentativité dans l'ensemble des documents disponibles (IDF : inverse document frequency).

- Fonction d'agrégation : TOP_KW

Une fois tous les termes du document formaté sont pondérés, un document est alors représenté par une liste ordonnée de termes classés par ordre de poids décroissant. Cette liste est prise en entrée par une fonction d'agrégation qui donne en sortie la liste des *m* principaux termes.

3. Conclusion

Dans ce chapitre, nous nous sommes intéressés aux travaux qui ont traité l'intégration et la manipulation du contenu textuel non structuré au sein de l'environnement OLAP en utilisant des techniques de la Recherche d'Information. Nous avons présenté deux catégories de travaux : les systèmes de recherche d'information multidimensionnels et les entrepôts de données textuels.

CHAPITRE 3
RESSOURCES ET SIMILARITE SEMANTIQUES

3.1 Introduction

Face à l'émergence des sources de données textuelles disponibles de plus en plus nombreuses et volumineuses, il est nécessaire de permettre une description de ces informations non seulement en termes de structure (aspect syntaxique et morphologique) mais aussi en termes de signification et sens véhiculé (aspect sémantique). La description de données peut être effectuée grâce aux ressources sémantiques. Ces dernières permettent en effet d'expliquer le contenu textuel surtout quand ce dernier est ambigu.

En plus de la description du sens véhiculé par un terme, les ressources sémantiques peuvent également identifier les liens sémantiques entre termes tels que : synonymie, hypéronymie ou encore des relations plus spécifiques au domaine. Le présent chapitre a pour objectif de présenter les différents types de ressources sémantiques existantes, notamment le réseau sémantique WordNet. Ce chapitre est organisé comme suit, en première partie, nous allons présenter les ressources sémantiques ; elles sont divisées en quatre types élémentaires : la taxonomie, le thésaurus, le réseau sémantique, et l'ontologie. En deuxième partie, nous allons évoquer la similarité sémantique ainsi que les mesures les plus répandues pour la calculer.

3.2 Ressources sémantiques

3.2.1 Notion de ressources sémantiques

La ressource sémantique représente un vocabulaire contrôlé. Il ressemble un ensemble de termes structurés qu'on appelle concepts. Ces concepts sont reliés par des relations sémantiques. Différentes ressources sémantiques sont utilisées telles que les thésaurus, les bases lexicales et les ontologies. Ce qui différencie principalement ces ressources est l'usage pour lequel elles ont été créées. Nous pouvons différencier quatre types de ressource sémantiques : la taxonomie, le thésaurus, le réseau sémantique, et l'ontologie.

3.2.2 Types de ressources sémantiques

3.2.2.1 Taxonomie

La taxonomie (du grec taxisnomos, taxis : rangement et nomos : loi) est la partie de la biologie visant à établir une classification systématique des êtres vivants. Par extension, le terme désigne une organisation de concepts formant un vocabulaire contrôlé. Une taxonomie se présente sous la forme d'une hiérarchie simple de concepts. Les liens hiérarchiques dans une taxonomie correspondent à des liens de spécialisation /généralisation.

3.2.2.2 Thésaurus

Un thésaurus constitue un dictionnaire hiérarchisé des vocabulaires contrôlés qui rassemble un ensemble de termes structurés choisis pour leur capacité à décrire un domaine. Ces termes, nommés descripteurs, sont utilisés pour décrire d'une manière précise le contenu d'un texte ; ils sont sélectionnés et normalisés. Liés à un domaine donné, les descripteurs présentent les termes génériques ou spécifiques qui dénotent les concepts de ce domaine. Dans un thésaurus, les termes sont organisés dans une hiérarchie de concepts liés par des relations sémantiques. Les relations couramment présentes dans un thesaurus sont des relations : d'hiérarchisation (spécialisation/généralisation), d'équivalence (synonymie), d'association (proximité sémantique, proche-de, relié-à, …) [GAMMOUDI, 1993].

Le thésaurus est exploré habituellement par ordre alphabétique, il présente ses termes selon un classement alphabétique. Un classement hiérarchique de termes ou un classement selon les occurrences des termes peuvent aussi être adoptés. Un thésaurus peut aussi être organisé selon des micro-thésaurus suivant une subdivision de champs sémantiques, où l'exploration va se faire par champs sémantiques.

Un thésaurus s'élabore, soit manuellement par la voie d'une personne ou de plusieurs (grâce à une intelligence humaine), soit de manière automatique. Pour l'élaboration automatique de thésaurus, plusieurs standards existent : pour le développement de thésaurus Monolingue (NISO, 1998 ; ISO, 1986) et pour le développement de thésaurus multilingues (ISO, 1985).Un des méta-thésaurus les plus connus est UMLS (*Unified Medical Language System*) du National Library of Medicine qui définit plus de 40 relations.

3.2.2.3 Réseau sémantique

Les réseaux sémantiques [Quillian, 1968] ont été conçus à l'origine comme un modèle de la mémoire humaine. Un réseau sémantique est un graphe (ou plus précisément multi-graphe)

orienté et étiqueté. Un arc lie (au moins) un nœud de départ à (au moins) un nœud d'arrivée. Les relations vont des relations de proximité sémantique aux relations partie-de, cause-effet, parent-enfant, etc. Les nœuds représentent les concepts, et les arcs liant les nœuds représentent les relations sémantiques qui lient les concepts. L'héritage des propriétés par les liens est matérialisé par un arc (sorte-de) entre les nœuds. Les liens de différentes natures peuvent être mélangés ainsi que les concepts et instances. Un des réseaux sémantiques les plus utilisés est WordNet, largement utilisé pour la Recherche d'Information. WordNet est un réseau lexical et sémantique qui a été initialement élaboré à partir du corpus Brown. Il regroupe les mots selon leur sens dans des synsets.

- WordNet : Exemple d'un réseau sémantique

WordNet est un réseau lexical et sémantique développé depuis 1985 à l'université de Princeton par une équipe de psycholinguistes et de linguistes sous la direction de G. Miller [FELLBAUM, 1998]. A l'origine, WordNet est conçu comme une base lexicale, par la suite, WordNet a été perçu comme un réseau sémantique.

Dans ce réseau sémantique, chaque nœud représente un concept. Un nœud (concept) est constitué par un ensemble de termes synonymes (ou synsets). Ces termes désignent le concept représenté par le nœud. Par exemple, dans WordNet 1.7, le concept anglais *car* est définie à l'aide de cinq synsets :

- *Synset1 : car, auto, automobile, machine, motorcar ;*
- *Synset 2 : car, railcar, railway car, railroad car ;*
- *Synset3 : car, gondola ;*
- *Synset 4: car, elevator car ;*
- *Synset 5: cable car, car.*

Dans WordNet, le lexique est divisé en quatre grandes catégories lexicales : les noms, les verbes, les adjectifs et les adverbes. D'autre part, les concepts sont reliés par des relations sémantiques où la relation de synonymie est la relation sémantique de base dans WordNet. Elle relie les termes d'un même nœud (par exemple les 5 synsets du concept car). Les nœuds (les concepts) peuvent être reliés par d'autres relations sémantiques telles que :

- Relation *Hyperonymie :* C'est le terme générique utilisé pour désigner une classe de concept englobant des instances de classes plus spécifiques. Y est un *hyperonyme* de X si X est un type de (kind of) Y.

- Relation *Hyponymie*: C'est le terme spécifique utilisé pour désigner un membre d'une classe (relation inverse de *Hyperonymie*). X est un *hyponyme* de Y si X est un type de (kind of) Y.
- Relation *Holonymie*: Le nom de la classe globale dont les noms *meronymes* font partie. Y est un *holonyme* de X si X est une partie de (is a part of) Y.
- Relation *Méronymie*: Le nom d'une partie constituante (part of), substance de (substance of) ou membre (member of) d'une autre classe (relation inverse de *l'holonymie*). X est un *méronyme* de Y si X est une partie de Y. exemple : {*voiture*} a pour *méronymes* {{*porte*}, {*moteur*}}.

Le tableau suivant (Tableau 3.1) présente un comptage des relations sémantiques de WordNet 2.1 par catégorie.

Relation	**Entre**	**Nombre**	**Exemple**
Hypernym/Hyponym	Verbe / verbe	13 124	EXHALE / BREATHE
	Nom /nom	75134	CAT/FELINE
Instance Hyponym	Nom / nom	8 515	EIFFEL TOWER / TOWER
Part	Nom / nom	8 874	PARIS / FRANCE
Member	Nom / nom	12 262	ALGERIA / AFRICAN UNION
Substance	Nom / nom	793	SERUM / BLOOD
Attribute	Adjectif / nom	643	INACCURATE / ACCURACY
Verb Group	Verbe / verbe	1 748	GELATINIZE#1 / GELATINIZE#2
Verb Entailment	Verbe / verbe	409	DREAM / SLEEP
Verb Cause	Verbe / verbe	219	ANESTHETIZE / SLEEP
Adjective Similar	Adjectif / adjectif	22 622	DYING / MORIBUND

Tableau 3.1 : Comptage des relations sémantiques de WordNet 1.7 par catégorie [Chaumartin, 2007]

WordNet jouit d'une énorme et grandissante popularité au sein de la communauté scientifique. Les raisons de cette large popularité sont dues au fait que cette base de données lexicale couvre de façon quasi-totale la langue anglaise, ce qui la place souvent en adéquation avec les données générales. Nous l'avons utilisée, pour notre part, sur des données de type presse (journaux et périodiques) lors de l'étape de repérage de liens sémantiques du processus de summarization, afin d'identifier les termes synonymes. Dans sa version 3.0, WordNet contient 155287 termes organisés en 117659 synsets. Le Tableau 3.2 présente des statistiques sur le nombre de mots et de concepts dans WordNet 3.0.

Catégorie	**Mots**	**Concepts**	**Paires Mot-Sens**
Nom	117798	82115	146312
Verbe	11529	13767	25047
Adjectif	21479	18156	30002
Adverbe	4481	3621	5580
Total	155287	117659	206941

Tableau 3.2 : Les statistiques sur le nombre de mots et de concepts dans WordNet 3.0.

WordNet est à la base de nombreux travaux et projets récents, en Recherche d'Information sémantique, qui visent l'accès aux textes par le sens. Parmi ces travaux, nous pouvons citer : EuroWordNet et MultiWordNet. EuroWordNet est un réseau sémantique multilingue couvrant les langues européennes. Il est composé de plusieurs bases lexicales (une pour chaque langue). Les bases lexicales sont connectées à WordNet, afin d'assurer les correspondances des termes dans différentes langues. MultiWordNet est une base lexicale multilingue. Dans cette base, les termes en langue italienne sont des traductions des termes de WordNet 1.6. Les relations sémantiques reliant les concepts sont directement importées de WordNet. La version actuelle de MultiWordNet contient 44,400 termes dans la langue italienne organisés en 35,400 concepts.

3.2.2.4 Ontologie

La définition la plus citée présente l'ontologie comme étant « une spécification explicite et formelle d'une conceptualisation partagée » [Gruber, 1993]. En d'autres termes, une ontologie

est une représentation formelle d'un domaine. C'est une conceptualisation dans le sens où elle fournit un vocabulaire formalisé de concepts et de leurs relations.

Les ontologies permettent, d'une part de décrire les connaissances d'un domaine spécifique et d'autre part de représenter des relations complexes entre les concepts, ainsi que des axiomes et règles qui manquaient aux réseaux sémantiques.

Sur le plan du contenu, nous pouvons distinguer deux types d'ontologie : les ontologies légères et les ontologies lourdes selon la présence ou non d'axiomes [Mothe et al, 2007]. Les ontologies légères sont constituées uniquement de concepts et de relations entre les concepts. Ces ontologies sont dites moins formelles. Contrairement aux ontologies légères, les ontologies lourdes sont dites formelles. Ces ontologies intègrent en plus des concepts et des relations, les règles d'inférence et les axiomes.

Une autre classification d'ontologies selon le type de structures utilisées dans l'ontologie a été proposée [Van et al, 1997] ; elle consiste à classer les ontologies en trois catégories:

- *Les ontologies terminologiques* : qui sont utilisées pour spécifier les termes du vocabulaire d'un domaine de connaissances.
- *Les ontologies d'information* : qui spécifient la structure ou le schéma d'une base de données pour permettre le stockage d'informations.
- *Les ontologies qui modélisent de la connaissance*, elles proposent des structures internes plus riches et qui sont davantage définies en fonction de leurs utilisations comme par exemple le partage d'informations.

3.3 Similarité sémantique

3.3.1 Notion de similarité sémantique

Dans le traitement sémantique du texte, les mesures de similarité sémantique jouent un rôle important dans l'identification des liens sémantiques qui peuvent lier les termes d'un texte. En effet, l'objectif des mesures de similarité sémantique est d'estimer la ressemblance sémantique qui existe entre les concepts auxquels les termes du texte sont rattachés. Un concept réfère à un sens particulier d'un terme donné.

La similarité est la fonction inverse de la distance ; plus deux concepts sont similaires, moins ils sont distants. Une distance est une mesure δ respectant les trois propriétés suivantes:

- nullité de la distance d'un concept avec lui-même : $\delta(ci, ci) = 0$;
- symétrie : $\delta(ci, cj) = \delta(cj, ci)$;

- inégalité triangulaire : δ(ci, cj)+ δ(cj, ck) ≥ δ(ci, ck)

Où ci, cj et ck sont trois concepts quelconques.

D'après les travaux de Tversky, les propriétés mathématiques citées ci-dessus ne concordent pas toujours avec les mesures de similarité sémantiques conformes à la perception humaine surtout pour les propriétés de symétrie et d'inégalité triangulaire [Tversky, 1977]. En Recherche d'Information, il est généralement admis qu'une mesure de similarité doit être réflexive et symétrique [Zargayouna, 2005].

3.3.2 Mesures de similarité sémantique

Plusieurs travaux sur la mesure de similarité sémantique entre les concepts d'une ressource sémantique ont été développés dans différents contextes. Dans la littérature, on distingue trois méthodes de calcul de ces mesures [Slimani et al, 2006] : les méthodes basées sur le comptage des distances d'arcs et la hiérarchie, les méthodes basées sur les nœuds utilisant la notion du contenu informationnel et les méthodes hybrides.

3.3.2.1 Méthodes basés sur le comptage des distances d'arcs

Le principe de calcul de similarité dans cette catégorie est basé sur l'idée suivante : plus le chemin entre deux concepts est court plus ils sont semblables [Rada et al, 1989] [Lee, 1993] [Wup et Palmer, 1994]. Enfin, les mesures de cette catégorie se servent de la structure hiérarchique de l'ontologie pour déterminer la similarité sémantique entre les concepts. L'autre notion qui caractérise cette catégorie de méthodes est que les arcs de la structure représentent des distances uniformes, par conséquent, elle présente l'inconvénient que tous les liens sémantiques possèdent le même poids ce qui impose des difficultés au niveau de la définition et du contrôle des distances des liens. Parmi les travaux classifiés sous cette catégorie, nous citons :

La mesure de Wu-Palmer: Dans une ressource sémantique en hiérarchie, la similarité est définie par rapport à la distance qui sépare deux concepts dans la hiérarchie et également par leur position par rapport à la racine [Wu et Palmer, 1994]. Etant donné c_1 et c_2 deux concepts, la similarité entre c_1 et c_2 est donnée par:

$$sim_{Wu\_Palmer}(c_1, c_2) = \frac{2 * prof(c)}{dist(c_1, c) + dist(c_2, c) + 2 * prof(c)} \qquad (3.1)$$

Où « *c* » est le concept le plus spécifique qui subsume (concept commun le plus spécifique) les deux concepts c_1 et c_2, prof (*c*) est le nombre d'arcs qui sépare *c* de la racine et *Dist(c_i, c)* est le nombre d'arcs qui séparent c_i de *c*.

Mesure de Rada: Cette mesure [Rada et al, 1989] est adoptée dans un réseau sémantique, elle est fondée sur le fait qu'on peut calculer la similarité en se basant sur les liens hiérarchiques «is-a». Pour calculer la similarité de deux concepts, on doit calculer le nombre des arcs minimums qui les séparent. Cette mesure, basée sur le calcul de la distance entre les nœuds par le chemin le plus court, présente un moyen des plus évidents pour évaluer la similarité sémantique dans une ressource sémantique hiérarchique.

La mesure de Resnik : Cette mesure utilise la notion de distance sémantique de la manière suivante [Resnik, 1995] : deux concepts sont d'autant plus similaires que la valeur de la distance sémantique entre eux est faible. La similarité est définie par rapport à la longueur des chemins qui relient deux concepts dans la hiérarchie. La similarité entre c_1 et c_2 est:

$$sim_{ResnikEdge}(c_1, c_2) = 2D - len(c_1, c_2) \quad (3.2)$$

Où *D* est le maximum des longueurs des chemins possibles qui relient c_1 et c_2 et *len(c_1 , c_2)* est le plus petit chemin entre c_1 et c_2.

3.3.2.2 Méthodes basées sur les nœuds

Le principe de calcul de similarité dans cette catégorie est basé sur le contenu informationnel. La notion de contenu informatif (CI) a été pour la première fois introduite par Resnik [Resnik, 1995]. Le contenu informatif d'un concept traduit la pertinence d'un concept dans le corpus en tenant compte de sa spécificité ou de sa généralité. La fréquence de concept dans le corpus est calculée pour retrouver le contenu informatif. Cette fréquence regroupe la fréquence d'apparition du concept lui-même ainsi que des concepts qu'il subsume. Resnik définit le contenu informationnel comme suit [Resnik, 1999]:

$$CI(c) = -log(P(c)) \quad (3.3)$$

P(c) est définie comme la probabilité de retrouver un mot du corpus qui est une instance du concept *c*:

$$P(c) = \frac{Freq(c)}{N} \qquad (3.4)$$

Où *N* est la taille totale d'échantillons de texte et *Freq(c)* est la fréquence d'occurrence des mots dénotant le concept *c* dans la collection.

La mesure de Resnik : La similarité entre deux concepts est liée au contenu informatif qu'ils partagent en commun [Resnik, 1999], indiquée par le plus spécifique concept (c_1 , c_2) qui les subsume. Le concept le plus spécifique est supposé être le concept qui a le contenu informatif le plus grand. La similarité entre deux concepts est proposée comme suit :

$$sim_{ResnikCI}(c_1 , c_2) = CI\left(psc(c_1 , c_2)\right) \qquad (3.5)$$

$psc(c_1 , c_2)$: est le concept le plus spécifique de (c_1 , c_2)

La mesure de Lin: La similarité entre deux concepts est mesurée par le ratio du contenu d'information nécessaire pour mesurer la « communalité » des deux concepts, sur le montant du contenu d'information nécessaire pour décrire chacun des deux concepts [Lin, 1998]. La similarité entre deux concepts c_1 et c_2 est :

$$sim_{Lin}(c_1 , c_2) = \frac{2 \times CI(psc(c_1 , c_2))}{CI(c_1) + CI(c_2)} \qquad (3.6)$$

3.3.2.3 Méthodes hybrides

Ces méthodes sont fondées sur un modèle mixte qui combine entre des approches basées sur le comptage des liens et d'autres approches basées sur le contenu informatif.

Jiang et Conrath : La similarité est définie comme une distance sémantique qui tient compte aussi des contenus informatifs dans la fonction de la similarité [Jiang et Conrath, 1997**]**. La distance sémantique est calculée comme suit :

$$\text{distance}\,(c_1 , c_2) = 2 \times \text{CI}(psc(c_1 , c_2)) - (\text{CI}(c_1) + \text{CI}(c_2)) \qquad (3.7)$$

$$\text{sim}_{\text{Jiang\_Corath}}(c_1 , c_2) = \frac{1}{\text{distance}\,(c_1 , c_2)} \qquad (3.8)$$

3.4 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux ressources sémantiques ainsi qu'aux mesures de calcul de similarité sémantique les plus répandues. D'abord, nous avons présenté quatre types de ressources sémantiques, à savoir : la taxonomie, le thésaurus, le réseau sémantique, et l'ontologie notamment le réseau sémantique WordNet que nous allons utiliser dans la suite de ce travail. Par ailleurs, nous avons cité les mesures de similarité sémantiques les plus utilisées, divisées en trois classes : celles basées sur le comptage des distances d'arcs et la hiérarchie, celles basées sur les nœuds et les mesures hybrides.

CHAPITRE 4
SUMMARIZATION PROPOSEE

4.1 Introduction

La summarization ou l'agréation du texte peut être définie comme le processus qui permet de construire le résumé. Il s'agit de construire un texte à partir d'un ou de plusieurs textes ; ce résumé contiendra le contenu le plus pertinent du texte ou des textes originaux [Lioret, 2006].

Dans le cadre des entrepôts de données, la summarization de données textuelles consiste à synthétiser ces données grâce à des opérateurs OLAP adaptés ayant des fonctions permettant de manipuler les données textuelles et les agréger ; cependant, ces opérateurs adaptés font actuellement défaut. Dans ce cadre, nous proposons un processus qui nous permettra d'agréger du contenu textuel au sein d'un environnement OLAP. Par la suite, et sur la base de ce processus, deux opérateurs OLAP pour l'agrégation du texte seront proposés et présentés dans le chapitre suivant.

Ainsi, le présent chapitre est organisé comme suit : d'abord, nous commençons par une présentation détaillée de notre processus de summarization accompagné d'un exemple d'illustration couvrant les différentes étapes du processus. Une mesure pour l'évaluation de la qualité de la summarization est également proposée. Nous continuons par une pseudo-formalisation du processus d'agrégation à travers un ensemble de pseudos algorithmes qui englobent les différentes étapes de la summarization. Finalement, nous finissons ce chapitre par une conclusion.

4.2 Processus de summarization

Le processus de summarization proposé permet la summarization d'un texte ou d'un ensemble de textes par un ensemble des *k* termes les plus importants. En effet, il permet de sélectionner les termes les plus représentatifs du texte à agréger en se basant sur des techniques de TAL (Traitement Automatique du Langage) et de la Recherche d'Information. Il permet aussi la prise en charge du contenu sémantique du texte lors de la summarization en faisant appel à un thésaurus pour la détection des liens de synonymie existants entre termes. Le processus

proposé se base sur le paragraphe comme unité thématique de base pour l'agrégation, il est composé de quatre étapes élémentaires présentées dans le schéma suivant (Figure 4.1) :

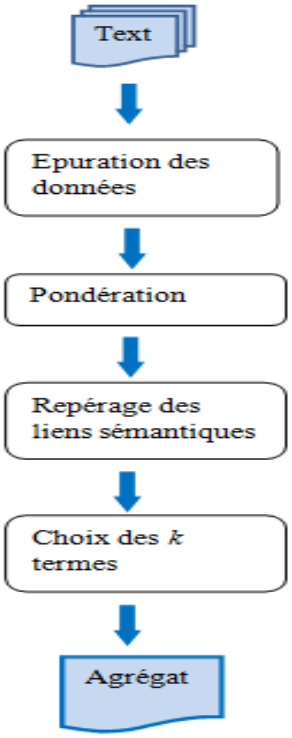

Figure 4.1: Processus de summarization

Afin de mieux expliquer les différentes étapes du processus de summarization et de montrer l'intérêt de chacune d'elles, nous allons utiliser un exemple d'illustration composé de trois (03) textes à agréger sur lesquels nous allons appliquer le processus de summarization. Il s'agit de trois textes tirés à partir d'une collection d'articles scientifiques qui portent sur les documents semi structurés XML.

Texte1 :

XML (Extensible Markup Language) is a flexible text format allowing the interchange and the representation of complex data. Finding an appropriate model for an XML data warehouse tends to become complicated as more and more solutions appear.
In this paper we present some different frameworks that use XML within data warehousing technology.
These proposals range from using XML data sources for regular warehouses to those using full XML warehousing resolutions Some researches papers merely focus on document storage facilities while others present adaptations of XML technology for OLAP and computation frameworks.

Texte2 :

We propose in this paper a definition for cube adapted for XML data warehouse, including a suitably generalized specification mechanism. We define a cube lattice over the aggregates so defined.
We then identify properties of the cube lattice that can be leveraged to allow optimized computation of the cube. We present the results of an extensive performance evaluation experiment in a cube gauging the behavior of alternative algorithms for cube computation.

Texte3 :

For the multidimensional analysis of XML documents, we need to construct an XML cube from them like the way we do for relational data In this paper, we propose a method for building XML cubes from an XML warehouse where XML documents constitute both fact and dimension data.
We also propose to build a special index called the Hierarchically-Structured Key Index for each dimension to associate dimension and fact XML documents for multidimensional access. We demonstrate the construction method for the specific XML cubes specified by a multidimensional expression language for XML documents.

Notre texte à agréger, formé par les 3 textes cités ci-dessus, est décrit par un ensemble de descripteurs choisis par un expert du domaine. Cet ensemble de descripteurs va nous aider plus tard à évaluer le processus de summarization proposé, il est composé des huit (08) termes suivants :{ multidimensional, model, cube, data, text, warehouse, analysis, method}.

4.2.1. Epuration des données

Le but de notre processus de summarization est de sélectionner un ensemble de termes représentatifs du texte ou de l'ensemble des textes à agréger. L'étape d'épuration de données est une étape de préparation du texte pour la suite du processus de summarization. Elle permet d'abord de découper le texte en un ensemble de paragraphes, puis d'éliminer les termes non susceptibles de faire partie de la liste des termes représentatifs. Finalement, l'ensemble des termes restants sont normalisés et homogénéisés afin de lever certaines ambiguïtés linguistiques à travers un ensemble de traitements morphologiques, syntaxiques et même sémantiques. L'épuration du texte à agréger consiste en quatre opérations élémentaires qui sont : le découpage du texte en paragraphes, la tokenisation, la suppression de mots vides et la lemmatisation.

4.2.1.1.Découpage du texte en paragraphes

Notre processus de summarization a pour but de permettre l'agrégation dans un environnement OLAP. Les analyses OLAP notamment l'agrégation doivent particulièrement être rapides et efficaces d'où vient l'intérêt de traiter le texte comme une unité susceptible d'être décomposée en unités plus fines tout en assurant la cohérence de ces nouvelles unités. En effet, cette décomposition a de nombreux avantages spécialement dans le cas des textes longs: d'une part, elle permet une représentation plus fine et moins encombrante du contenu textuel. D'une autre part, elle permet un accès plus précis et plus localisé à l'information pertinente. Ainsi, la summarization doit se faire non plus sur le texte dans sa globalité mais sur des unités plus petites.

Ainsi, nous devons procéder au découpage du texte ou des textes à agréger en un ensemble d'unités de texte. Ces unités doivent être autonomes du point de vue sémantique et syntaxique afin d'assurer la cohérence de la summarization. Nous pouvons distinguer plusieurs approches de découpage de texte [Lallich et al, 1998] :

- Découpage en une suite de mots.
- Découpage en phrases.
- Découpage en paragraphes.
- Découpage en unités logiques répercutées dans le sommaire.

Ainsi, l'unité de découpage du texte à agréger peut être : la suite de mots, la phrase, le paragraphe ou une unité logique déduite à partir du sommaire. Dans notre cas, le texte à agréger est un texte qui ne dispose pas généralement de sommaire, par conséquent, nous ne pouvons pas envisager de le découper en unités logiques déduite à partir du sommaire. D'autre part, le découpage en suites de mots est réalisé d'une manière arbitraire, il ne prend pas l'aspect syntaxique ou sémantique de la suite de mots en considération. De même, la décomposition en phrases ne présente pas de garantie de complétude syntaxique et sémantique.

En revanche, le découpage du texte en paragraphes se présente comme relativement plus fiable que le découpage en suite de mots ou en phrases. Le paragraphe est une unité cohérente dans le sens où chaque paragraphe désigne un "thème" d'un point de vue particulier mais reste une unité plus fine que le texte tout entier. De là, nous avons décidé d'adopter le paragraphe comme unité thématique d'agrégation. Notre texte sera découpé en paragraphes en se basant sur la ponctuation et le sens.

Pour ce faire, nous considérons dans ce travail que le paragraphe est un bloc de texte (chaine de caractères) délimité par deux alinéas [Bessonnat, 1988]. Ainsi, et à chaque nouveau alinéa, nous avons un nouveau paragraphe. En d'autre termes, nous considérons que le paragraphe est un ensemble d'éléments (termes, signes de ponctuation, etc.) dont l'alinéa constitue l'ouverture et la fermeture de cet ensemble. Selon le cas, un alinéa peut correspondre à : un simple saut de ligne, un saut de ligne avec indentation, plusieurs sauts de lignes, .etc.

Ainsi, cette première étape d'épuration de données consiste à découper le texte à agréger en paragraphes en considérant l'alinéa comme séparateur de paragraphes. A l'issue de cette étape nous aurons notre texte à agréger transformé en un ensemble de paragraphes.

- Application sur l'exemple d'illustration :

Lors de cette étape, nous allons découper les textes de notre exemple d'illustration en un ensemble de paragraphes, nous obtenons le résultat suivant :

Texte à agréger :

Paragraphe 1: XML (Extensible Markup Language) is a flexible text format allowing the interchange and the representation of complex data. Finding an appropriate model for an XML data warehouse tends to become complicated as more and more solutions appear.
Paragraphe 2: In this paper we present some different frameworks that use XML within data warehousing technology.
Paragraphe 3: These proposals range from using XML data sources for regular warehouses to those using full XML warehousing resolutions. Some researches papers merely focus on document storage facilities while others present adaptations of XML technology for OLAP and computation frameworks.
Paragraphe 4: We propose in this paper a definition for cube adapted for XML data warehouse, including a suitably generalized specification mechanism. We define a cube lattice over the aggregates so defined.
Paragraphe 5: We then identify properties of the cube lattice that can be leveraged to allow optimized computation of the cube. We present the results of an extensive performance evaluation experiment in a cube gauging the behavior of alternative algorithms for cube computation.
Paragraphe 6 : from them like the way we do for relational data In this paper, we propose a method for building XML cubes from an XML warehouse where XML documents constitute both fact and dimension data.
Paragraphe 7: We also propose to build a special index called the Hierarchically-Structured Key Index for each dimension to associate dimension and fact XML documents for multidimensional access. We demonstrate the construction method for the specific XML cubes specified by a multidimensional expression language for XML documents

A l'issue de l'étape du découpage du texte en paragraphes, le contenu textuel à agréger est formé de sept (07) paragraphes.

4.2.1.2 Tokenisation:

Après avoir décomposé le texte en un ensemble de paragraphes, l'étape de tokenisation consiste à reconnaitre l'ensemble des termes qui composent le texte, cela revient évidemment à reconnaitre les différents termes qui composent l'ensemble des paragraphes du texte. D'une manière générale, la tokenisation sert à segmenter un texte en « token ». Il n'y a pas que les termes qui sont considérés comme des tokens mais aussi : les chiffres, les signes de ponctuation, les parenthèses et les guillemets.

Par ailleurs, la tokenisation se base aussi sur la notion de séparateur. Les langages alphabétiques (arabe, anglais, français, etc.) utilisent habituellement le caractère « espace » pour séparer les termes. Ainsi, la tokenisation d'un texte peut être simplement réalisée en considérant l'espace comme séparateur de termes et en éliminant les signes de ponctuation, les parenthèses, et les guillemets qui peuvent se retrouver aux extrémités d'un terme. Cette méthode simple est tout à fait efficace, car l'espace et les signes de ponctuation sont des indicateurs assez fiables des extrémités d'un terme [Christopher et al, 2009]. A l'issue de cette étape, nous aurons la liste des termes composant chacun des paragraphes du texte.

- Application sur l'exemple d'illustration :

Dans cette étape, nous allons essayer d'identifier les différents termes qui composent chacun des sept paragraphes du texte à agréger. Le résultat de la tokenisation sur les paragraphes 1,2 et 3 est présenté ci-dessous:

Texte à agréger :

Pararaphe 1= {XML, Extensible, Markup, Language, is, a, flexible, text, format, allowing, the, interchange, and, the, representation, of, complex, data, Finding, an, appropriate, model, for, an, XML, data, warehouse, tends, to, become, complicated, as, more, and, more, solutions, appear}
Paragraphe 2 = {In, this, paper, we, present, some, different, frameworks, that, use, XML, within, data, warehousing, technology}
Paragraphe 3 = {These, proposals, range, from, using, XML, data, sources, for, regular, warehouses, to, those, using, full, XML, warehousing, resolutions, Some, researches, papers, merely, focus, on, document, storage, facilities, while, others, present, adaptations, of, XML, technology, for, OLAP, and, computation, frameworks}

4.2.1.3 Suppression de termes vides

Cette étape sert à filtrer la liste des termes de chacun des paragraphes du texte en éliminant les termes vides. Les termes vides sont tous les termes non discriminants pour notre texte assemblés dans une liste appelée «stop liste ».

En effet, nous avons trois types de termes vides : les termes non porteurs de sens, les termes athématiques et les termes communs. Les termes non porteurs de sens comprennent : les articles, les pronoms, les prépositions et les mots outils. Les termes athématiques sont les termes qui exposent le sujet sans le traiter comme : contenir, appartenir.etc. Pour ces deux types de termes vides, nous pouvons avoir une liste fixe de termes vides pour chaque langue que nous pouvons utiliser dans le traitement textuel.

Quant aux termes communs, ceux sont les termes qui se répètent régulièrement dans des textes réunis autour d'un thème commun et qui sont liés directement à ce thème ce qui leur fait perdre leur pouvoir discriminatoire même s'ils sont fréquents. Ainsi, nous avons besoin d'ajouter ces termes à la liste fixe des termes non porteurs de sens et des termes athématiques pour avoir une liste de termes vides complète.

Dans ce travail, nous avons choisi de travailler avec la stop liste de Fox [Fox, 1990] qui comprend les termes non porteurs de sens et les termes athématiques de la langue anglaise. La figure suivante (Figure 4.2) présente la liste des termes vides que propose Fox.

1 a	36 b	71 did	106 felt	141 h
2 about	37 back	72 differ	107 few	142 had
3 above	38 backing	73 different	108 find	143 has
4 across	39 backs	74 differently	109 finds	144 have
5 after	40 be	75 do	110 first	145 having
6 again	41 because	76 does	111 for	146 he
7 against	42 become	77 done	112 four	147 her
8 all	43 becomes	78 down	113 from	148 herself
9 almost	44 became	79 downed	114 full	149 here
10 alone	45 been	80 downing	115 fully	150 high
11 along	46 before	81 downs	116 further	151 higher
12 already	47 began	82 during	117 furthered	152 highest
13 also	48 behind	83 e	118 furthering	153 him
14 although	49 being	84 each	119 furthers	154 himself
15 always	50 beings	85 early	120 g	155 his
16 among	51 best	86 either	121 gave	156 how
17 an	52 better	87 end	122 general	157 however
18 and	53 between	88 ended	123 generally	158 i
19 another	54 big	89 ending	124 get	159 if
20 any	55 both	90 ends	125 gets	160 important
21 anybody	56 but	91 enough	126 give	161 in
22 anyone	57 by	92 even	127 given	162 interest
23 anything	58 c	93 evenly	128 gives	163 interested
24 anywhere	59 came	94 ever	129 go	164 interesting
25 are	60 can	95 every	130 going	165 interests
26 area	61 cannot	96 everybody	131 good	166 into
27 areas	62 case	97 everyone	132 goods	167 is
28 around	63 cases	98 everything	133 got	168 it
29 as	64 certain	99 everywhere	134 great	169 its
30 ask	65 certainly	100 f	135 greater	170 itself
31 asked	66 clear	101 face	136 greatest	171 j
32 asking	67 clearly	102 faces	137 group	172 just
33 asks	68 come	103 fact	138 grouped	173 k
34 at	69 could	104 facts	139 grouping	174 keep
35 away	70 d	105 far	140 groups	175 keeps
176 kind	211 mr	246 once	281 problem	316 sides
177 knew	212 mrs	247 one	282 problems	317 since
178 know	213 much	248 only	283 put	318 small
179 known	214 must	249 open	284 puts	319 smaller
180 knows	215 my	250 opened	285 q	320 smallest
181 l	216 myself	251 opening	286 quite	321 so
182 large	217 n	252 opens	287 r	322 some
183 largely	218 necessary	253 or	288 rather	323 somebody
184 last	219 need	254 order	289 really	324 someone
185 later	220 needed	255 ordered	290 right	325 something
186 latest	221 needing	256 ordering	291 room	326 somewhere
187 least	222 needs	257 orders	292 rooms	327 state
188 less	223 never	258 other	293 s	328 states
189 let	224 new	259 others	294 said	329 still
190 lets	225 newer	260 our	295 same	330 such
191 like	226 newest	261 out	296 saw	331 sure
192 likely	227 next	262 over	297 say	332 t
193 long	228 no	263 p	298 says	333 take
194 longer	229 non	264 part	299 second	334 taken
195 longest	230 not	265 parted	300 seconds	335 than

196 m	231 nobody	266 parting	301 see	336 that
197 made	232 none	267 parts	302 sees	337 the
198 make	233 nothing	268 per	303 seem	338 their
199 making	234 now	269 perhaps	304 seemed	339 them
200 man	235 nowhere	270 place	305 seeming	340 then
201 many	236 number	271 places	306 seems	341 there
202 may	237 numbers	272 point	307 several	342 therefore
203 me	238 o	273 pointed	308 shall	343 these
204 member	239 of	274 pointing	309 she	344 they
205 members	240 off	275 points	310 should	345 thing
206 men	241 often	276 possible	311 show	346 things
207 might	242 old	277 present	312 showed	347 think
208 more	243 older	278 presented	313 showing	348 thinks
209 most	244 oldest	279 presenting	314 shows	349 this
210 mostly	245 on	280 presents	315 side	350 those
351 though	365 turning	379 w	393 when	407 worked
352 thought	366 turns	380 want	394 where	408 working
353 thoughts	367 two	381 wanted	395 whether	409 works
354 three	368 u	382 wanting	396 which	410 would
355 through	369 under	383 wants	397 while	411 y
356 thus	370 until	384 was	398 who	412 year
357 to	371 up	385 way	399 whole	413 years
358 today	372 upon	386 ways	400 whose	414 yet
359 together	373 us	387 we	401 why	415 you
360 too	374 use	388 well	402 will	416 young
361 took	375 uses	389 wells	403 with	417 z
362 toward	376 used	390 went	404 within	
363 turn	377 v	391 were	405 without	
364 turned	378 very	392 what	406 work	

Figure 4 .2 : La Stop liste proposée par Fox [Fox, 1990]

- Application sur l'exemple d'illustration :

Comme les textes de notre exemple d'illustration portent sur les documents XML semi structurés, nous avons jugé utile d'ajouter la liste de termes vides présentée ci-dessous à la liste de Fox. En effet, cette liste contient la liste des termes communs relatifs au domaine des recherches sur les documents XML et qui sont par conséquent non discriminants comme ça été expliqué précédemment dans cette section.

Liste des termes communs = {xml, document, documents, structure, research, researches, paper, papers, definition, technology}. Ainsi, le résultat de la suppression des termes vides sur le texte à agréger est le suivant :

Texte à agréger :

Pararaphe 1={Extensible, Markup, Language, flexible, text, format, allowing, interchange ,representation, complex, data, Finding, appropriate, model, data, warehouse, tends, more, complicated, solutions, appear}
Paragraphe 2 = {data, frameworks, warehousing}
Paragraphe 3 = {proposals, range, using, data, sources, regular, warehouses, using, warehousing, resolutions, merely, focus, storage, facilities, adaptations, OLAP, computation, frameworks}
Pararaphe 4 = {propose, cube, adapted, data, warehouse, including, suitably, generalized, specification, mechanism, define, cube, lattice, aggregates, defined}
Paragraphe 5 = {identify, properties, cube, lattice, leveraged, allow, optimized, computation, cube, results, extensive, performance, evaluation, experiment, cube, gauging, behavior, alternative, algorithms, cube, computation}
Paragraphe 6 = {multidimensional, analysis, construct, cube, relational, data, propose, method, building, cubes, warehouse, constitute, dimension, data}
Paragraphe 7 = {propose, build, special, dimensions, called, hierarchically-structured, key, index, dimension, associate, dimension, multidimensional, access, demonstrate, construction, method, specific, cubes, specified, multidimensional, expression, language}

4.2.1.4 Lemmatisation

Il s'agit de la quatrième étape de l'épuration de données. Lors de cette étape, notre but est de regrouper les termes qui font partie d'une même famille et qui partagent un même radical. Pour ce faire, et comme nous l'avons vu précédemment (cf. section1.2.2.2.3), nous pouvons choisir la lemmatisation comme nous pouvons aussi choisir d'appliquer la radicalisation.

Pour ce travail, nous avons opté pour la lemmatisation. Chaque terme est remplacé par son lemme associé qui est déduit par le lemmatiseur. Le choix de la lemmatisation au lieu de la radicalisation dans notre processus de summarization sémantique est justifié par la cohérence linguistique qu'offre la lemmatisation. En effet, Le lemme est un terme réel, qui appartient au vocabulaire de la langue, et qui peut être sujet à des comparaisons sémantiques, contrairement au radical.

La lemmatisation est réalisée en utilisant l'étiqueteur morphosyntaxique « TreeTagger » [Schmid, 2011]. TreeTagger est un outil de lemmatisation en contexte, il génère pour chaque terme son lemme associé et sa catégorie grammaticale (nom, verbe, adjectif,..). Remplacer chaque terme par son lemme associé va nous permettre de lever de nombreuses ambigüités liées aux flexions de genre et de nombre. Quant à la catégorie grammaticale, elle va nous servir à effectuer un filtrage sur les termes.

En effet, nous jugeons que les termes susceptibles d'être représentatifs, appartiennent généralement à certaines catégories grammaticales telles que : nom, verbe, ou adjectif. Notre lemmatisation est flexible, l'utilisateur peut aisément choisir et changer la liste des catégories grammaticales filtrées selon les besoins de la summarization effectuée. Par ailleurs, ce filtrage

va alléger le processus de summarization en diminuant du volume du texte traité lors des étapes restantes.

A la fin de cette première étape d'épuration de données, le texte à agréger est un ensemble de paragraphes où chacun est composé d'une liste de lemmes (termes).

- Application sur l'exemple d''illustration :

L'application de la lemmatisation sur les textes de notre exemple d'illustration donne les résultats présentés ci-dessous. Notons que nous avons choisi de ne garder que les deux catégories grammaticales suivantes : nom et adjectif. Les termes qui ont été changés par la lemmatisation sont marqués en bleu.

Texte à agréger :

Paragraphe 1 = {extensible, markup, language, flexible, text, format, representation, complex, datum, finding, appropriate, model, datum, warehouse, complicated, more, solution}
Paragraphe 2 = {framework, datum}
Paragraphe 3 = {proposal, range, datum, source, regular, warehouse, resolution, focus, storage, facility, adaptation, OLAP, framework, computation}
Paragraphe 4 = {cube, datum, warehouse, specification, mechanism, define, cube, lattice, aggregate}
Paragraphe 5 = {property, cube, lattice, leveraged, computation, cube, result, extensive, performance, evaluation, experiment, cube, behavior, alternative, algorithm, cube, computation}
Pararaphe 6 = {multidimensional, analysis, cube, relational, datum, method, building, cube, warehouse, dimension, datum}
Paragraphe 7 = {special, dimension, hierarchically-structured, key, index, dimension, associate, dimension, multidimensional, access, construction, method, specific, cube, multidimensional, expression, language}

4.2.2 Pondération

Une fois la lemmatisation réalisée, le texte à agréger est épuré. La prochaine étape de la summarization est de déterminer la représentativité (l'importance) de chacun des termes au sein du texte afin de pouvoir classer les termes selon leur représentativité et choisir ainsi les meilleurs d'entre eux.

A l'issue de cette étape, nous aurons une liste ordonnée de termes pondérés. Afin d'arriver à cette liste, nous devons d'abord quantifier la représentativité de chaque terme du texte. Une des méthodes les plus connues en Recherche d'Information pour la quantification de la représentativité d'un terme au sein d'un document est *TF-IDF* [Salton et al, 1975]. *TF-IDF* est une formule de pondération qui se base sur la notion de fréquence (nombre d'apparition) d'un terme dans un texte (cf. section 1.2.3). Elle est le résultat de la combinaison des deux types de fréquence : La fréquence locale *tf* et la fréquence globale *idf*.

- tf_{ij} : est la fréquence du terme t_i au sein du document D_j, elle est définit comme suit :

$$tf_{ij} = \frac{tf_{ij}}{\sum_{t_i \in D_j} tf_{ij}} \qquad (4.1)$$

Où $\sum_{t_i \in D_j} tf_{ij}$ est le nombre d'occurrences de tous les termes dans le document D_j. Cette version normalisée de la formule de calcul de tf_{ij} a l'avantage de réduire les différences entre les valeurs de fréquence associées aux termes ; elle prend ses valeurs dans l'intervalle[0,1].

- *Idf* est la fréquence globale du terme, elle représente l'importance du terme au sein de la collection du document.

$$Idf_i = \log\left(\frac{N}{n_i}\right) \qquad (4.2)$$

Où *N* est le nombre de documents de la collection et n_i est le nombre de documents où le terme t_i apparait.

Finalement, le poids *TF-IDF* s'obtient par le produit des deux fréquences *tf* et *idf* comme suit :

$$TF - IDF = tf * idf \qquad (4.3)$$

Traditionnellement, *TF-IDF* sert à quantifier l'importance d'un terme dans un document par rapport à une collection de documents. Ainsi, le document est l'unité lexicale choisie pour la pondération *TF-IDF* en Recherche d'Information. Une adaptation de la formule à notre problématique de summarization du texte consiste à choisir le paragraphe comme une unité lexicale au lieu du document. En effet, le texte est généralement composé d'un ensemble de paragraphes où chaque paragraphe présente une seule idée principale. De la, on peut considérer le paragraphe comme un fragment textuel lexicalement indépendant. La formule *TF-IDF* va nous servir à calculer la représentativité d'un terme dans un paragraphe par rapport à l'ensemble du texte qui est une collection de paragraphes. Les formules de calcul de *tf* et *idf* adaptées sont les suivantes:

$$tf_{ij} = \frac{tf_{ij}}{\sum_{t_i \in p_j} tf_{ij}} \qquad (4.4)$$

Où $\sum_{t_i \in p_j} tf_{ij}$ est la somme des valeurs tf_{ij} des termes du paragraphe p_j.

$$Idf_i = \log\left(\frac{N}{n_i}\right) \qquad (4.5)$$

Où N est le nombre total de paragraphes et n_i est le nombre de paragraphes où le terme t_i apparait.

Ainsi, lors de cette étape de pondération, nous allons calculer pour chaque terme son poids de représentativité en utilisant la formule de pondération *TF-IDF* adaptée et l'organisation du texte en paragraphes. A l'issue cette étape, le texte à agréger est représenté par une liste de termes pondérés.

- Application sur l'exemple d'illustration :

Nous allons calculer les deux fréquences locale et globale pour le texte de notre exemple d'illustration selon les formules présentées précédemment. Les tableaux suivants contiennent respectivement les résultats obtenus pour le calcul de la fréquence locale tf_{ij} et la fréquence globale Idf_i pour quelques termes du texte à agréger. En effet, par manque d'espace et afin de ne pas encombrer cette section, nous n'allons pas présenter les résultats du calcul des fréquences globales et locales pour tous les termes du texte.

Termes	***Tf***						
	p_1	p_2	p_3	p_4	p_5	p_6	p_7
markup	0.06	0	0	0	0	0	0
datum	0.12	0.50	0.07	0.13	0	0.18	0
……..							
framework	0	0.50	0.07	0	0	0	0
.....							
warehouse	0.06	0	0.07	0.13	0	0.09	0
resolution	0	0.07	0	0	0	0	0
model	0.06	0	0	0	0	0	0
……							
mechanism	0	0	0	0.13	0	0	0
cube	0	0	0	0.25	0.24	0.18	0.06
……							
result	0	0	0	0	0.06	0	0
computation	0	0.07	0	0	0.12	0	0
representation	0.06	0	0	0	0	0	0
……							
multidimensional	0	0	0	0	0	0.09	0.12
relational	0	0	0	0	0	0.09	0
…….							
Method	0	0	0	0	0	0	0.06

Tableau 4.1 : Résultats du calcul de la fréquence locale tf_{ij} pour l'exemple d'illustration

Terme	***Idf***
markup	0.85
datum	0.15
……..	
framework	0.54
.....	
warehouse	0.24
resolution	0.85
model	0.84
……	
mechanism	0.85
cube	0.24
……	
result	0.85
computation	0.54
representation	0.84
……	
multidimensional	0.54
relational	0.85
…….	
method	0.54

Tableau 4.2 : Résultats du calcul de la fréquence globale *Idf.*

Une fois les deux fréquences *tf* et *idf* calculées, nous pouvons calculer le poids *TF-IDF* pour chaque terme. Les huit (08) premiers termes de la liste des termes du texte à agréger, après classement selon le poids sont les suivants :

position	***terme***	***TF-IDF***
1	framework	0.31
2	cube	0.18
3	datum	0.14
4	dimension	0.14
5	multidimensional	0.11
6	aggregate	0.11
7	specification	0.11
8	mechanism	0.10

Tableau 4.3 : Résultat de la summarization

4.2.3 Repérage de liens sémantiques

La formule de pondération *TF-IDF* donne une bonne évaluation du pouvoir représentatif d'un terme dans un texte. En effet, il s'agit d'une formule qui a prouvé sa robustesse et son efficience dans le traitement des documents textuels [Robertson, 2004]. Néanmoins, *TF-IDF* ignore un aspect important du texte qui est sa sémantique. En effet, *TF-IDF* ne s'intéresse qu'à l'apparition ou non du terme dans un texte ; elle considère le terme seulement comme une suite de caractères, son sens et sa sémantique sont totalement ignorés dans le calcul de son poids de représentativité.

Par exemple, si plusieurs termes synonymes apparaissent en même temps dans le même texte, le poids de chacun de ces termes est calculé indépendamment de la présence des autres termes qui lui sont synonymes. Or, si un terme a un ou plusieurs de ses synonymes présents avec lui, cela augmente raisonnablement de son importance et de l'importance du contenu sémantique qu'il véhicule et par conséquent, le poids qui est lui assigné doit être plus important.

Afin de permettre la prise en charge de la sémantique du texte lors de sa summarization, nous allons essayer de comptabiliser les liens sémantiques qui existent entre les termes d'un même texte, en ajustant les poids de représentativité associés aux termes selon ces liens sémantiques. Pour ce faire, nous avons besoin d'abord d'identifier tous les liens de synonymie qui peuvent exister entre les termes du texte. Une ressource lexico-sémantique peut alors être utilisée pour identifier ces liens de synonymie.

Par ailleurs, l'utilisation de ressources sémantiques (thésaurus, ontologies, etc.) peut être d'un grand intérêt pour le traitement du texte et spécialement en utilisant des techniques de la Recherche d'Information. L'efficacité de cette identification de liens sémantiques dépend fortement de la richesse de la ressource sémantique, cette dernière doit couvrir tous les termes du contenu textuel traité, d'où la nécessité d'avoir une ressource sémantique disposant d'une terminologie suffisamment riche pour couvrir le domaine dont traite le contenu textuel à agréger.

Dans le cas de domaine général, la base de données lexicale Word Net est une des ressources les plus utilisées en Recherche d'Information, notamment dans le cas de collections de documents de type "news", tels que les journaux et les magazines. D'autres bases de données lexicales existent pour des domaines spécifiques tels que le réseau conceptuel médical UMLS pour le domaine médical.

Ainsi, et grâce à la ressource sémantique, nous pouvons déterminer le degré de similarité sémantique de deux termes en calculant leur distance sémantique. La distance sémantique est une valeur numérique calculée sur la base des liens taxonomiques qui lient les termes au sein de la ressource sémantique [Rada et al, 1989]. Cette distance est une mesure sémantique qui peut nous indiquer si deux termes sont similaires sémantiquement ou pas ; plus cette distance est petite, plus les deux termes comparés sont considérés comme synonymes.

Dans la littérature, plusieurs travaux sur la mesure de la similarité sémantique en utilisant une ressource sémantique ont été développés [Slimani et al, 2007]. Dans ce travail, nous allons utiliser la mesure de similarité de lin [Lin, 1998]. Cette mesure est basée sur la notion du contenu informationnel introduite initialement par Resnik [Resnik, 1995] et qui se base conjointement sur le corpus et sur la ressource sémantique. La mesure de similarité est définie comme suit :

$$Sim(X,Y) = \frac{2 * \log(P(CS(X,Y)))}{\log(P(X)) + \log(P(Y))} \qquad (4.6)$$

Où :

- X et Y sont des concepts de la ressource sémantique
- CS(X, Y) représente le concept le plus spécifique (qui maximise la valeur de similarité) qui subsume (situé à un niveau hiérarchique plus élevé) les deux concepts X et Y dans la ressource sémantique.

Le repérage de liens sémantiques va se faire comme suit, d'abord nous allons calculer la distance sémantique entre tous les termes résultants de l'étape précédente de pondération deux à deux, en utilisant le thésaurus WordNet 2.1[Chaumartin, 2007]. La distance calculée sera comparée à un seuil de similarité α. Si la distance résultante est inférieure à α, nous considérons les deux termes comme similaires. Pour chaque deux termes dits similaires, nous ne gardons qu'un seul et nous éliminons l'autre. Le choix du terme à garder se fait par rapport aux poids des deux termes, nous gardons le terme qui a le poids le plus fort, avec un poids correspondant à la somme de son ancien poids et le poids du terme similaire qui a été éliminé.

Par ailleurs, la détermination du seuil α doit être faite d'une manière non aléatoire. Une façon de faire est de déterminer cette valeur expérimentalement ; pour cela, nous allons utiliser un corpus de test composé de 20 termes tirés du thésaurus WordNet 2.1. Après le calcul de la similarité sémantique pour tous les couples de termes, des sujets humains vont évaluer les valeurs de similarité obtenues relativement aux couples de termes considérés. Finalement, sur la base des évaluations faites, nous allons fixer la valeur du seuil α.

A la fin de cette étape, nous aurons une liste de termes pondérés, où le poids d'un terme ne représente pas seulement l'importance de la présence du terme mais aussi celle de son contenu sémantique. La prochaine étape consiste finalement à choisir les *k* termes les plus représentatifs.

- Application sur l'exemple d'illustration :

L'étape de repérage de liens sémantiques sert à identifier les liens de synonymie qui existent entre les termes du texte à agréger et d'ajuster par la suite les poids *TF-IDF* selon les liens trouvés. Pour notre exemple d'illustration, nous avons choisi un seuil de similarité α = 0.8. Les termes synonymes repérés sont :

Sim(model,representation)=0.93	*Sim(source, facility)=0.83*
Sim(model,framework)=1	*Sim(proposal, special)=0.83*
Sim(solution, resolution)=1	*sim(property, dimension)=1*
Sim (resolution, result)=1	*Sim(warehouse, storage)=0.96*
Sim(language, construction)=0.81	*Sim(finding, resolution)=0.84*
Sim(language, expression)=0.84	*Sim (resolution, analysis)=0.92*
Sim(computation, experiment)=0.89	

Une fois les liens de synonymie repérés, les poids *TF-IDF* seront ajustés. Par exemple, nous avons :

- *Sim*(model, representation)=0.93 avec *TF-IDF*(model)=0.06 et *TF-IDF*(representation) =0.06. Alors, les deux termes « model » et « representation » sont considérés comme synonymes. Comme ils ont des poids égaux, l'algorithme d'ajustement choisit arbitrairement d'éliminer le terme « representation » et de ne garder que le terme « model ». Ce dernier aura comme poids : *TF-IDF*(model)=0.06 +0.06=0.12
- Puis, nous avons *Sim*(model,framework)=1, avec *TF-IDF*(model)=0.12 et *TF-IDF*(framework)=0.31, « model » et « framework » sont considérés comme synonymes. Ainsi, nous éliminons le terme du moindre poids « model », et nous gardons le terme qui a le poids le plus important « framework » avec *TF-IDF*(framework)=0.31+0.12=0.43.

Après l'ajustement sémantique des poids *TF-IDF*, nous remarquons que la liste des huit termes ayant les poids *TF-IDF* les plus importants change, elle devient :

position	***terme***	***TF-IDF***
1	framework	0.43
2	resolution	0.29
3	dimension	0.19
4	cube	0.18
5	language	0.16
6	computation	0.15
7	datum	0.14
8	source	0.11

Tableau 4.4 : Résultat de la summarization après repérage de liens sémantiques

4.2.4 Choix des *K* termes

Le choix des *k* termes les plus représentatifs constitue la dernière étape dans le processus de summarization. En effet, ce sont ces *k* termes qui vont constituer l'agrégat qui est le résultat attendu du processus de summarization. Afin de choisir ces *k* termes, nous allons d'abord ordonner la liste des termes résultante de l'étape précédente selon le poids décroissant des termes. Une fois la liste des termes ordonnée, les *k* premiers termes de la liste seront choisis comme les *k* termes formant l'agrégat.

Le paramètre *k* est un paramètre qui fixe la taille de l'agrégat. A priori, ce paramètre est choisi par l'utilisateur selon la taille de l'agrégat voulue, néanmoins nous proposons dans ce processus une valeur pour le paramètre *k* qui permette d'avoir une meilleure qualité de l'agrégat. Cette valeur est obtenue grâce à des expérimentations sur un corpus d'évaluation sur la base d'une mesure de qualité que nous allons présenter dans la suite de ce chapitre.

Ainsi, ce processus de summarization s'achève par la constitution de l'agrégat en choisissant les *k* termes ayant les poids de représentativité les plus importants.

- Application sur l'exemple d'illustration :

Posons *k*=7, notre agrégat est formé de sept termes. Ainsi, la summarization du texte nous donne l'agrégat= {framework, resolution, dimension, cube, language, computation, datum}.

4.2.5 Mesure de qualité de la summarization

Le but du processus de summarization proposé est de synthétiser un contenu textuel par ses *k* termes les plus représentatifs, ces *k* termes forment ce qu'on appelle l'agrégat du contenu textuel. Ainsi, plus les termes formant l'agrégat sont pertinents et représentatifs du contenu textuel agrégé plus l'agrégat est considéré comme meilleur.

Dans le but d'évaluer la qualité de la summarization proposée, nous allons consacrer cette section à la proposition d'une mesure pour l'évaluation de la qualité du processus de summarization. Comme nous nous sommes basés sur des techniques de Recherche d'Information, nous avons choisi de nous inspirer des mesures d'évaluation des Systèmes de Recherche d'Information afin de proposer une mesure de qualité pour la summarization.

Notre mesure pour l'évaluation de la qualité de la summarization, que nous appelons « *T_Mesure* » est inspirée de la mesure *E-Mesure* proposée par Rijsbergen [Rijsbergen, 1979]. La *E-Mesure* sert à évaluer les résultats obtenus par un Système de Recherche d' Information pour une requête donnée. Elle se base sur deux mesures fondamentales pour l'évaluation des systèmes de recherche d'informations : le rappel et la précision.

- Rappel : il mesure la proportion des documents pertinents restitués comme résultat à une requête par rapport aux documents pertinents existants dans la collection de documents pour la même requête. Si le rappel est égal à 1 cela veut dire que tous les documents pertinents qui existent dans la collection des documents ont été restitués par le système de recherche d'information. Cette mesure permet aussi de déterminer le silence qui est la proportion de documents pertinents non restitués.

$$Rappel = \frac{D_{r,p}}{D_{r,p} + D_{\neg r,p}} \quad (4.7)$$

- Précision : elle mesure la proportion des documents pertinents restitués par rapport à l'ensemble des documents restitués. La précision vaut 1 quand tous les documents restitués sont pertinents et elle est nulle quand aucun document pertinent n'est restitué. Cette mesure permet également de déterminer la proportion des documents non pertinents restitués appelé bruit.

$$Précision = \frac{D_{r,p}}{D_{r,p} + D_{r,\neg p}} \quad (4.8)$$

- *E-Mesure* :

La précision et le rappel permettent d'évaluer la qualité des résultats restitués par le SRI(Système de Recherche d'Information) comme réponse à une requête mais leur considération l'un séparément de l'autre peut diminuer de leur efficacité. La mesure *E-Mesure* combinent les

deux mesures *précision* et *rappel* afin d'avoir une meilleure évaluation du SRI, elle est calculée comme suit :

$$E - Mesure(\text{b}) = 1 - \frac{1+b^2}{\frac{b^2}{rappel} + \frac{1}{pr\acute{e}cision}} \quad (4.9)$$

Cette mesure est paramétrée par *b*. En effet, elle permet de donner plus d'influence à la précision ou au rappel à travers le paramètre b, plus *b* est grand, plus l'importance donnée au rappel est grande et inversement. Pour un *b*=1, la précision et le rappel ont la même influence sur le calcul de la mesure.

- *T-Mesure* :

En se basant sur la *E-Mesure*, notre mesure de qualité de summarization proposée *T-Mesure* est définie comme suit :

$$T - Mesure(\text{b}) = 1 - \frac{1+b^2}{\frac{b^2}{t-rappel} + \frac{1}{t-pr\acute{e}cision}} \quad (4.10)$$

Où : $$t\text{-}rappel = \frac{nombre\ \ de\ termes\ \ d'agr\acute{e}gat\ \ connus\ \ comme\ \ descripteurs}{nombre\ \ de\ descripeurs\ \ \ du\ texte} \quad (4.11)$$

Et : $$t - pr\acute{e}cision = \frac{nombre\ \ de\ termes\ \ d'agr\acute{e}gat\ \ connus\ \ comme\ \ descripteurs}{nombre\ \ de\ termes\ \ d'agr\acute{e}gat\ (k)} \quad (4.12)$$

En effet, dans un texte à agréger, nous pouvons distinguer les ensembles de termes suivants :

1. Ensemble des termes descripteurs du texte : est un ensemble de termes connus pour être des descripteurs du texte à agréger. Leur nombre peut varier d'un texte à un autre.
2. Ensemble des termes d'agrégat : est l'ensemble des *k* termes qui forment l'agrégat obtenu à l'issue du processus de summarization.

Ainsi, le rappel mesure la proportion des termes, d'agrégat résultat de la summarization, qui sont connus comme descripteurs du texte par rapport aux termes descripteurs du texte. Si le rappel est égal à 1 cela veut dire que tous les termes descripteurs du texte à agréger font partie de l'agrégat obtenu à l'issue de la summarization.

La précision mesure la proportion des termes représentatifs obtenus parmi l'ensemble des termes d'agrégat. Ainsi, la précision nous aide à mesurer la représentativité des termes obtenus par la summarization. La précision vaut 1 quand tous les termes d'agrégat sont représentatifs (connus comme descripteurs), elle est nulle quand aucun terme d'agrégat n'est représentatif.

Par ailleurs, notre summarization proposée est sémantique, elle permet de comptabiliser le contenu sémantique du terme. Ce même principe, est valable dans l'étape de l'évaluation, si deux termes ont une similarité sémantique égale ou dépassant le seuil de similarité α fixé lors de la summarization, ils sont considérés comme identiques. Cela dit, si un terme qui fait partie de l'agrégat a un synonyme qui figure dans la liste de descripteurs, on considère que le terme lui-même figure dans la liste des descripteurs.

D'autre part, et comme nous l'avons cité précédemment, le paramètre *b* de la mesure $T - Mesure(\mathrm{b})$ permet de donner plus d'influence à la précision ou au rappel. Plus *b* est grand, plus l'importance donnée au rappel est grande et inversement.

La valeur de *T-Mesure* va de 0 à 1, plus la valeur de *T-Mesure* est petite plus le résultat de la summarization est meilleur. Par ailleurs, le but du processus de summarization proposé est de sélectionner un ensemble des termes représentatifs du texte qui vont former l'agrégat. Ainsi, plus les termes sélectionnés pour former l'agrégat sont représentatifs, plus le résultat de la summarization est meilleur. Par conséquent, nous allons donner plus d'influence à la précision qu'au rappel. Nous allons retenir la valeur de 0.30 pour le paramètre b afin de donner plus d'importance à la précision qu'au rappel. Enfin nous aurons :

$$T - Mesure(\mathrm{b}) = 1 - \frac{1.09}{\frac{0.09}{t-rappel} + \frac{1}{t-pr\,écision}} \qquad (4.13)$$

D'autre part, la mesure de qualité proposée va nous servir à estimer une valeur du paramètre *k* dite optimale. En effet, nous allons proposer pour chaque opération de summarization une valeur dite optimale pour le paramètre *k*. La valeur optimale du paramètre *k* est la valeur qui permet de maximiser sa qualité en se basant sur la mesure *T-Mesure*. Pour ce faire, nous allons procéder comme suit, nous allons produire des agrégats d'une taille allant de k=1 à k=n, tel que : n < nombre de termes du texte à agréger. Puis, nous calculons pour chaque agrégat la valeur de *T-Mesure* correspondante. Finalement, nous gardons l'agrégat ayant obtenu la meilleure valeur de *T-Mesure*, ainsi que la valeur du paramètre *k* correspondante qui représente la valeur dite optimale de *k*.

- Application sur l'exemple :

L'application de la mesure *T-Mesure* sur notre exemple d'illustration va nous servir d'une part, à évaluer la qualité de l'agrégat obtenu, et d'autre part, à estimer une valeur optimale pour le paramètre *k*. Pour ce faire, nous allons commencer d'abord par évaluer l'agrégat obtenu précédemment dans ce chapitre à l'aide de la mesure *T-Mesure.*

Le texte de notre exemple d'illustration est décrit par huit descripteurs, qui sont :{model, cube, multidimensional, warehouse, text, data, analysis, method}. L'agrégat obtenu par le processus de summarization est composé de sept termes (*k*=7), qui sont : {framework, resolution, dimension, cube, language, computation, datum}.

Ainsi, nous allons évaluer la qualité de l'agrégat par le calcul de la mesure *T-Mesure*, nous avons :

$$t\text{-}rappel = \frac{nombre\ \ de\ termes\ \ d'agr\acute{e}gat\ \ connus\ \ comme\ \ descripteurs}{nombre\ \ de\ descripeurs\ \ \ du\ texte} = 0.50$$

Pour le dénominateur, nous avons huit descripteurs du texte, et pour le numérateur, nous avons quatre termes de l'agrégat connus comme descripteurs, qui sont :{framework, cube, resolution, datum}. Notons que le terme « framework » est considéré comme synonyme du terme « model » car nous avons : *Sim*(model, framework)=1. De même, nous avons « datum » synonyme de « data », et « resolution » synonyme de « analysis ».

$$t - pr\acute{e}cision = \frac{nombre\ \ de\ termes\ \ d'agr\acute{e}gat\ \ connus\ \ comme\ \ descripteurs}{nombre\ \ de\ termes\ \ d'agr\acute{e}gat\,(k)} = 0.57$$

Pour le dénominateur, nous avons sept termes qui forment l'agrégat. Pour le numérateur, nous avons quatre termes de l'agrégat connus comme descripteurs.
Ainsi :

$$T - Mesure = 1 - \frac{1.09}{\frac{0.09}{t-rappel} + \frac{1}{t-pr\acute{e}cision}} = 0.43$$

D'autre part, nous allons essayer d'estimer une valeur optimale pour le paramètre *k*, celle qui correspond à une meilleure qualité de l'agrégat obtenu. Pour cela, nous allons construire les agrégats correspondants à k=1, 2 ,3 , 4,...,30 tout en calculant la valeur de *T-Mesure* correspondante à chaque agrégat. Comme *T-Mesure* est inversement proportionnelle à la qualité de summarization, nous introduisons la mesure *QS*, avec *QS*=1-*T-Mesure*. *QS* est proportionnelle à la qualité de summarization, plus la valeur de QS est importante, plus la qualité

de la summarization est meilleure. Les résultats obtenus sont représentés dans le tableau suivant, les lignes de tableau grisées correspondent à la restitution des termes reconnus comme descripteurs :

k	*T-Mesure*	*QS*	*K*	*T-Mesure*	*QS*
1	0.37	0,63	16	0,54	0,46
2	0,20	0,80	17	0,57	0,43
3	0,41	0,59	18	0,59	0,41
4	0,31	0,69	19	0,61	0,39
5	0,42	0,57	20	0,63	0,37
6	0,51	0,49	21	0,65	0,35
7	0,43	0,56	22	0,66	0,34
8	0,37	0,63	23	0.68	0.32
9	0,44	0,56	24	0,69	0,31
10	0,38	0,61	25	0,66	0,34
11	0,44	0,56	26	0,67	0,32
12	0,49	0,51	27	0,68	0,32
13	0,52	0,48	28	0,70	0,30
14	0,56	0,44	29	0,71	0,39
15	0.58	0.42	30	0,72	0,28

Tableau 4.5 : Les résultats obtenus pour T-Mesure et QS.

De ces résultats, nous remarquons que les trois meilleurs agrégats obtenus sont :

1. Agrégat (k=2)= {framework, resolution}
2. Agrégat (k=4)= {framework, resolution, dimension, cube}
3. Agrégat (k=8)= {framework, resolution, dimension, cube, language, computation, datum, warehouse}

4.3 Algorithme de summarization

Dans la section précédente, nous avons détaillé les différentes parties de notre processus de summarization. Dans cette section, nous allons essayer de décrire formellement ce processus de summarization en présentant les pseudo-algorithmes décrivant chaque étape du processus afin de mieux l'illustrer.

Notons que le texte à agréger est un contenu textuel de *n* termes tel que Text= $\{t_1, t_2, t_3,...,t_n\}$. Ces termes sont organisés selon un ensemble de paragraphes, Text= p_1U p_2U…..Up_p où chaque paragraphe p_i est composé d'un ensemble de termes.

Comme nous l'avons vu précédemment, notre processus de summarization consiste en quatre opérations élémentaires qui sont :

1. Epurations des données
2. Pondération des termes
3. Repérage de liens sémantiques
4. Choix des K termes

4.3.1 Epuration des données

La première étape de notre processus de summarization est l'étape d'épuration de données, elle consiste en quatre sous étapes : le découpage du texte en paragraphes, la tokenisation, la suppression de mots vides et la lemmatisation.

4.3.1.1 Découpage du texte en paragraphes

La première étape d'épuration de données consiste à découper le texte à agréger en paragraphes. Nous considérons que le paragraphe est une chaine de caractère délimitée par deux alinéas [Bessonnat, 1988]. Dans notre travail, l'alinéa est considéré comme un signe typographique jouant le rôle de délimiteur de paragraphes, il correspond à un retour à la ligne suivi d'une indentation (un retrait de la première ligne d'un paragraphe). Ainsi, pour découper le texte Text en paragraphe nous procédons comme suit :

Algorithme 1 : Découpage du Text en paragraphes

Entrée Text : un texte à découper

Sortie $p_1, p_2, \ldots\ldots, p_m$: Les paragraphes formant Text

Variables Text [], p_i[] : une chaine de caractères sous forme d'un tableau

Alinéa : délimiteur de paragraphes, Alinéa=retour en ligne+ indentation

FinText : caractère signalant la fin du Text

Début

Tant que (Text[j] <> FinText*) faire*

{ i++ ;

Tant que ((Text[j] <> FinText*) et (Text[j]* <>*Alinéa))*

{ Insérer (p_i , Text[j]) ;

J++;

} Fin Tant que

Si (Text[j] <> FinText *) alors J++ ;*
} Fin Tant que
} ***Fin***

A la fin de cette opération de découpage, nous aurons la liste de paragraphes qui compose le texte à agréger, Text= p_1U p_2U…..Up_p avec p est le nombre de paragraphes que contient Text.

4.3.1.2 Tokenisation

La deuxième étape de l'épuration de données est la tokenisation. Il s'agit dans cette étape de segmenter le texte Text en termes (tokens) afin de reconnaitre les différents termes qui composent chaque paragraphe p_i du texte. En effet, nous allons transformer chaque paragraphe d'une liste de caractères à une liste de termes (tokens). La tokenisation se base sur la notion d'expressions rationnelles, elle consiste à comparer le texte à chaque expression rationnelle en allant le plus loin possible, une fois la comparaison s'arrête, on récupère le token et on recommence la comparaison. La tokenisation est réalisée comme suit :

Algorithme 2 : Tokenisation du texte

Entrées *: une liste des p paragraphes* p_i *, avec* p_i *une chaine de caractères*
List_Exp : tableau de n expressions rationnelles définissant les tokens

Sortie *: liste de p paragraphes avec tokens* $pt_i[]$ *où* $pt_i[]$ *est un tableau de tokens*

Variables : *i, j, z*

Début

{ Tant que (i<p) Faire

{ Pour j=1 jusqu'à taille (List_Exp[]) faire :

Tant que (p_i *<> null) Faire*

{ Pos=Comparer(p_i *,* $List\_Exp[j]$*)//on récupère la position d'arrêt*

$pt_i[t]$*=*$Extraire(p_i$ *, pos)// récupération et effacement du token à partir du* p_i *;*

t++ ;}Fin Tant que

} Fin Pour

} Fin Tant que

} Fin

4.3.1.3 Suppression de termes vides

Il s'agit de supprimer tous les mots vides de chaque paragraphe du texte Text. Cette étape est réalisée en utilisant la liste de mots vides List_Fox présentée précédemment. Pour ce faire, nous allons comparer, chaque terme de chacun des *p* paragraphes du TEXT avec tous les termes de la liste List_Fox, si on rencontre un des termes de List_Fox dans un des paragraphes de Text, on supprime le terme du paragraphe. Cette procédure est présentée dans l'algorithme 3 ci-dessous.

Algorithme 3 : Suppression de termes vides

Entrée *une liste des p paragraphes* $pt_i[]$ *avec* $pt_i[]$*: un tableau de chaines de caractères*
List_fox : tableau de l termes vides

Sortie *la liste des p paragraphes* $pt_i[]$

Variables *i , j , k , t : entiers*

Début
{ *Pour i=1 jusqu'à p faire :*

{ t=taille ($pt_i[]$*)*
Pour j=1 jusqu'à t faire :
{ Pour k=1 jusqu'à l faire :
{ Si $pt_i[j] = List\_fox[k]$ *Alors supprimer (*$pt_i[j]$*) Fin Si*
} Fin pour
} Fin pour
} Fin pour
} Fin

4.3.2 Pondération

Cette étape sert à identifier l'importance de chaque terme de chacun des paragraphes du Text en lui affectant un poids qui quantifie cette importance. Pour cela, nous allons utiliser la formule de pondération *TF-IDF*. Nous allons calculer pour chaque terme de chacun des paragraphes de Text sa fréquence locale *tf* et sa fréquence globale *idf* selon les formules citées dans la section 1.4.2.2 de ce chapitre, elles sont calculées comme suit :

Algorithme 4 : Calcul de la fréquence locale *tf*

Entrée *$List_i$: Liste distincte des termes du paragraphe pt_i*
P : nombre de paragraphes dans Text
$Nbr_i[j]$: Nombre d'occurrence du terme $pt_i[j]$ dans le paragraphe pt_i

Sortie *tf_i : Liste des fréquences locales des termes du paragraphe pt_i*
Variables *i, j, t:entiers*
Début
{ Pour i=1 jusqu'à p faire :

{ t=taille ($List_i$)
Pour j=1 jusqu'à t faire :
{ $tf_i[j] = 0.5 + 0.5 * \frac{Nbr_i[j]}{Max\ Nbr_i}$
}}} ***Fin***

Algorithme 5 : Calcul de la fréquence globale *idf*

Entrée *$List_i$: Liste distincte des termes du paragraphe pt_i*
P : nombre de paragraphes dans Text
$Nbr\_pt_i[j]$: Nombre de paragraphe où le terme $pt_i[j]$ apparait

Sortie *idf_i : liste des fréquences globales des termes du paragraphe pt_i*
Variables *i, j, t:entiers*
Début
{ *Pour i=1 jusqu'à p faire :*

{ t=taille ($List_i$)
Pour j=1 jusqu'à t faire :
{ $idf_i[j]$=log(p/$Nbr\_p_i[j]$)
*}}}**Fin***

Avec le calcul de la fréquence globale *idf*, l'étape de pondération prend fin, il ne reste maintenant que le calcul du poids final *TF-IDF* pour chaque terme. *TF-IDF* est le produit de la fréquence locale *tf* et la fréquence globale *idf*.

A l'issue de l'étape de pondération, nous obtenons une liste de termes avec leurs poids *TF-IDF* respectifs pour chaque paragraphe du texte. A partir d'ici, nous n'avons plus besoin de la structuration du texte en paragraphes ; par conséquent, nous allons fusionner toutes les listes de termes de paragraphes en une seule liste *List_Terms* en conservant le poids de chaque terme dans une liste appelé List_Scor. Evidemment, si un terme existe dans plusieurs listes (dans

plusieurs paragraphes), il ne figure qu'une seule fois dans la liste finale tout en cumulant ses différents poids.

4.3.3 Repérage de liens sémantiques

Afin de prendre en considération la sémantique du texte à agréger, nous allons essayer d'identifier les liens sémantiques qui existent entre les termes du texte et d'ajuster les poids de représentativité associés aux termes selon ces liens. L'identification des liens sémantiques s'effectue en utilisant une fonction de calcul de similarité Sim() basée sur la mesure de Lin et un seuil α de similarité. L'ajustement des poids de termes selon les liens sémantiques qui existent entre eux se fait comme suit :

Algorithme 7 : Ajustement sémantique des poids *TF-IDF*

Entrée *List_terms : liste de termes retenus*
α : seuil de similarité
List_scor : liste des poids des termes

Sortie *List_scor : liste des poids des termes*
Variables *i, j, t:entiers*
Début
{ *Size=taille(List_terms)*
Pour i=1 jusqu'à Size-1 Faire
{ Pour j=i+1 jusqu'à Size Faire
{ Si Sim(List[i],List[j])>α // Sim() est une fonction de calcul de similarité sémantique
Alors { t_1 = Sup(List[i],List[j])
t_2 = Inf(List[i],List[j])
Eliminer(List_terms,t_2)
List_scor [t_1]= List_scor [t_1]+List_scor [t_2]
Eliminer(list_scor,t_2)
*} } }**}***

A la fin de cette étape, nous aurons deux listes : *List_terms* pour les termes retenus et *List_scor* pour les scores des termes.

4.3.4 Choix des *K* termes

Finalement, cette étape consiste à choisir les *k* termes qui forment l'agrégat du Text. Il s'agit d'ordonner la liste des termes *List_terms* selon les scores de la liste *List_scor* en utilisant la fonction *Trier()* et de choisir par la suite les *k* premiers termes de la liste *List_terms.*

Algorithme 8 : Choix des *K* termes

Entrée *List_terms : liste de termes retenus*
List_scor : liste des poids des termes

Sortie *List_agregat : liste des termes qui forment l'agrégat*

Variables
List_tri : Liste de termes triée selon la liste List_scor
K : taille de l'agrégat
Debut
{ *List_tri=Trier(List,List_scor)*
Pour i=1 jusqu'a K faire
{List_agregat [i]= List_tri[i]
}}
Fin

A la fin de cette étape, nous avons la liste *List_agregat* qui héberge l'agrégat du texte. Ainsi, notre processus de summarization s'achève et notre texte de départ Text est agrégé en la liste *List_agregat.*

4.4 Conclusion

La summarization de données textuelles revient à résumer ces données par le contenu le plus pertinent. Notre processus de summarization proposé consiste à agréger un texte par ses termes les plus représentatifs. Le choix des termes pertinents formant l'agrégat se fait sur la base d'un poids calculé grâce à une forme adaptée de la formule de pondération *TF-IDF*. De plus, notre summarization est sémantique, puisque nous ajustons les poids obtenus par *TF-IDF* selon les liens de synonymies existants entre termes à l'aide du thésaurus WordNet2.1. La taille de l'agrégat, résultat de la summarization est variable, elle est fixée par l'utilisateur selon son besoin. Aussi, une mesure d'évaluation de la qualité de summarization *T-Mesure* est proposée. Cette mesure nous a également permis de calculer les valeurs de *k* pour lesquelles la qualité de summarization est meilleure.

Le processus de summarization proposé est assez flexible, il s'applique sur un large éventail de textes de thématiques, de volumes et de formes différents ; aucune structure donnée n'est exigée. A travers la lemmatisation, et l'identification des catégories grammaticales, notre processus de summarization permet d'avoir un agrégat composé juste de noms, d'adjectifs ou de toute autre catégorie grammaticale souhaitée. De même, l'ajustement sémantique est réglable par l'utilisateur selon le niveau de similarité toléré entre termes.

Néanmoins, ce processus s'applique particulièrement sur des textes où nous disposons d'une certaine cohérence thématique dans l'écriture du texte. Avoir une organisation structurelle basée sur le paragraphe comme unité d'écriture est aussi important. Également, une certaine rigueur vis à vis de l'aspect orthographique est aussi nécessaire, notre processus n'est pas capable de reconnaitre les termes faussement écrits ou de corriger des fautes d'orthographe. Dans le chapitre suivant, nous allons détailler comment ce processus va permettre la summarization de données textuelles au sein de l'entrepôt de données.

CHAPITRE 5
SUMMARIZATION AU SEIN DE L'ENTREPOT DE DONNEES

1. Introduction

L'objectif principal de ce chapitre est de créer un environnement multidimensionnel adapté à l'agrégation de données textuelles à travers de nouveaux opérateurs OLAP. En effet, l'utilisation des entrepôts de données classiques fait massivement intervenir des opérateurs d'agrégation pour synthétiser les données. Ces fonctions d'agrégation sont très efficaces dans l'environnement OLAP classique pour synthétiser des valeurs numériques, mais restent inefficaces devant les autres types de données, notamment les données textuelles. En effet, l'environnement OLAP ne fournit pas de fonctions d'agrégation adaptées aux données textuelles.

Dans ce chapitre, nous allons essayer de résoudre l'incompatibilité de l'environnement OLAP avec les données textuelles par la proposition de deux nouveaux opérateurs pour la summarization du texte: *Term_up* et *Term_down* basés sur un modèle multidimensionnel capable de prendre en charge les données textuelles. Les deux opérateurs proposés sont construits sur la base du processus de summarization proposé dans le chapitre précédent. Afin de mieux présenter le modèle multidimensionnel et les opérateurs, un entrepôt d'articles de presse va servir comme cas d'étude.

Ainsi, ce chapitre est organisé de la façon suivante. Tout d'abord, et avant d'aborder les entrepôts textuels et la summarization, nous allons rappeler quelques généralités sur les entrepôts de données classiques et leurs caractéristiques, un état de l'art peut être trouvé dans [Inmon, 1992]. Ensuite, nous allons évoquer les principales difficultés à intégrer et analyser des données textuelles dans un environnement OLAP. Un entrepôt d'articles de presse est présenté comme cas d'étude. Finalement, nous allons présenter le modèle de données multidimensionnel adapté aux données textuelles ainsi que les deux opérateurs de summarization *Term_up* et *Term_down*.

5.2 Entrepôt de données classiques

Un entrepôt de données est un système pour le stockage de données qui a pour objectif final l'analyse des données en vue de la prise de décision. En 1992, Bill définit un entrepôt de données comme étant une «collection de données orientées sujet, intégrées, non volatiles et

historisées, organisées pour le support du processus d'aide à la décision » [Inmon, 1992]. Les données sont «orientées sujet» dans la mesure où elles sont organisées par thème, elles sont aussi dites « intégrées » du faite qu'elles proviennent de sources différentes (bases de données transactionnelles, fichiers, .etc.). En outre, les données sont « historisées » afin de rendre possible l'analyse des données par rapport à un référentiel temporel associé. De plus, les données sont dites « non volatiles», cela signifie que les données stockées au sein de l'entrepôt de données ne peuvent pas être supprimées. Cela doit permettre de conserver la traçabilité des décisions prises. Enfin, les données sont stockées selon une organisation multidimensionnelle propice à l'analyse.

L'organisation multidimensionnelle des données au sein d'un entrepôt de données est faite selon les besoins d'analyse. Le cube ou l'hyper-cube est une représentation primitive qui est très répandue car elle permet une expression graphique meilleure du besoin d'analyse. Le cube permet de représenter les données d'un entrepôt sous la forme de points dans un espace à plusieurs dimensions. Néanmoins, la modélisation en cube est très limitée en termes de représentation des dimensions, pour concevoir des entrepôts de données plus élaborés, une nouvelle forme de modélisation multidimensionnelle a été définie.

Cette modélisation se base sur deux concepts fondamentaux : le concept de fait et le concept de dimension. Un fait représente un sujet d'analyse, caractérisé par une ou plusieurs mesures, qui ne sont autres que des indicateurs décrivant le sujet d'analyse. Ce fait est analysé selon des axes d'observation ayant aussi à leur tour des descripteurs. Ces axes d'observation sont appelés dimensions. Ces dimensions peuvent présenter des hiérarchies qui offrent la possibilité de réaliser des analyses à différents niveaux de granularité (niveaux de détail).

Ces concepts de base ont permis de définir trois types de schémas classiques du modèle de données pour l'entrepôt de données : schéma en étoile, schéma en flocon de neige, et schéma en constellation. Le schéma en étoile se compose d'une table de faits centrale et d'un ensemble de tables de dimensions. Le schéma en flocon de neige est semblable à un schéma en étoile sauf que ses dimensions sont normalisées, faisant ainsi apparaitre des hiérarchies de dimension. La normalisation permet un gain d'espace de stockage en évitant la redondance de données, mais engendre une dégradation des performances, dans la mesure où elle multiplie le nombre de jointures à effectuer pour l'analyse et rallonge le temps de réponse aux requêtes d'analyse. Finalement, le schéma en constellation, aussi appelé flocon de faits, est un schéma qui fait coexister plusieurs tables de faits qui peuvent partager ou non des dimensions communes.

L'entrepôt de données a pour objectif final l'analyse des données en vue de la prise de décision. Différents types d'analyses peuvent être réalisés, notamment l'analyse en ligne OLAP (Online Analytical Processing). Il s'agit d'une navigation exploratrice des données dont le but est d'arriver au cours de la navigation à détecter des points intéressants qu'on essaye de décrire, d'expliquer en naviguant, par exemple en allant chercher davantage de détails où par contre en agrégeant les données. Le rôle de l'utilisateur est central ici puisque c'est lui qui réalise la navigation, il doit connaitre le domaine afin d'être en mesure de savoir si les valeurs des mesures sont intéressantes ou non.

Afin de réaliser la navigation, différents operateurs s'appliquent au niveau d'un entrepôt de données. Chaque opérateur prend en entrée un cube OLAP et fournit un autre cube en sortie. Un cube OLAP représente dans ses cellules des faits à analyser (matérialisés par des mesures numériques) en fonction des dimensions (axes d'analyse) décrites par des attributs susceptibles d'être hiérarchisées (par exemple, une dimension géographique : ville, région, pays). Globalement, nous pouvons classer les opérateurs OLAP en trois catégories [ESPINASSE, 2010] :

1. *Les opérateurs classiques* : la sélection (slice et dice), la projection, la jointure.etc.

2. *Les opérateurs agissant sur la structure* : la rotation, la permutation, la division, l'emboitement, .etc.

3. *Les opérateurs agissant sur la granularité* : Le forage vers le haut (*Roll-up*) et le forage vers le bas (*Drill-down*). Les opérations agissant sur la granularité ou ce qu'on appelle communément les opérations d'agrégation permettent de naviguer au travers des hiérarchies et ainsi agréger les données en fonction des requêtes utilisateurs à travers les deux opérateurs *Roll-Up* et *Drill-Down*. En effet, l'opérateur *Roll-Up* permet d'agréger les données de l'entrepôt vers un niveau supérieur de granularité. A l'inverse, l'opérateur *Drill-down* permet de représenter les données à un niveau de granularité inferieur donc sous une forme plus détaillée.

5.3 La summarization au sein d'un entrepôt de données

Les entrepôts de données aident les entreprises à exploiter le volume d'information croissant qui transitent au sein de leur système d'information. En effet, L'essor des technologies de l'information et l'avènement d'Internet et des réseaux, ont accru le volume des informations disponibles de manière considérable. L'exploitation de ces informations est d'une grande importance pour le processus de prise de décision. En revanche, les entrepôts de données et les

technologies OLAP permettent un traitement synthétique de l'information pour faciliter les prises de décisions grâce à des méthodes et des outils puissants qui permettant l'analyse des informations obtenues à partir de bases de données transactionnelles [Sullivan, 2001].

En effet, l'analyse des données multidimensionnelles à travers des mesures numériques est une tâche bien maitrisée actuellement [Sullivan, 2001]. Les technologies OLAP représentent un bon exemple, elles nous offrent des opérateurs fiables et robustes pour l'analyse au sein d'un entrepôt de données. Entre autre, les opérateurs d'agrégation : *Roll-Up* et *Drill-Down* nous permettent de naviguer au travers les hiérarchies et de synthétiser les données numériques en fonction des requêtes utilisateurs.

L'agrégation au sein d'un entrepôt de données consiste à représenter les données à un niveau de granularité supérieur conformément à la hiérarchie définie sur la dimension. Une fonction d'agrégation (somme, moyenne,…) en paramètre de l'opération indique comment sont calculées les valeurs du niveau supérieur à partir de celles du niveau inférieur. Les fonctions d'agrégation sont pour la majorité des fonctions arithmétiques (somme, moyenne,..) compatibles aux données numériques mais non applicables aux données textuelles. Ainsi, l'agrégation au sein de l'entrepôt de données est limitée aux données numériques, les données textuelles ne sont ni exploitables ni prises en charge par ces entrepôts.

D'après les travaux de Tseng et Chou, les données numériques exploitées par l'entrepôt de données ne représentent que 20% des données du système d'information de l'entreprise [Tseng et Chou, 2006]. Les 80% restantes, généralement contenues dans des documents électroniques, et constituées principalement par des données textuelles restent hors de portée de l'entrepôt de données ce qui va priver l'utilisateur d'une quantité d'information assez importante.

Pour que la donnée textuelle soit exploitable, notamment agrégée au niveau de l'entrepôt de données, il faut absolument étendre les approches classiques d'entrepôts de données et les opérateurs d'agrégation OLAP adaptés jusqu'à présent qu'aux données numériques. Deux problématiques ressortent de l'adaptation des entrepôts de données aux données textuelles :

- l'intégration des données textuelles : qui nécessite une représentation conceptuelle multidimensionnelle adaptée aux données textuelles.
- l'agrégation de données textuelles : qui nécessite la mise en place des opérateurs OLAP d'agrégation adaptés aux données textuelles

Dans la suite de ce chapitre, nous allons essayer d'apporter des solutions à ces deux problématiques.

5.4 Cas d'étude : Entrepôt d'articles de presse

5.4.1 Description de l'entrepôt d'articles de presse

Les articles de presse constituent une source illimitée et très diverse d'informations périodiques. Ils couvrent un large éventail de faits et d'événements dans différents domaines (politique, économique, sanitaire, artistique, etc.) et sur des périodes qui peuvent être très longues.

Dans un entrepôt d'articles de presse classique, les requêtes d'analyse possibles sont simples et se limitent aux capacités des fonctions d'analyse OLAP disponibles. Ainsi, l'analyse OLAP classique des articles de presse se limite à un comptage ou un listing d'instances à travers les deux fonctions : *Count* et *List* [Tseng et Chou, 2006]. Par exemple, dans la figure 5.1, nous avons la représentation tabulaire d'une requête type d'un utilisateur sur un entrepôt d'articles de presse. Dans cet exemple, l'utilisateur désire connaitre le degré de l'intérêt porté sur la crise économique qu'a connue la majorité des pays industrialisés à partir de 2008 à travers les journaux quotidiens suivants : « the Guardian », « the New York times », et « ChainaDaily ». L'entrepôt d'articles de presse nous permet de connaitre le nombre d'articles qui portent sur la crise économique de chacun des trois journaux grâce à la fonction de comptage d'instance COUNT. Par exemple, le journal américain « The New York Times» est celui qui a publié le plus d'articles sur l'économie en 2010 avec un score de 523 articles.

Count (comptage d'articles)	Type publication : journal Domaine : **économique**		
publication	**The Guardian**	**The New York Times**	**Chinadaily**
Année			
2009	256	489	159
2010	489	523	296
2011	475	369	352

Figure 5.1 : Exemple de requête-type sur un entrepôt d'articles de presse représentée par une table multidimensionnelle.

Par ailleurs, d'autres requêtes utilisateurs sur cet entrepôt relativement au sujet de la crise économique de 2009 peuvent être soumises. Par exemple, connaitre les différents faits et événements qui ont accompagnés la crise économique de 2009 en Grande Bretagne. Cela

revient à analyser le contenu textuel des 489 articles publiés par « The Guardian » ce qui n'est pas possible avec les entrepôts de données classiques. En effet, ces derniers manquent d'opérateurs et de fonctions adéquats pour le traitement du contenu textuel.

Traiter le contenu textuel au sein d'un entrepôt de données va nous permettre d'analyser le contexte et les circonstances des faits observés. Par exemple, dans la figure 5.1, nous pouvons analyser les différentes circonstances qui ont accompagnés la crise économique de 2009. Le décideur peut se poser des questions telles que : comment était l'activité de la bourse pendant l'année 2009 ? Quel était le taux d'inflation dans tel pays ? Pour tel mois ?...etc. Toutes ces questions peuvent trouver des réponses si nous disposons d'un entrepôt de données textuel capable de manipuler notamment d'agréger du contenu textuel.

5.4.2 Modèle conceptuel de données pour l'entrepôt d'articles de presse

Le modèle conceptuel de données représente le cœur d'un entrepôt de données. Pour l'entrepôt d'article de presse, nous avons besoin d'un modèle qui permet de modéliser les besoins d'analyse et de stocker et de manipuler par la suite le contenu textuel de l'article de presse. Le modèle conceptuel de données adopté pour l'entrepôt d'articles de presse est basé sur un schéma en étoile adapté aux données textuelles (Figure 5.2). En effet, nous avons utilisé une mesure textuelle qui permet d'abriter des données textuelles.

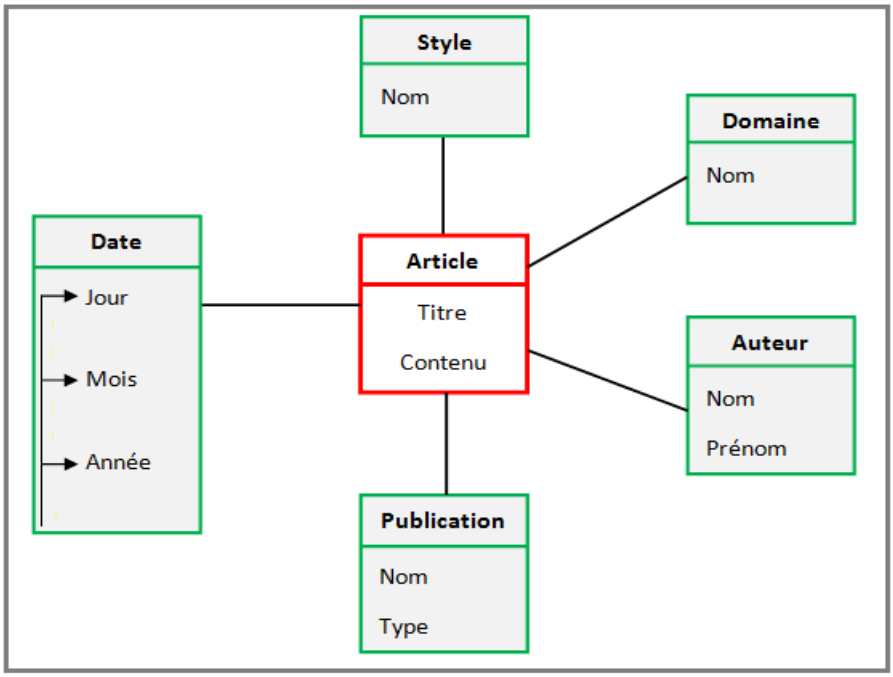

Figure 5.2 : Modèle de données pour l'entrepôt d'articles de presse

Comme nous l'avons mentionné précédemment dans ce chapitre, dans le modèle de données en étoile, le fait modélise un sujet d'analyse el les dimensions modélisent les axes d'analyse. Le sujet d'analyse *Article* nous permet de traiter le contenu de l'article, il dispose de deux mesures : *titre* et *contenu.* Chaque article de presse possède un titre, un ensemble de termes qui exprime l'idée générale du sujet traité par l'article, le reste du contenu textuel de l'article forme la mesure *contenu.* En effet, *Titre* représente l'idée générale de l'article du point de vue de l'auteur et *Contenu* représente l'ensemble des détails qu'a apporté l'auteur sur le sujet abordé, selon le type d'analyse voulue : détaillée ou non, nous choisissons la mesure *Titre* ou *Contenu*.

En outre, nous avons cinq dimensions (axes d'analyse) qui vont nous aider à analyser le fait *Article,* à savoir :

1. *Auteur* : Le nom et le prénom de l'auteur.
2. *Domaine* : le domaine que couvre l'article, ça peut être : la politique, l'économie, l'art, le sport, l'éducation...etc.
3. *Publication* : le nom de la publication à partir de laquelle a été tiré cet article, ainsi que son type : journal, hebdomadaire, magazine...etc.
4. *Date* : la date de la publication de l'article, elle comporte : le jour, le mois, et l'année.
5. *Style* : le style de l'article, l'article peut être : informatif, interview, humoristique, reportage, etc.

5.5 Formalisation de l'entrepôt de données textuel

5.5.1 Modèle de donnée

La modélisation conceptuelle multidimensionnelle vise à représenter les besoins utilisateurs en termes d'analyse. Cette modélisation, indépendante de toute contrainte d'implantation logique et physique facilite la compréhension des données mises à disposition du décideur [Golfarelli et al, 2002]. La modélisation que nous allons proposée est flexible, elle n'est pas restrictive par rapport aux formes que peut prendre un contenu textuel. Elle prend en charge une variété de documents textuels : article (presse, scientifique, …), document, ouvrage, livre…etc.

Néanmoins, cette modélisation est faite en prenant en considération trois caractéristiques importantes :

- La structure : les données textuelles sont organisées généralement de façon hiérarchique, par exemple (des chapitres contenant des sections, elles-mêmes composées de paragraphes). Le contenu textuel que nous traitons se base sur le paragraphe comme unité textuelle.
- Les métadonnées : notre modélisation repose aussi sur la représentation de certaines métadonnées liées au contenu textuel tel que : date de publication, auteur, nom de publication, domaine, etc.
- Analyse OLAP du texte : traditionnellement, l'analyse OLAP du texte ne permet que le comptage d'instances ou le listing. Notre modélisation doit permettre d'effectuer des opérations OLAP notamment l'agrégation sur le contenu textuel.

Notre but est de permettre la summarization du texte au sein d'un environnement OLAP. Ainsi, nous proposons de représenter ce besoin par un modèle permettant l'analyse du contenu textuel, où le fait à analyser serait ce contenu. Pour ce faire, nous avons besoin de mesures adaptées aux données textuelles, nous pouvons distinguer deux types de mesures : les mesures numériques et les mesures textuelles.

Une mesure numérique est exclusivement composée de données numériques. Elle est soit : additive, semi additives ou non additive. Une mesure textuelle est une mesure dont les données textuelles sont à la fois non numériques et non additives. Le contenu d'une mesure textuelle peut être une phrase, un fragment de texte, ou tout simplement un ensemble de termes.

En adoptant le principe de mesure textuelle pour représenter le contenu textuel à agréger, nous proposons un modèle de données basé sur le schéma de données en constellation. Dans un schéma en constellation, nous avons plusieurs tables de faits qui coexistent et qui peuvent partager ou non des dimensions communes [Kimball, 1996]. Notons qu'un schéma en étoile est un schéma en constellation avec un seul fait. Dans le modèle proposé, le contenu textuel est modélisé en tant que sujet d'analyse (table de fait), d'où le besoin d'une mesure textuelle.

En adoptant la formalisation du modèle de données proposée par Teste, notre modèle de données en constellation *C* est défini par [Teste, 2009]:

$C = (F^c, D^c, Star^c)$ où :

- $F^c = \{F_1, F_2, \dots, F_n\}$ un ensemble de faits ;
- $D^c = \{D_1, D_2, \dots, D_m\}$ un ensemble de dimensions ;

- $Star^c = F^c \rightarrow P(D^c)$ est une fonction liant chaque fait à ses dimensions associées, P(D^c) est l'ensemble des sous-ensembles de D^c.

Un fait est défini par :

$F = (N^F, M^F)$ où :

- N^F est le nom du fait
- $M^F = \{M_1, \dots, M_t\}$ un ensemble de mesures

Une mesure est définie par :

$M = (m, f_{agg})$ avec :

- *m* est la mesure
- $f_{agg} = \{f_1, \dots, f_n\}$ est un ensemble de fonctions d'agrégation compatibles avec la mesure.

Une dimension est définie par :

D= (N^D, A^D, H^D) où :

- N^D est le nom de la dimension
- $A^D = \{A_1, A_2, \dots, A_n\}$ est l'ensemble des attributs
- $H^D = \{H_1, H_2, \dots, H_m\}$ est l'ensemble des hiérarchies qui ordonnent les attributs de la dimension.

Une hiérarchie est définie par :

H= (N^H,P^H,$Weak^H$) où :

- N^H est le nom de la hiérarchie
- $P^H = \{P_1, P_2, \dots, P_n, All\}$ est un ensemble ordonné d'attributs appelés paramètres

Avec : $\forall\, k \in (1, \dots, n)$, $P_k \in A^D$ où *n* est le nombre de paramètres, nous avons : P_k est une racine commune(le paramètre de moindre granularité) pour les paramètres tel que : $P_1 = A_1$, $\forall\, H \in H^D$

- $Weak^H : P^H \rightarrow$ P(A^D-P^H) une fonction associant le reste des attributs aux paramètres, avec P(A^D-P^H) est l'ensemble des sous-ensembles de (A^D-P^H)

Toutes les hiérarchies d'une dimension commencent par le même paramètre racine et se terminent par le paramètre de plus haute granularité.

- Modèle de données pour l'entrepôt de presse :

Formellement, le modèle de données en constellation présenté dans la figure 5.2 est le suivant :

CP= (F^{cp}, D^{cp}, $Star^{cp}$) où *:*

- F^{cp} *= {Article}*
- D^{cp} *= {Auteur, Publication, Date, Style, Domaine}*
- $Star^{cp}$ *= {Article* $\longrightarrow$ *{Auteur, Publication, Date, Style, Domaine}}*

Le fait *Article* est défini par : $Article = (Article, \{Titre, Contenu\})$

Les deux mesures *Titre* et *Contenu* sont définies par :

- *Titre=(Titre,{ Term_Up,Term_Down })*
- *Contenu=(Contenu,{ Term_Up,Term_Down })*

Les dimensions sont définies comme suit :

- *Auteur = (auteur, {nom, prénom})*
- *Publication = (publication, {nom, type}*
- *Style = {Style, {nom}}*
- *Domaine = {Domaine, {nom}}*
- *Date =* (N^{Date}, A^{Date}, H^{Date})

Avec: N^{Date} *= Date*

A^{Date} *= {jour, mois, année}*

$H^{Date} = \{H\}$ avec H= (N^H,P^H,$Weak^H$) où

N^H *: Hiérarchie de la date*

P^H *= {jour, mois, année}* et $Weak^H$*={}*

5.6 Déroulement du processus de summarization

La summarization au sein de l'entrepôt textuel comporte deux étapes essentielles, l'alimentation des mesures, et l'agrégation sémantique à l'aide des deux opérateurs : *Term_Up* et *Term_Down* (Fig 5.3).

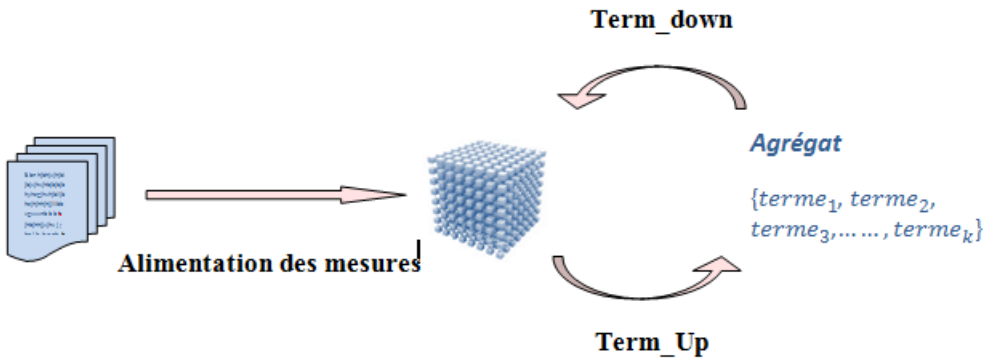

Figure 5.3 : Déroulement du processus de summarization

5.6.1 Alimentation des mesures

Cette étape consiste à alimenter l'entrepôt d'articles de presse par les données textuelles qui ne sont autre que le contenu des articles de presse organisé selon les deux mesures textuelles : *Titre* et *Contenu*. En effet, nous avons précisé dans la section précédente la localisation du contenu des deux mesures : *Titre* et *Contenu* par rapport à l'article de presse. Néanmoins, il ne s'agit pas d'un copiage fidèle du contenu, les mesures *résumé* et *contenu*, ne contiendront pas le contenu textuel intégral correspondant dans l'article de presse mais un ensemble de termes choisis à partir de ce contenu selon leurs degrés d'importance.

Ainsi, nous allons identifier le contenu textuel correspondant à chacune des deux mesures par rapport au contenu de l'article. Par la suite, nous allons appliquer un ensemble de techniques issues du domaine de TAL en vue d'épurer ces données. Cette étape d'épuration de données aide à normaliser ce contenu initial de mesures et de lever certaines ambigüités. Elle consiste en quatre opérations élémentaires qui sont :

- le découpage du texte en paragraphes ;
- la tokenisation ;
- la suppression de mots vides ;
- la lemmatisation.

L'étape d'épuration de données est expliquée en détail dans le chapitre précédent. Une fois l'étape d'épuration de données accomplie, le contenu textuel est homogène et organisé en

paragraphes, il est prêt à être alimenté dans l'entrepôt de données. Ainsi, nous aurons un entrepôt de donnés textuel prêt pour une manipulation multidimensionnelle en occurrence la summarization. L'alimentation de l'entrepôt de données par les données textuelles épurées respecte l'architecture multidimensionnelle qui organise les données selon des faits et des dimensions. Ainsi, comme résultat de l'étape d'alimentation des mesures, nous aurons une base multidimensionnelle textuelle que nous appelons *BD*.

Pour la matérialisation de l'entrepôt textuel, nous avons *BD* une base multidimensionnelle qui est composée des données textuelles BD_text et de m dimensions $BD = \{DIM_1, DIM_2, DIM_3, \dots, DIM_m, BD\_text\}$. Une cellule simple de la base multidimensionnelle *BD*, là où nous n'avons effectué aucune agrégation, est donnée sous la forme $c = \{c_1, c_2, c_3, \dots, c_m, cel\_text\}$ avec c_i est une valeur pour la dimension DIM_i et *cel_text* représente le contenu de la mesure textuelle. La cellule *cel_text* est soit vide, *cel_text* = {}, soit composée d'un ensemble de paragraphes, *cel_text* = $\{p_1, p_2, \dots\dots, p_n\}$ où n est le nombre de paragraphes qui composent le contenu textuel de la cellule, sachant que chaque paragraphe est composé d'un ensemble de termes, nous avons : *cel_text* = $\{t_1, t_2, \dots, t_m\}$, avec *m* est le nombre global de termes composant le contenu textuel de la cellule.

- Application sur l'entrepôt d'article de presse

L'alimentation de l'entrepôt d'articles de presse nous donne une base multidimensionnelle que nous appelons *BD_Article*. Nous avons *BD_Article = {Auteur, Domaine, Style, Date, Publication, BD_Titre, BD_Contenu}.* Une cellule de la base multidimensionnelle est donnée sous la forme : c = {Auteur_val, Domaine_val, Style_val, Date_val, Publication_val, cel_Titre, cel_Contenu} avec : Auteur_val est une valeur pour la dimension *Auteur*, de même pour le reste des dimensions et *cel_Titre et cel_Contenu* sont le titre et le contenu textuel de l'article ou des articles correspondants aux valeurs des dimensions.

cel_Titre est l'ensemble de termes qui composent le titre, il est donné par *cel_Titre*=$\{t_1, t_2, \dots, t_m\}$. De même, *cel_Contenu* est un contenu textuel composé d'un ensemble de paragraphes, tel que *cel_Contenu* = $\{p_1, p_2, \dots\dots, p_n\}$, où *n* est le nombre de paragraphes. Sachant que chaque paragraphe est composé d'un ensemble de termes, nous avons : *cel_Contenu* = $\{t_1, t_2, \dots, t_m\}$, avec *m* est le nombre global des termes composant la mesure *cel_Contenu.*

5.6.2 Manipulation multidimensionnelle granulaire

Dans cette partie, nous allons nous intéresser à un champ de la manipulation multidimensionnelle au sein de notre entrepôt textuel qui est la manipulation à travers la granularité ou ce qui est appelé également les opérations de forage. L'analyse des données à travers des opérations de forage permet d'analyser les données tout en ayant la possibilité de modifier le niveau de détail utilisé pour observer ces données. Dans cette partie, nous proposons deux opérateurs de forage :

- *Term_Up* : ou le forage vers le haut, il consiste à observer les données textuelles avec une vision synthétisée (plus générale). Cet opérateur est basé sur le processus de summarization sémantique proposé.
- *Term_Down* : ou le forage vers le bas, il consiste à observer les données avec une vision plus détaillée.

5.6.2.1 Forage vers le haut : Opérateur Term-Up

Le forage vers le haut dans un entrepôt textuel aide à obtenir une vision plus globale des données textuelles ce qui revient à les agréger en des données plus synthétisées à l'aide d'un opérateur d'agrégation adapté. Dans ce travail, nous proposons un nouvel opérateur *Term_Up* pour l'agrégation des données textuelles organisées en paragraphes dans un environnement OLAP. L'opérateur *Term_Up* se base sur un processus de summarization (cf. chapitre 4) qui consiste à agréger l'ensemble des paragraphes en *k* termes les plus représentatifs. L'opérateur de summarization *Term_Up* a comme entrée un contenu textuel qui représente le résultat d'une requête multidimensionnelle sur l'entrepôt de données textuel.

Etant donnée *Q* une requête multidimensionnelle soumise à notre entrepôt de données textuel de *m* dimensions, elle est donnée sous la forme *Q*= $\{dim_1, dim_2, dim_3, \dots, dim_m\}$ avec $dim_i \in DIM_i$ U {*}. $dim_i = *$ veut dire que les données sont agrégées sur la dimension DIM_i. La réponse à la requête est un ensemble de cellules qui forme un cube *C*, tel que : *C*= $\{dim_1, dim_2, dim_3, \dots, dim_m, Res\_Text\}$ avec $dim_i \in DIM_i$ U $\{*\}$ et Res_Text est le texte contenu dans les cellules réponses, il est donné par : *Res_Text*=$\{cel\_text_1, cel\_text_2, \dots cel\_text_r\}$ avec r le nombre de cellules réponses. Nous avons : cel_text = $\{t_1, t_2, \dots, t_m\}$, avec m le nombre de termes composant le contenu textuel de la cellule. Ainsi, *Res_Text* représente la donnée d'entrée à notre opérateur *Term_Up*, il est formé par un ensemble de termes. L'opérateur *Term_Up* sert à agréger le contenu textuel de *Res_Text* en un ensemble de *k* termes les plus représentatifs. *Term_Up* est donné par :

Term_Up : $T \longrightarrow T$

$\{ t_1, t_2, ..., t_n \} \longrightarrow \{ t_1, t_2, ..., t_k \}$

Avec :

T est l'ensemble des termes formant le contenu textuel de BD_text

$\{ t_1, t_2, ..., t_n \}$ est l'ensemble de termes formant le contenu textuel à agréger *Res_Text*

$\{ t_1, t_2, ..., t_k \}$ est l'ensemble de k termes formant l'agrégat

5.6.2.2 Forage vers le bas : opérateur Term_Down

Le forage vers le bas dans un entrepôt textuel consiste à réaliser un « zoom » sur les données en passant des données agrégées aux données détaillées. Dans ce travail, nous proposons un nouvel opérateur pour le forage vers le bas, que nous appelons *Term_Down*. L'opérateur *Term_Down* consiste à restituer pour chaque terme *t* de l'agrégat, l'ensemble des paragraphes (données détaillées) où le terme *t* apparait en se basant sur la structure de données *index inverse*.

La structure *index inverse* est une structure qui abrite pour chaque terme, la liste de tous les paragraphes où il apparait. Posons *IV(t)* comme l'*index inverse* du terme *t*, nous avons : $IV(t_i) = \{(p_j , tf_{ij}) \mid tf_{ij} > 0\}$. *Index inverse* est construit pour chaque terme lors de la summarization en se basant sur la fréquence du terme au sein du paragraphe, cette fréquence nous permet d'ordonner les paragraphes selon l'importance du terme au sein de chacun d'eux.

Ainsi, nous avons :

Term_Down : $T \longrightarrow P$

$t \longrightarrow < p_1, p_2, ..., p_n >$

Avec :

T est l'ensemble des termes formant le contenu textuel de BD_text

P est l'ensemble des paragraphes formant le contenu textuel de BD_text

t est le terme sur lequel s'applique le forage vers le bas

$< p_1, p_2, ..., p_n >$ est un ensemble ordonné de paragraphes selon l'importance du terme t dans chacun des paragraphes.

5.7 Conclusion

L'objectif de ce chapitre était de proposer un environnement multidimensionnel adapté à l'agrégation de données textuelles à travers deux nouveaux opérateurs OLAP : *Term_Up, Term_Down.* Afin de prendre en charge les données textuelles, un modèle multidimensionnel de données en constellation a été proposé, ce modèle repose sur des mesures textuelles pour abriter les données textuelles. De plus, ce modèle est doté de deux opérateurs pour la manipulation des données textuelles à travers la granularité : *Term_Up* pour le forage en haut, et *Term_Down* pour le forage en bas. Un entrepôt d'articles de presse nous a servi pour l'illustration des propositions faites.

CHAPITRE 6
IMPLEMENTATION : RESULTATS ET DISCUSSION

6.1 Introduction

De manière à évaluer notre proposition, nous consacrons ce chapitre à un ensemble d'expérimentations réalisées sur un entrepôt d'articles de presse. Les données de l'entrepôt de données utilisées correspondent à des articles de presse tirés à partir de plusieurs publications internationales anglophones telles que : « The Guardian », « The New York Times », « The Times of India»...etc. Ils sont puisés à partir des sites officiels de ces journaux sur Internet. Par conséquent, nos collections de données sont limitées en volume et en source par le faible taux d'accès libres aux archives de grandes publications internationales.

L'évaluation de notre proposition, constituée essentiellement du processus de summarization, va se focaliser sur deux aspects importants :

- L'évaluation des performances du système proposé en matière de consommation de ressources système, à savoir : le temps de calcul et l'espace de stockage.
- L'évaluation de la qualité de la summarization proposée à l'aide de la mesure proposée à cet effet *T-Mesure*.

Enfin, ce chapitre est organisé de la manière suivante : d'abord nous allons présenter en détail l'approche retenue pour l'implémentation du processus de summarization ainsi que l'environnement de développement adopté. Par la suite, nous allons enchainer sur les expérimentations effectuées en commençant par celles liées aux performances système, et ensuite celles liées à l'évaluation de la summarization. Une discussion des résultats obtenus est abordée suivie par la présentation du fonctionnement de l'opérateur *Term_Down.* Nous finissons ce chapitre par une conclusion.

6.2 Approche retenue pour l'implémentation

Afin d'évaluer la faisabilité et les performances techniques de l'approche proposée, nous l'avons implémenté et réalisé quelques expériences. Par manque de plateforme dédiée, notre implémentation ne couvre pas la matérialisation de l'entrepôt de données, elle se focalise plutôt sur la summarization (l'implémentation des opérateurs) et l'évaluation de sa qualité. En effet, nous avons développé une application d'agrégation et de restitution « *Summarize* » qui représente le système d'analyse sur la granularité de l'entrepôt de données.

6.2.1 Architecture de l'application *Summarize*

« *Summarize* » est une application Java basée sur le concept orienté objet. Elle est modulaire, chaque module réalise une des fonctionnalités proposées par l'approche de summarization. L'application *Summarize* est composée des modules suivants (Figure 6.1):

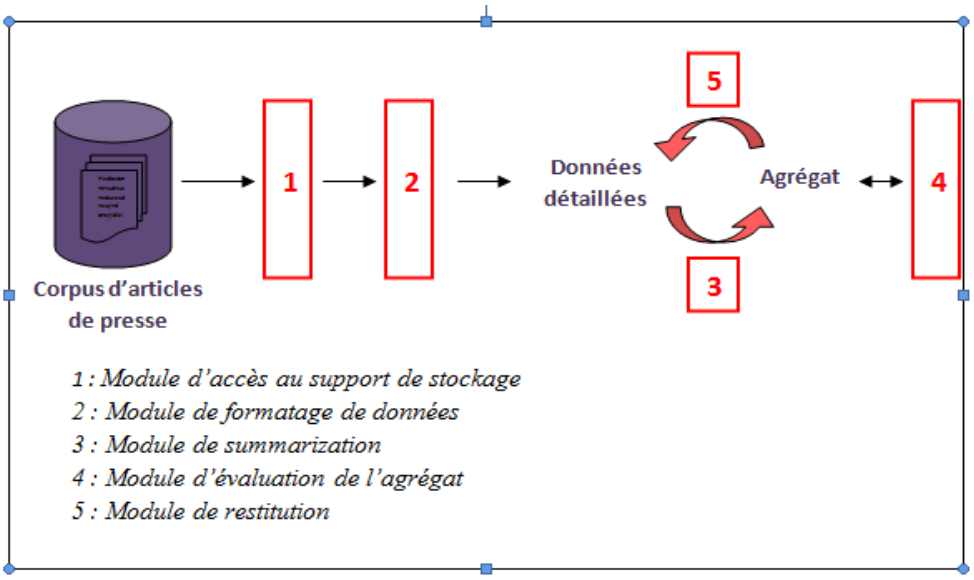

Figure 6.1 : Architecture de l'application « *Summarize* »

1. *Module d'accès au support de stockage :* il s'agit d'assurer l'accès au système de fichier afin de récupérer les données textuelles à agréger à partir du corpus.
2. *Module de formatage de données* : ce module est responsable de l'opération d'épuration de données, afin que ces dernières soient prêtes à être alimentées au sein de l'entrepôt de données.
3. *Module de summarization (forage vers le haut)*: Ce module sert à construire l'agrégat à partir des données détaillées selon la taille d'agrégat requise.
4. *Module d'évaluation de l'agrégat* : Ce module consiste à évaluer la qualité de la summarization réalisée et d'estimer une meilleure valeur pour la taille de l'agrégat.

5. *Module de restitution (forage vers le bas)* : Ce module consiste à restituer les données détaillées (paragraphes) à partir de l'agrégat.

6.2.2 Environnement de développement

L'implémentation a été réalisée dans un environnement Java en utilisant l'éditeur de Java Eclipse Indigo sur une machine avec une fréquence de 2.10 GHZ et une mémoire de 2 GO. Les données d'entrée de l'application sont des collections de données textuelles formant un corpus d'articles de presse, elles sont stockées directement dans le système de fichier NTFS.

6.3 Expériences et résultats de summarization

6.3.1 Déroulement d'expérimentation

Les expériences menées ont pour objectif global l'évaluation de la summarization proposée sur des collections d'articles de presse supposées être le résultat de requêtes multidimensionnelles sur l'entrepôt de données. Chaque collection comporte deux types de données textuelles : titre de l'article de presse et son contenu (Figure 5.2). De plus, chaque collection possède une liste de descripteurs donnés par un expert humain du domaine que couvrent les articles de presse. D'autre part, la suppression des termes vides est une partie intégrante de notre processus de summarization, la liste de termes vides retenue nécessite une personnalisation selon le domaine que couvre la collection. Ainsi, pour chaque collection, nous avons une liste de termes vides à ajouter à la liste Fox_List [Fox, 1990] retenue.

L'application de la summarization sur les données d'une collection nous donne un agrégat de taille *k*. Pour chaque collection, nous allons construire des agrégats de différentes tailles, par la suite nous allons évaluer la qualité des agrégats construits un par un. Ainsi, nous allons déduire un ensemble de meilleures valeurs pour la taille de l'agrégat (paramètre *k*) selon la qualité de la summarization effectuée.

6.3.2 Résultats pour les performances du système

Les systèmes OLAP sont des systèmes qui nécessitent des temps de réponse très courts. Nos expériences ont été effectuées sur 5 collections de test de différentes tailles. Les résultats relatifs au calcul des agrégats textuels en matière de temps de calcul et d'espace de stockage sont présentés dans le tableau suivant :

Collection	Taille de la collection		Temps de calcul (seconde)		Espace de stockage (MB)	
	articles	termes	Titre	Contenu	Titre	Contenu
Collection1	22	16633	3.06	946.21	5.4	6
Collection2	26	25300	3.21	991.95	6.3	7
Collection3	49	41955	7.64	2362.63	6.6	7.3
Collection4	71	83890	7.88	2706.10	11.7	18.5
Collection5	120	125845	8.76	3208.50	19.5	37.7

Tableau 6.1 : Résultats d'expérimentation pour les performances du système.

Les résultats obtenus en matière d'espace de stockage et surtout en matière de temps de calcul sont relativement bons et confirment la faisabilité technique et l'intérêt du processus de summarization proposé.

6.3.3 Résultats pour l'évaluation de la qualité de la summarization

Cette étape consiste à évaluer la qualité de la summarization effectuée à l'aide de la mesure de qualité de summarization proposée *T-Mesure.* En effet, nous allons évaluer la qualité de la summarization (forage vers le haut) selon la taille de l'agrégat. Cette évaluation est basée sur deux collections de test :

- Collection A :

Cette collection couvre les événements politiques majeurs qui se sont déroulés de 1990 à 1999. Elle est composée de 29 articles.

Liste de descripteurs :{diana, dodi, car, accident, iraq, kuwait, war, arab, muslim, world, people, reaction, us, air, attack, bush, serbian, conflict, palestine, israel, official, negociation, mandela, release, photographer, escape}

Requête : *Domaine* : politique

Date : de 01/01/1990 à 31/12/1999

Publication: The Guardian, the New York Times, Time US

Liste des termes vides à ajouter à List-Fox:{policy, political, politic, politics, time, day, month, year, hour, hours, news, press, days, months, years, times, night, nights,

yesterday, today, tomorrow, countries, country , city, cities, town, towns, zone, president, presidents, nation, nations}

- Collection B

Cette collection a pour objectif de couvrir la période de la crise économique qu'a connu la majorité des pays à partir de 2008. Elle est composée de 39 articles.

Liste des descripteurs : {bank, market, crisis, sector, consequence, government, debt, credit, increase, trade, service, security, people, purchasing , power, political, party}

Requête : *Domaine* : économie

Date : de 01/01/2008 à 01/07/2012

Publication: The Age, The Moscow Times, Today's Zaman, The New York Times, South China Morning Post.

Liste des termes vides à ajouter à List-Fox ={economy , economies, money, country, countries, land, lands, day, days, month, months, year, years, week, weeks, hour, hours, time, world, percent, dollar, euro}

6.3.3.1 Evaluation de la qualité de la summarization selon la taille de l'agrégat

Cette section consiste à évaluer la summarization proposée selon la taille de l'agrégat (le paramètre *k*). Cette évaluation est réalisée à l'aide de la mesure d'évaluation proposée *T-Mesure* et cela sur deux collections de test. Pour chaque collection, nous allons construire les agrégats correspondants à *k*=1, 2 ,3 , 4,..., 30 tout en calculant la valeur de *T-Mesure* correspondante à chaque agrégat. Sur la base des valeurs *T-Mesure* obtenues, nous allons proposer la liste des meilleurs agrégats avec les valeurs du paramètre *k* correspondantes. *T-Mesure* est calculée comme suit :

$$T - Mesure(\text{b}) = 1 - \frac{1.09}{\frac{0.09}{t-rappel} + \frac{1}{t-précision}} \quad (6.1)$$

$$avec : \quad t\text{-}rappel = \frac{nombre\ \ de\ termes\ \ d'agrégat\ \ connus\ \ comme\ \ descripteurs}{nombre\ \ de\ descripeurs\ \ \ du\ texte} \quad (6.2)$$

$$\text{Et :} \quad t - précision = \frac{nombre\ \ de\ termes\ \ d'agrégat\ \ connus\ \ comme\ \ descripteurs}{nombre\ \ de\ termes\ \ d'agrégat(k)} \quad (6.3)$$

Pour une meilleure lecture des résultats, nous introduisons *QS* avec : *QS=1 - T-Mesure.* Notons que nous avons adopté un seuil de similarité =1.

6.3.1.1.1 Résultats obtenus

Les résultats obtenus pour le calcul de pondération pour le contenu des articles des collections A et B sont représentés ci-dessous :

- Collection A :

Figure 6.3 : Représentation des résultats de pondération de la collection A par un nuage de points.

k	*terme*	*TF-IDF*	*K*	*terme*	*TF-IDF*
1	people	1.40	16	bush	0.85
2	car	1.30	17	photographer	0.82
3	diana	1.30	18	iraqi	0.82
4	kuwait	1.09	19	palestine	0.79
5	arab	1.04	20	hotel	0.77
6	world	1	21	convoy	0.77
7	road	0.98	22	woman	0.75
8	attack	0.96	23	palestinian	0.75
9	iraq	0.95	24	official	0.75
10	government	0.94	25	welcome	0.74
11	child	0.89	26	Mandela	0.73
12	serbian	0.88	27	air	0.73
13	home	0.86	28	muslim	0.73
14	dodi	0.86	29	half	0.72
15	party	0.85	30	report	0.72

Tableau 6.2 : Les résultats de pondération de la collection A.

➢ Collection B :

Figure 6.4 : représentation des résultats de pondération de la collection B par un nuage de points.

k	*terme*	*TF-IDF*	*K*	*terme*	*TF-IDF*
1	bank	2.78	16	inflation	1.41
2	crisis	2.17	17	currency	1.32
3	turkey	1.81	18	European	1.28
4	government	1.74	19	people	1.28
5	financial	1.71	20	cyprus	1.24
6	market	1.70	21	gdp	1.23
7	russia	1.70	22	debt	1.19
8	sector	1.57	23	export	1.15
9	rate	1.57	24	credit	1.13
10	oil	1.57	25	minister	1.09
11	global	1.53	26	central	1.07
12	economic	1.52	27	trade	1.06
13	growth	1.52	28	figure	1.04
14	price	1.50	29	import	1.02
15	Turkish	1.42	30	investment	0.99

Tableau 6.3 : Les résultats de pondération de la collection B.

Les résultats obtenus pour le calcul de *QS* pour les collections A et B sont présentés dans Tableau 6.4 et Tableau 6.5. Les lignes de tableau mises en gras correspondent à la restitution des termes connus comme descripteurs.

➢ Collection A :

k	*terme*	*QS*	*K*	*terme*	*QS*
1	**people**	**0.33**	**16**	**bush**	**0.65**
2	**car**	**0.50**	**17**	**photographer**	**0.68**
3	**diana**	**0.61**	18	iraqi	0.64
4	**kuwait**	**0.69**	**19**	**palestine**	**0.66**
5	**arab**	**0.74**	20	hotel	0.63
6	**world**	**0.78**	21	convoy	0.61
7	road	0.70	22	woman	0.58
8	**attack**	**0.74**	23	palestinian	0.56
9	**iraq**	**0.77**	**24**	**official**	**0.58**
10	government	0.71	25	welcome	0.56
11	child	0.65	**26**	**Mandela**	**0.58**
12	**serbian**	**0.68**	**27**	**air**	**0.59**
13	home	0.64	**28**	**muslim**	**0.61**
14	**dodi**	**0.67**	29	half	0.59
15	party	0.63	30	report	0.57

Tableau 6.4 : Résultats du calcul de QS pour la collection A

- Collection B :

k	*terme*	*QS*	*K*	*terme*	*QS*
1	**bank**	**0.43**	**16**	**inflation**	**0.81**
2	**crisis**	**0.62**	17	currency	0.76
3	**turkey**	**0.72**	18	european	0.72
4	**government**	**0.79**	19	people	0.69
5	**financial**	**0.83**	20	cyprus	0.66
6	**market**	**0.87**	21	gdp	0.63
7	**russia**	**0.89**	22	debt	0.60
8	**sector**	**0.91**	23	export	0.62
9	**rate**	**0.93**	24	credit	0.60
10	**oil**	**0.94**	25	minister	0.57
11	global	0.87	26	central	0.55
12	economic	0.81	**27**	**trade**	**0.57**
13	**growth**	**0.82**	28	figure	0.55
14	**price**	**0.84**	29	import	0.53
15	turkish	0.79	30	investment	0.52

Tableau 6.5 : Résultats du calcul de QS pour la collection B

Notons que nous avons *Sim*(issue, consequence)=1.

Une représentation graphique des résultats nous donne les deux graphiques ci-dessous, ils représentent les résultats obtenus pour QS des collections A et B.

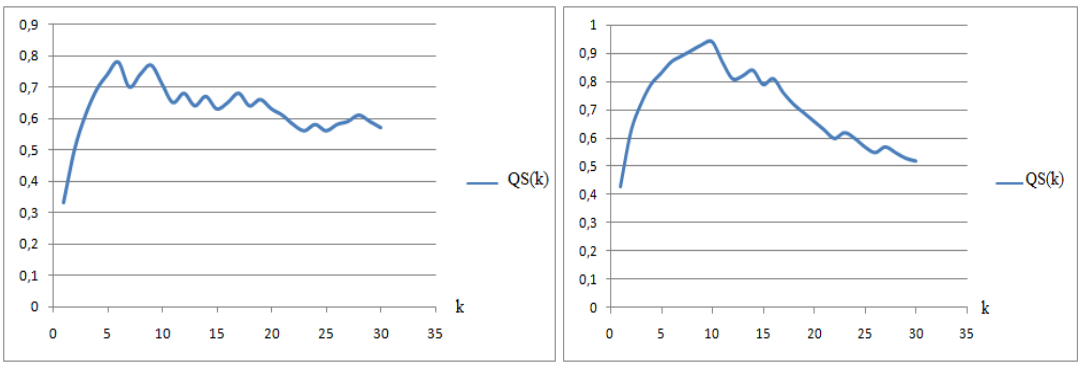

Figure 6.5 : Représentation graphique des résultats obtenus pour QS(k) des collections A et B.

Nous pouvons remarquer plusieurs pics (agrégats de meilleure qualité) dans la courbe représentant QS(k) pour les premières valeurs de k (allant de 1 à 20 pour la collection A, et de 1

à 15 pour la collection B). Au delà, la qualité des agrégats construits tend à diminuer progressivement (figure 6.5).

Par ailleurs, les meilleurs agrégats obtenus sont les suivants :

- Le meilleur agrégat obtenu pour la collection A est : {diana, car, people, arab, world, kuwait } avec k=6 et QS=0.78. Le deuxième meilleur agrégat obtenu est : {diana, car, people, arab, world, kuwait, road, attack, iraq} avec k=9 et QS=0.77.
- Le meilleur agrégat obtenu pour la collection B est : {bank, crisis, turkey, government, financial, market, russia, sector, rate, oil}, avec k=10 et QS=0.94. Le deuxième meilleur agrégat obtenu est :{ bank, crisis, turkey, government, financial, market, russia, sector, rate }, avec k=9 et QS=0.93.

- 6.3.1.1.2 Discussion

Les résultats obtenus sont globalement bons, et les agrégats obtenus sont relativement satisfaisants. Par ailleurs, en analysant les résultats obtenus, nous pouvons arriver à plusieurs conclusions :

- D'abord, il faut noter que la qualité de la summarization dépend fortement de l'indexation, puisque il s'agit finalement d'une comparaison paramétrée entre les descripteurs déduits par l'indexeur et les termes d'agrégat construit par le processus de summarization. Pour cela, il faut que l'indexation soit réalisée par des experts reconnus du domaine que traitent les données. Il faut également que l'indexation respecte certaines règles, par exemple : les descripteurs choisis doivent faire partie du vocabulaire de la collection de données, aussi, ces descripteurs doivent être des termes simples puisque la summarization ne prend pas en charge les termes composés.
- Ensuite, nous remarquons que la qualité du meilleur agrégat de la collection B (QS=0.94) est meilleure que la qualité du meilleur agrégat de la collection A(QS=0.78). Ce résultat est justifié, entre autre, par le fait que la qualité de la summarization est meilleure quand il s'agit des données cohérentes du point de vue thématique. En effet, les données de la collection B sont des données spécialisées et plus cohérentes puisque elles couvrent spécifiquement la crise économique de 2008. En revanche, les données de la collection B sont assez générales et dispersées thématiquement, elles couvrent tous les événements politiques et faits divers qu'a connus le monde de 1990 à 1999.
- Finalement, nous avons repéré la présence de plusieurs termes composés dans les agrégats construits même si le lien n'est pas établi par la summarization. Par exemple,

dans le meilleur agrégat obtenu pour la collection B, agrégat= {bank, crisis, turkey, government, financial, market, russia, sector, rate, oil}, nous avons « banque crisis » qui désigne la crise bancaire qui était le cœur de la crise économique, nous avons aussi « oil sector » et « financial market », qui tous les deux sont des secteurs qui ont connu beaucoup de perturbations durant la crise économique. En effet, souvent le recours aux termes composés pour exprimer certains concepts s'avère indispensable.

6.3.4 Forage vers le bas

L'approche proposée permet de réaliser un forage vers le bas via l'opérateur *Term_Down.* L'opérateur *Term_Down* permet d'aller des données agrégées vers les données détaillées ; il consiste à restituer pour chaque terme appartenant à l'agrégat, l'ensemble des paragraphes (données détaillées) où le terme apparait. En effet, les termes appartenant à l'agrégat sont associés via une structure appelée *Term_index* aux données textuelles d'origine (détaillées) issues des articles de presse. La structure *Term_index* est construite lors de l'application de l'opérateur de summarization Term_Up. Ainsi, pour chaque terme de l'agrégat, nous allons restituer l'ensemble des paragraphes d'où il a été tiré.

Par exemple, pour la collection B, nous avons l'agrégat {bank, crisis, turkey, government, financial, market, russia, sector, rate, oil}. Voici un extrait du résultat de l'application de l'opérateur *Term_Down* sur le terme « bank » :

Paragraphe1 : THIS month, the people's **Bank** of China finally gave the world a glimpse of what the country s post financial crisis economyis constructed on, It is built upon 10,000 local government investment platforms which, in turn, are floating on unsustainable rivers of credit, Ever since Beijing succeeded in reflating the economy two years ago, it has wrestled with competing imperatives, The long-term economic need was to dramatically slow overall credit growth to contain inflation, asset prices and resource misallocation

Paragraphe2 : First, as Andy Rothman of researchers CLSA puts it: China does not have a banking system, The **banks** that hold the bulk of local government debt are really just arms of the greater Chinese Communist Party enterprise, which can be topped up by other arms of the enterprise at any time, A systemic crash of the financial system is extremely unlikely, as the Party is arguably the world s most liquid financial institution, says Rothman

Paragraphe3 : Second, just as Chinese **banks** avoid market discipline, so do local governments, They are finding new ways to raise money, particularly by issuing bonds, which are being snapped up by different arms of the Party including state owned **banks**, state owned enterprises and other local government investment platforms, The PBOC has come clean on the extent of local government debt, but that s not the same as containing the problem.

Paragraphe4 : China s Big Four commercial **banks** are in no near term danger, says Anne Stevenson, of J Capital China, after returning from an investigation of local investment platforms in several cities, But we also believe that China s local debt is a far deeper systemic problem than we had understood, In the capitalist system, waste, corruption and excess eventually lead to belt tightening or bankruptcy, In China, there is always the capacity and therefore the temptation to gouge households a little harder to keep the industrial machine running at full throttle

Paragraphe5 : The Australian dollar has surged to a two month high, putting the squeeze on the economy and defying efforts by the Reserve **Bank** to curb the soaring currency, THE Australian dollar has surged to a two month high, putting the squeeze on the economy and defying efforts by the Reserve **Bank** to curb the soaring currency, The dollar broke the key US105 barrier early on Friday, defying a cut in interest rates and further entrenching its break from commodity prices, which have been a traditional driver, And most analysts believe the dollar will stay high

Paragraphe6 : Economists said Australia s triple A credit rating one of just a handful of countries in the world to have one was part of the reason why the currency remains high, At the same time, the official cash rate, while low by historical measures, is the highest among the developed countries, The shrinking pool of countries with the top credit grade Australia is now one of just seven left with a triple A rating means more global funds looking for investment havens, You generally welcome the fact that you ve got a triple A rating, but it s literally killing the **economy** with kindness, particularly the export oriented sectors of the **economy**, Arab **Bank** Australia treasury dealer David Scutt said

Paragraphe7 : Both the US and France have lost their top ratings, while Britain and Germany are on a negative outlook for their AAA ratings, meaning 2013 could see more money being pumped into the Australian market and dollar, Mr Scutt said, At the same time, QE, or quantitative easing, which has seen hundreds of billions pumped into the global financial system, has become a popular tool of monetary policy for the central **banks** of troubled economies this year, Westpac s chief currency strategist, Robert Rennie, said the dollar had traded in the US102 to US106 range this year as central **banks** in the US, Britain, Switzerland and Japan engaged in quantitative easing, This had a depressing influence on yields offshore and increased the attractiveness of the dollar s yields to global investor

6.4 Conclusion

Ce chapitre a été consacré à l'implémentation de l'approche de summarization proposée. Des expérimentations sur un corpus d'articles de presse ont été effectuées. Elles visent d'une part l'évaluation de la qualité de summarization proposée, et d'autre part de tester ses performances système. Les résultats obtenus pour l'évaluation de la summarization sont relativement satisfaisants, néanmoins, ces résultats dépendent fortement de la qualité des données du point de vue thématique et de la qualité de l'indexation effectuée sur ces données. Par ailleurs, les performances système obtenues sont relativement moyennes.

CONCLUSION

Synthèse :

Le travail présenté dans ce mémoire se situe dans le contexte général des entrepôts de données textuels, plus particulièrement, dans le cadre de l'intégration et la summarization des données textuelles au sein des entrepôts de données notamment à travers la technologie OLAP. Actuellement, les entrepôts de données se limitent à l'exploitation des données numériques, étendre ces entrepôts aux données textuelles revient principalement à les adapter relativement à deux volets :

- Une nouvelle modélisation multidimensionnelle de données permettant l'intégration de données textuelles.
- De nouveaux opérateurs OLAP capables d'agréger des données textuelles.

Dans ce cadre, nos principales contributions sont les suivantes :

***Sur le plan théorique** :*

- La proposition d'un processus de summarization sémantique de données textuelles permettant l'agrégation d'un contenu textuel par ses *k* termes les plus représentatifs. Ce processus se base sur une forme adaptée de la formule de pondération *TF-IDF* [Salton et al., (1975)], ainsi qu'un ajustement sémantique des poids des termes. L'ajustement sémantique des poids des termes est basé sur la comptabilisation des liens de synonymie existants entre les termes du texte à agréger à l'aide de la mesure de similarité sémantique de Lin [Lin, 1998] et du thésaurus WordNet [Chaumartin, 2007]. Un ordonnancement des termes du texte à agréger permet la sélection des *k* termes les plus représentatifs. Les différentes étapes du processus de summarization proposé sont formalisées par des pseudo-algorithmes.

- La proposition d'une mesure *T-Mesure* pour l'évaluation de la qualité de la summarization proposée. *T-Mesure* est inspirée de la mesure *E-Mesure* bien connue en recherche d'information, elle permet d'évaluer la qualité de l'agrégat obtenu

relativement à un ensemble de descripteurs prédéfinis. De plus, elle permet d'estimer les valeurs de *k* pour lesquelles la qualité de summarization est meilleure.

- La définition d'un modèle de données multidimensionnel adapté incluant des mesures textuelles. Ces dernières permettent l'intégration des données textuelles au sein de l'entrepôt de données.
- La proposition de deux opérateurs de summarization : *Term_Up* et *Term_Down* basé sur le processus de summarization élaboré. *Term_Up* et *Term_Down* sont deux opérateurs de forage à travers la granularité. *Term_Up* permet d'agréger un contenu textuel en *k* termes, tandis que *Term_Down* permet de restituer pour chacun des termes composants l'agrégat la liste des paragraphes où il est apparu.

Sur le plan pratique et technique :

Nous avons évalué notre approche à travers des collections d'articles de presse. Les expériences menées ont permis d'une part de mesurer les performances en temps de calcul et en espace de stockage, et d'autre part à mesurer la qualité de la summarization proposée. Les résultats obtenus pour les performances système montrent la faisabilité technique de l'approche proposée. De plus, les résultats obtenus pour l'évaluation de la qualité de summarization sont relativement bons et confirment l'intérêt de l'approche proposée. Néanmoins, ces résultats dépendent fortement de la qualité des données du point de vue thématique et de la qualité de l'indexation effectuée sur les données.

Perspectives :

Le travail réalisé dans ce mémoire ouvre diverses perspectives, nous citons les plus importantes :

- *La summarization par des termes composés* : Le processus de summarization proposé permet seulement la sélection des termes simples. Cependant, les termes composés sont plus spécifiques et peuvent porter plus de contenu sémantique. De plus, il y a des concepts qui ne peuvent être décrits avec exactitude qu'avec les termes composés. Etendre l'approche proposée aux termes composés va aider à donner plus de pertinence à l'agrégat.
- *L'utilisation d'une ontologie de domaine* : Le repérage de liens de synonymie se fait à l'aide du thésaurus WordNet. WordNet est un thésaurus avec un vocabulaire général non spécialisé, avoir une ontologie propre au domaine de l'analyse des articles de presse

va permettre d'affiner l'opération de recherche de termes synonymes et d'explorer davantage de liens sémantiques.

- *La validation de l'approche* : Montrer l'intérêt de l'approche proposée et la valider par la participation à une des compagnes d'évaluation spécialisées telle que TREC.

REFERENCES

1. Adamson G, Boreham J. "The use of an association measure based on characterstructure to identify semantically related pairs of words and document titles". In Information Storageand Retrieval, 10, p 253–60, 1974.

2. Anderson, J., Pérez-Carballo, J., "The nature of indexing: how humans and machines analyze messages and texts for retrieval: part ii: machine indexing, and the allocation of human versus machine effort", In Information Processing and Management 37, pages 255–277. Tarrytown, NY, USA, PergamonPress, Inc, 2001.

3. Bessonnat, D. « Le découpage en paragraphes et ses fonctions ». Revue Pratiques 57 : 81-100. 1988

4. Boughanem, M., « Systèmes de recherche d'informations : d'un modèle classique à un modèle connexionniste », Thèse de doctorat, Université Paul Sabatier, Toulouse III, 1992.

5. Ben Messaoud, R., Boussaïd, O., Loudcher, S, « A Data Mining-Based OLAP Aggregation of Complex Data : Application on XML Documents », International Journal on Data warehousing and Mining, Vol. 2, No 4, p.1-26, 2006

6. Sandra Bringay, Anne Laurent, Pascal Poncelet, Mathieu Roche, Maguelonne Teisseire, « Bien cube, les données textuelles peuvent s'agréger ! », EGC 2010: 585-596.

7. Chaudhuri, S., Dayal, U., Narasayya, V. R. « An overview of business intelligence technology ». Communications of the ACM, Vol.54, n°8, p.88-98. 2011.

8. Chaumartin François-Régis, « WordNet et son écosystème : un ensemble de ressources linguistiques de large couverture », BDL-CA, Montréal, 2007.

9. Christopher D. ,Manning, Prabhakar Raghavan, Hinrich Schütze. "An Introduction to Information Retrieval". Cambridge University Press Cambridge, England, 2009

10. Cleverdon, C. W. "Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems". Cranfield, U.K.: College of Aeronautics. Aslib Cranfield Research Project, 1962.

11. Codd, E., S. Codd, et C. Salley. « Providing OLAP (on-line analytical processing) to user analysts: An it mandate". In White Paper. 1993

12. Colliat, G. "OLAP, Relational and Multidimensional database systems". SIGMOD Record, vol.25(3), ACM Press, p. 64–69.

13. CROFT, W. B., and D. J. HARPER. "Using Probabilistic Models of Document Retrieval Without Relevance Information." *Documentation*, 35(4), 285-95. 1979

14. Deerwester S, Dumais S, Furnas G, Landauer T, et Harshman R, "Indexing by latex semantic analysis", Journal of the American Society for Information Science, 41(6):391–407, 1990.

15. Drzadzeswski, G. Tompa, F.W. "Exploring and analyzing documents with OLAP". Proceeding of the 5 th Ph.D Workshop Information and Knowledge. New York, USA. 2012

16. M.Ehrig, P.Haase, M.Hefke et N.Stojanovic. "Similarity for ontology-a comprehensive framework". In Workshop Enterprise Modelling and Ontology: Ingredients for Interoperability. 2004.

17. ESPINASSE.B. **«** Entrepôts de données et analyse en ligne ». l'Université d'Aix-Marseille, France. 2010.

18. FELLBAUM, C., "WordNet : an Electronic Lexical Database", The MIT Press. 1998.

19. Fox, C. , Frakes,W.B., Baeza-Yates,"Lexical analysis and stoplists", pages 102–130. R (eds) Prentice Hall, New jersey. 1992.

20. Fox, C., "A stop list for general text", ATetT Bell Laboratories Lincroft, New Jersey, 1990

21. Frakes, W. B., "Stemming Algorithms", in Frakes, William B. and Baeza-Yates, Ricardo (Eds.), *Information Retrieval: Data Structures and Algorithms*, Englewood Cliffs, NJ: Prentice-Hall, 1992. pp. 131-160

22. Inmon, B., "Building the Data Warehouse", 1st Edition, Wiley and Sons. 1992

23. Inmon, W., Strauss, D., Neushloss, G., "DW 2.0 The architecture of the next generation of the datawarehousing", Elsevier Press. 2008

24. Golfarelli, M., Rizzi, S., Saltarelli,E., "WAND: A CASE Tool for Workload-Based Design of a Data Mart", Decimo Convegno Nazionale suSistemi Evoluti per Basi di Dati (SEBD), p. 422–426, 2002.

25. Gammoudi M. M, « Méthode de décomposition rectangulaire d'une relationbinaire : une base formelle et uniforme pour la génération automatique des thesaurus et la recherche documentaire ». Thèse de doctorat de l'Université de Nice-Sophia Antipolis, 1993.

26. GRUBER T.R.,"Toward Principles for the design of Ontologies used for Knowledge Sharing".In Proc of International Workshop on Formal Ontology, Padova, Italy, March 1993.

27. Jiang, J.,Conrath, D.,"Semantic similarity based on corpus statistics and lexical taxonomy". InProceedings on International Conference on Research in Computational Linguistics, Taiwan, 1997.

28. Keith, S., Kaser, O. , Lemire, D. , "Analyzing large collections of electronic text using OLAP". Technical Report TR-05-001, UNBSJ CSAS. 2005

29. Kimball, R. , "The data warehouse toolkit: practical techniques for building dimensional data warehouses ", John Wiley et Sons, ISBN 0-471-15337-0, 1996.

30. Korhage, R., "Information storage and retrieval" John Wiley and Sons, Inc. 1997.

31. Kwok, K., "A neural network for probabilistic information retrieval", In Proceedings of ACM SIGIR, pages 21–30, 1989.

32. Lallich,G.,Ouerfelli,T., « La segmentation pour l'indexation d'un document technique : principe et méthodes ». Rencontre internationale sur l'extraction, le filtrage et le résumé automatique, Sfax. 1998.

33. Leacock, C., et Chodorow, M. , « Combining Local Context and WordNet Similarity for Word Sense Identification In WordNet : An Electronic Lexical Database" . C. Fellbaum, MIT Press. 1998.

34. Lee, J., H.,Kim, M.,H.,Lee, Y., J., "Information Retrieval Based on Conceptual Distance in IS-A Hierarchy", Journal of Documentation 49, pp. 188-207, 1993.

35. J.H.Lee, M.H.Kim et Y.J.Lee. "Information Retrieval Based on Conceptual Distance in IS-A Hierarchy". Journal of Documentation 49, pp. 188-207. 1993

36. Lee, J.,Grossman, D., Orlandic, R., "MIRE: A Multidimensional Information Retrieval Engine for Structured Data and Text". In Proceedings of the International Conference on Information Technology: Coding and Computing, pages 224–229. IEEE Computer Society, Washington, DC. 2002.

37. Lin., D., "An Information-Theoretic Definition of similarity". In Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98)". Morgan- Kaufmann: Madison, WI. 1998.

38. Lin, C. X., Ding, B., Han, J., Zhu, F., et Zhao, B, " Text Cube: Computing IR measures for multidimensional text database analysis", Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, 905-910. 2008

39. Lin, C. X., Ding, B., Zhao, B., Han, J., Zhai, C., "TopCells: Keyword-Based Search of Top-k Aggregated Documents in Text Cube", IEEE Transactions on Knowledge and Data Engineering (TKDE) (Special Issue: Keyword Search on Structured Data), 23(12):1795-1810, 2011.

40. Lioret,E., "Text summarization: an overview", Université Alicante, Espagne. 2006.

41. Luhn, H. "A statistical approach to mechanised encoding and searching of litary information". IBM Journal of Research and Development. 1957

42. McCabe,C., Lee, J., Chowdhury,A., Grossman, D.,A., Frieder, O., "On the design and evaluation of a multi-dimensional approach to information retrieval", 23rd Intl. ACM Conf. on Research and Development in Information Retrieval (SIGIR), ACM Press, p. 363–365. 2000.

43. Maron, M., Kuhns,J., "On relevance, probabilistic indexing and information retrieval".Journal of the Association for ComputingMachinery, 7: 216–244. 1960.

44. MOTHE, J., « Modèle connexionniste pour la recherche d'information, expansion dirigée de requête et apprentissage », Thèse de doctorat, Université Paul Sabatier, Toulouse III, 1994.

45. MOTHE, J., HERNANDEZ N, H., Nigro, O., Císaro, S. G., Xodo, D., "TtoO: Mining thesaurus and texts to build and update a domain ontology, In: Data Mining with Ontologies: Implementations, Findings, and Frameworks". Idea Group Inc, 2007.

46. Nassr,N, « Croisement de langues en recherche d'information : traduction et désambiguïsation de requêtes », Université Paul Sabatier de Toulouse, thèse de doctorat, 2002.

47. Park,B.K., Han,H., Song,Y., "XML-OLAP: A Multidimensional Analysis Framework for XML Warehouses". 7th Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), LNCS 3589, Springer, p. 32–42. 2005.

48. Pérez,J.M., Berlanga, R., Aramburu,J.M « A Document Model Based on Relevance Modeling Techniques for Semi-structured Information Warehouses". In Proc. of DEXA, pages 318–327. 2004.

49. Pérez,J.M., Berlanga, R., Aramburu,J.M , Pedersen,T.,B., "A Relevance-Extended Multi-dimensional Model for a Data Warehouse Contextualized with Documents", In Proceedings of the 8th ACM international workshop on Data warehousing and OLAP, pages 19–28. ACM Press, New York, NY. 2005

50. Pérez,J.M., Berlanga, R., Aramburu,J.M , Pedersen,T.,B.," Integrating Data Warehouses with Web Data: A Survey", DB technical report, TR-18. 2006.

51. Piwowarski, B., « Techniques d'apprentissage pour le traitement d'informations structurées : application à la recherche d'information », Thèse de doctorat de l'université de PARIS 6, 2003.

52. Pujolle, G., Ravat, F.,Teste, O., Tournier, R., « Fonctions d'agrégation pour l'analyse en ligne (OLAP) de données textuelles ». Ingénierie des Systèmes d'Information 13(6), 61–84. 2008

53. Quillian M.R, « Semantic information processing », Semantic memory, 1968.

54. Rada, R. , Mili, H. , Bichnell, E., Blettner, M.« Development and application of a metric on semantic nets ». IEEE Transaction on Systems, Man, and Cybernetics: pp 17-30, 1989.

55. Ravat, F., Teste, O., Tournier, R., Zurfluh, G. "Top-keyword: An aggregation function for textual document OLAP". Lecture Notes in Computer Science, 5182, 55-64.

56. Resnik, P.,"Using information content to evaluate semantic similarity in a taxonomy". In IJCAI, pages 448–453, 1995.

57. Resnik, P.,"Semantic similarity in taxonomy: An information-based measure, and its application to problems of ambiguity in natural language", Journal of Artificial Intelligence Research, pages 95–130, 1999.

58. Rijsbergen, C., V., "In Information retrieval experiments". Information retrieval, Butter worths, London, 2nd edition, 1979.

59. Robertson, S., Sparck-Jones, K., "Relevance weighting of search terms", JASIS, 27(3):129–146, 1976.

60. Robertson S. Sparck-Jones K., "Simple proven approaches to text retrieval", Tech rep tr356, Computer Laboratory University of Cambridge, 1997.

61. Robertson, S. E., Walker, S. , "Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval", In Proceedings of SIGIR'94. 1994

62. Robertson, S., "Understanding Inverse Document Frequency: On theoretical arguments for IDF". Microsoft Research, London, UK, Journal of Documentation 60 no. 5, pp 503–520. 2004.

63. Salton, G. "The SMART retrieval system: Experiments in automatic document processing ». Prentice-Hall. 1971

64. Salton , G.,Wong. A. , Yang, C. S. "A vector space model for automatic indexing". Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975

65. Salton, G., McGill, M.J. "Introduction to Modern Information Retrieval".McGraw-Hill, 1983.

66. Salton, G., Buckley, C.,"Term Weighting Approaches in Automatic Text Retrieval", Cornell University, Ithaca, NY, 1987.

67. Salton, G., Yang, C., "On the specification of term values in automatic indexing", Journal of Documentation, 29:351–372, 1973

68. Savoy, J. "Stemming of French words based on grammatical categories", Journal of the American Society for Information Science, 44(1), p. 1-9, 1993.

69. Sheridan, P., Smeaton, A., "the application of Morpho-syntactic language processing to effective phrase matching", Information Processing and Management, 28(3):349–370, 1992.

70. Schmid, H. "TreeTagger – a language independent part–of–speech tagger". Institute for Computational Linguistics of the University of Stuttgart. 2011

71. Slimani T. Yaghlane B. B, Mellouli K. "A new similarity measure based on edge counting". In Proceedings of world academy of science, engineering and technology, Vol. 17, December 2006.

72. Slimani,T., Ben Yaghlane, B., Mellouli, K., "Une extension de mesure de similarité entre les concepts d'une ontologie ». 4rth International Conference: Sciences of Electronic,Technologies of Information and Telecommunications– TUNISIA. 2007

73. Sullivan, D., "Document Warehousing and Text Mining", Wiley John et Sons, 2001

74. Teste, O., « Modélisation et manipulation des systèmes OLAP : de l'intégration des documents à l'usager ». Memoire pour l'obtention de l'habilitation a diriger des recherches. Université Paul Sabatier (Toulouse III). 2009

75. Tournier, R., « Analyse en ligne (OLAP) de documents », doctorat de l'université de Toulouse. 2007

76. Tversky, A., "Features of similarity:Psychological Review", 84(4):327–352, July1977.

77. Tseng F.S.C., Chou A.Y.H "The concept of document warehousing for multidimensional modeling of textual-based business intelligence". Journal of Decision Support Systems (DSS), vol.42(2), Elsevier, p. 727–744. 2006

78. Van, G, A., Schreiber, Th.,Wielinga, B. J.,"Using explicit ontologies in KBS development". International Journale Human-Computer studies, 1997.

79. Wong, S., Ziarko, W., et Wong, P., "Generalized vector space model information retrieval", In Proceedings of the 8th annual international ACM SIGIR Conference on Research and development in Information Retrieval, pages 18–25. ACM, 1985.

80. Wu, Z. et Palmer, M., "Verb semantics and lexical selection", In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, pp 133- 138. 1994.

81. Yates, R.B., Neto, R. « Modern Information Retrieval », ACM Press, Addison Wesley, pages: 70-77, 1999.

82. Zargayouna, H., « Indexation sémantique de documents XML ». Thèse en informatique, Université Paris XI Orsay, 2005.

83. ZHANG, D., ZHAI, C., HAN, J. , "MITEXCUBE: micro text cluster cube for online analysis of text cells", Proc. 2011 NASA Conf. on Intelligent Data Understanding (CIDU'11), Mountain View, CA, Oct. 2011

84. Zipf ,G., "Human Behaviour and the Principle of Least Effort", Addison-Wesley, 1949.