F.S.D .....N°D'ordre .....

Université Saad DAHLAB de Blida

Faculté des Sciences

Département d'Informatique

Mémoire présenter par :

M^elle^Khouaouci Hafida
M^r^ Taoug Mohamed Chahine

En vue d'obtenir le diplôme de Master

Domaine: MI
Filière: Informatique
Spécialité: Informatique

Option: Ingénierie de logiciel

Sujet :

Un entrepôt texte pour l'analyse des revues de presse

Soutenu le: devant le jury composé de :

M.Bala Mahfoud	Président
M. Hadj yahya	Rapporteur
M. Chikhi	Examinateur
M^lle^Benblidia Nadjia	Promotrice
M^lle^Attaf Sarah	Promotrice

2012-2013

Remerciement

DEDICACE

Je dédie ce travail ;
À mes parents que je ne remercierai jamais assez pour toute l'aide qu'ils m'ont prodiguée,
À mes frères et mes sœurs
À tous mes amis,
À toutes les personnes que je connais et que je n'ai pas citées

Hafida.

Liste des figures

Table des figures

Liste des abréviations

Liste des abréviations :

DTPC	District Tourism Promotion Councils
EM	Espérance-Maximisation
GUI	Graphical user interface
HA	Hiérarchie d'Attributs
HO	Hiérarchie d'Objets
HPLSA	Hierarchical Probabilistic Latent Semantic Analysis
IBM	International Business Machines Corporation
ID	Identifiant(Attribut)
IDE	Integrated Development Environment
J2EE	Java 2 Enterprise Edition
J2ME	Java2 Micro Edition
JMF	Java Media Framework
LDA	Latent Dirichlet Allocation
LSA	Analyse sémantique latente
MASHA	Multinomial Asymmetric Hierarchical Analysis
OLAP	On-Line AnalyticalProcessing
OMG	l'Object Management Group
OMT	Object Modeling Technique
OOSE	Object Oriented Software Engineering
OSGI	Open Services Gateway initiative
PLSA	Probabilistic Latent Semantic Analysis
QCM	Questionnaires à choix multiples
SGBDR	Système de Gestion de Bases de Données Relationnelles
SVD	Décomposition en valeurs singulières
TAL	Traitement Automatique dela Langues
TF-IDF	Term Frequency-Inverse Document Frequency
UML	Unified Modelling Language
WSD	Word Sense Disambiguation

Sommaire

Partie II Conception et implémentation

Résumé

Résumé

Analyser les données textuelles est devenu très essentiel à cause de leurs volumes importants et la quantité d'information qu'elles puissent contenir. Afin de pouvoir prendre profit des informations qu'elles contiennent et d'extraire de l'information pertinente, il est devenu plus que nécessaire d'adapter un modèle multidimensionnel de représentation de données, permettant de décrire ces données textuelles de façon suffisamment formelle, pour qu'elles puissent être prêtes à l'analyse. A cette fin, notre choix s'est porté sur l'implémentation d'un outil d'entreposage de données textuelles.

Mots clés : Entrepôt de données, Données textuelles, modèle de représentation de données.

Abstract

Because of their large volume and the amount of information they may contain, analyzing textual data is very essential in order to take advantage of the information they contain. To analyze this type of data, it is extremely necessary to adapt a multidimensional model of data representation which allows the textual data to be sufficiently described in a formal way, so they can be ready for the analysis. To this end, implementing a tool for storing textual data is our choice.

Keywords: data warehouses, textual data, models of data representation.

ملخص

نظرا لأحجام البيانات النصية الكبيرة وكمية المعلومات التي يمكن أن تستوعبها فإنه من الضروري جدا تحليلها, وذلك من أجل الاستفادة من المعلومات التي تحتوي عليها، و لكي نستطيع تحليل هذه البيانات من الضروري تكييف نموذج متعدد الابعاد لتمثيل البيانات الذي يسمح بوصف البيانات النصية بشكل رسمي كافٍ بحيث تكون جاهزة للتحليل ، ومن أجل تحقيق ذلك وقع اختيارنا على استعمال أداة لتخزين البيانات النصية.

كلمات البحث: مستودع البيانات،البيانات نصية،نماذج تمثيل البيانات.

I. Avant-propos

L'essor des technologies de l'information, avec l'avènement d'internet et des réseaux, a accru le volume des informations disponibles de manière considérable. Désormais, les entreprises font face à un volume croissant de données qui transitent au sein de leur système d'information et cette masse d'informations rend difficile leur exploitation.
Pour faire face au problème de volume, les systèmes d'aide au processus de prise de décision (ou systèmes d'aide à la décision) ont été mis en place au sein des entreprises. Ces systèmes permettent un traitement synthétique de l'information pour faciliter les prises de décisions.
Les résultats de ces systèmes sont exploités par les décideurs en vue d'effectuer des analyses pour piloter au mieux les entitéséconomiques dont ils sont responsables.

L'entreposage et l'analyse en ligne (OLAP) sont deux techniques utilisées dans les systèmes d'aide à la décision. Ces techniques ont maintenant largement fait leurs preuves, dans les domaines de traitement de données numériques simples.

II. Problématique

Le processus décisionnel ou les systèmes d'information décisionnels sont nés d'un besoin exprimé par les entreprises, besoin non satisfait par les systèmes de bases de données traditionnels. De ce besoin sont apparus dans les années quatre-vingt-dix les entrepôts de données (data warehouses), comme une technologie clef pour les entreprises désirant améliorer l'analyse de leurs données. Les technologies d'entreposage de données, d'analyse en ligne et de fouille de données, ont maintenant largement fait leurs preuves, dans les domaines de la gestion de données et de l'extraction de connaissances à partir de données (ECD).

A l'heure actuelle, avec l'accroissement continu du volume de l'information dans les entreprises, Les données à analyser ne sont plus seulement numériques ou symboliques, mais sous différents formats (textes, images, son, vidéos…etc.), provenant de sources différentes. De cela est né un nouveau besoin dans les entreprises, celui de l'analyse de données dite « complexes ». L'exploration des données textuelles dans ce cadre implique de nombreux problèmes, notamment celui de leur structuration et stockage d'une part et leur analyse d'autre part.

Dans ce mémoire de master nous posons la problématique de l'analyse des données textuelles ainsi : quels sont les modèles adaptés à l'analyse des données textuelles ? Que serait-il notre mesure d'analyse ? Et par rapport à quel axe ?

III. Objectifs :

Nous visons à travers ce travail à construire un entrepôt textuel qui nous permettra d'effectuer des analyses multidimensionnelles sur les données textes afin de prendre profit des informations qu'elles contiennent. Pour cela nous allons suivre les étapes suivantes :

1. Choisir un modèle d'entrepôt de textes afin de l'implémenter.
2. Alimenter le modèle implémenté en faisant appel à des méthodes de recherche d'information et de fouilles de données.
3. Construire des cubes OLAP textuels.

IV. Organisation du mémoire

On a structuré le mémoire comme suit :

- La première partie l'état de l'art est composée de deux chapitres :

Nous présentons dans le premier chapitre une étude détaillée sur les entrepôts textes.

Dans le deuxième chapitre nous présentons les différentes méthodes d'analyse de données textuelles.

- Ensuite dans la deuxième partie conception et implémentationnous allons aborder dans le premier chapitre le volet conception et dans le deuxième chapitre les détails de la réalisation (L'environnement de développement matériel et logiciel) et la présentation des différentes interfaces de notre application.

Partie I

L'état de l'art

Chapitre I

Les entrepôts textes

1. Introduction

Les entrepôts de données et les systèmes d'analyse en ligne OLAP (On-Line Analytical Processing) fournissent des méthodes et des outils permettant l'analyse de données issues des systèmes d'information d'entreprises. Mais, seules 20% des données d'un système d'information constituent des données analysables par les systèmes OLAP actuels. Les 80% restantes, sont constituées de documents et demeurent hors de portée de ces systèmes faute d'outils ou de méthodes adaptées. Pour pallier à ce problème, plusieurs travaux qui portent sur la modélisation multidimensionnelle de ces documents car ce sont aussi des données qui doivent être analysées car leur contenance est importante pour effectuer une analyse total et correcte surtout dans des domaines ou l'information est importante (du point de vued'analyse et prise de décision), parmi ces données nous allons nous intéresser à des données complexes qui sont des données textes.

Pour représenter les données textes dans un entrepôt de données plusieurs travaux ont été faits qui ont données des déviations sur des modèles multidimensionnel pour les représenter, dans ce chapitre, une description détaillée de ces modèles va être présentée, à la fin de ce chapitre nous allons déduire le modèleà adopter durant le reste de notre mémoire.

2. Modèles Multidimensionnels d'entrepôts textes

2.1Modèle Galaxie [Tou, 07]

Le modèle en Galaxie a été conçu pour répondre à la problématique d'analyse en ligne (OLAP) de documents XML, en prenant en compte : la structuration hiérarchique des données textuelles des documents, la représentation des liens intra ou inter-documents, l'intégralité des données textuelles qui constituent les documents ainsi que les métadonnées associées à un document. Le modèle en Galaxie est définit par la généralisation du concept de constellation **[Kim, 96]**, son approche consiste à décrire un schéma multidimensionnel par l'unique concept de dimension ; la notion de fait est supprimée. Une galaxie est un regroupement de dimensions liées entre elles par un ou plusieurs nœuds centraux ; chaque nœud

modélise les dimensions compatibles pour une même analyse. De plus, les éléments constituant les dimensions peuvent être interconnectés en travers de liens.

Une galaxie est définit par un ensemble de dimensions noté « **DG** », une fonction associant chaque dimension **Di** à une autre dimension **Dj** compatible pour une analyse tel que i≠j, noté StarG et un ensemble de liens qui représentent les liens intra ou inter-documents noté **LKG**.

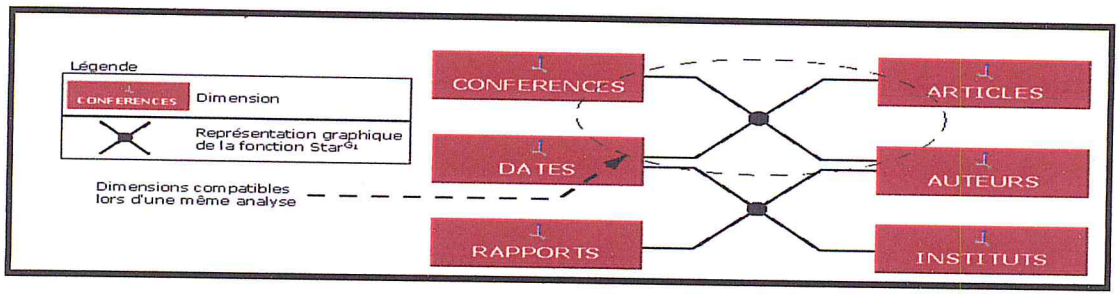

Figure 01 : Exemple simplifié d'une galaxie G [Kim, 96].

Une dimension dans le modèle Galaxie est caractérisée par des attributs organisés de manière hiérarchique. Chaque attribut modélise un niveau de granularité de l'axe d'analyse Les dimensions pouvant être considérées comme des axes d'analyse, les attributs sont autant d'indicateurs d'analyse potentiels. Sur une dimension il ya deux sortes d'attributs : des attributs appelés paramètres qui représentent les niveaux de granularité de la dimension, et d'autres attributs dits « attribut faibles »associés aux paramètres, complétant la sémantique de ceux-ci. La hiérarchie de la dimension Auteur est représentée dans la figure suivante :

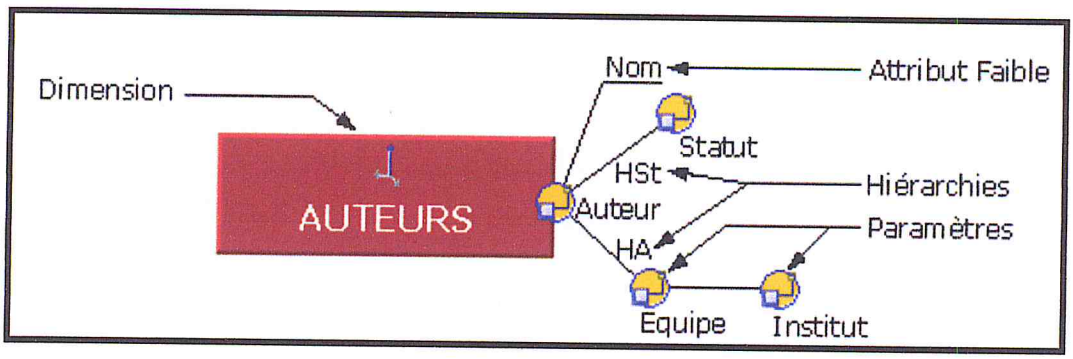

Figure 02: hiérarchie de la dimension Auteur [Kim, 96].

L'application du modèle en galaxie à l'analyse de documents, permet la représentation de la structure hiérarchique de ces derniers au moyen de dimensions documentaires. Les liens internes ou externes des documents sont conservés et représentés dans le modèle pour permettre leurs utilisation lors de la navigation. Enfin, l'absence de sujet d'analyse prédéfinis fournit à l'utilisateur une flexibilité adéquate pour lui permettre de réorienter l'analyse en cas de non-sens sur une analyse de texte.

2.2. Topic Cube [Zha et al,09]

Topic Cube est un modèle de donnée qui étend le data cube traditionnel en ajoutant une hiérarchie de thèmes, ainsi que les mesures probabilistes des documents analysés à travers un modèle thématique probabiliste. Topic Cube, supporte les deux composants de base d'OLAP sur la dimension texte :

- **Hiérarchie de la dimension sujet (dimension thématique) :** transformer la sémantique des documents en une hiérarchie thématique arbitraire spécifiée par un analyste.

- **Mesure contenu dans le texte :** résumer le contenu des documents texte dans une cellule.

Le choix des auteurs, a porté sur l'utilisation du PLSA (Probabilistic latent Semantic Analysis) comme étant le modèle thématique probabiliste, un choix justifié par le fait que PLSA répond parfaitement à un grand nombre de problèmes d'exploration de données, telle que la modélisation hiérarchique de thèmes.

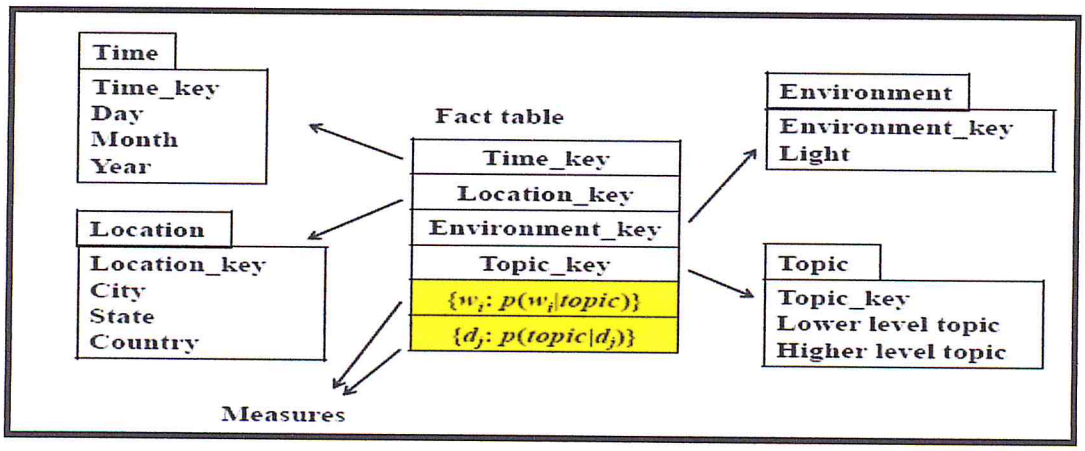

Figure 03 : schéma en étoile d'un Topic Cube [Zha et al,09] .

Topic cube est construit à base d'une base de donnée texte D, et un arbre hiérarchique de thème H. De plus les dimensions standards définies dans D, Topic cube a aussi une dimension qui correspond à l'arbre hiérarchique de thème H. Les Drill down, roll up dans cette dimension, sont des techniques d'exploration permettent à l'utilisateur de visualiser les données par rapport à plusieurs granularités de thèmes.

2.3. MicroTextCluster [Zhan et al,11]

L'infrastructure nommée « MicroTextCluster Cube »vise à rendre l'analyse en ligne des cellules texte plus rapide, en effectuant autant que possible de prétraitements dans la phase hors ligne. Le prétraitement des documents consiste à générer un bon nombre de micro-clusters pour compresser les documents similaires dans la même cellule texte (chaque cellule texte contient un certain nombre de document), cette compression (en micro-cluster) permet de retenir des informations sémantiques essentielles sur les cellules textes. Chaque micro-cluster contient sa taille et son propre vecteur mean (un vecteur de termes pondérés, où le poids de chaque terme est le poids moyen de ce terme par rapport à tous les documents contenus dans un micro-cluster précis), la figure suivante nous illustre un exemple d'un MiTexCube.

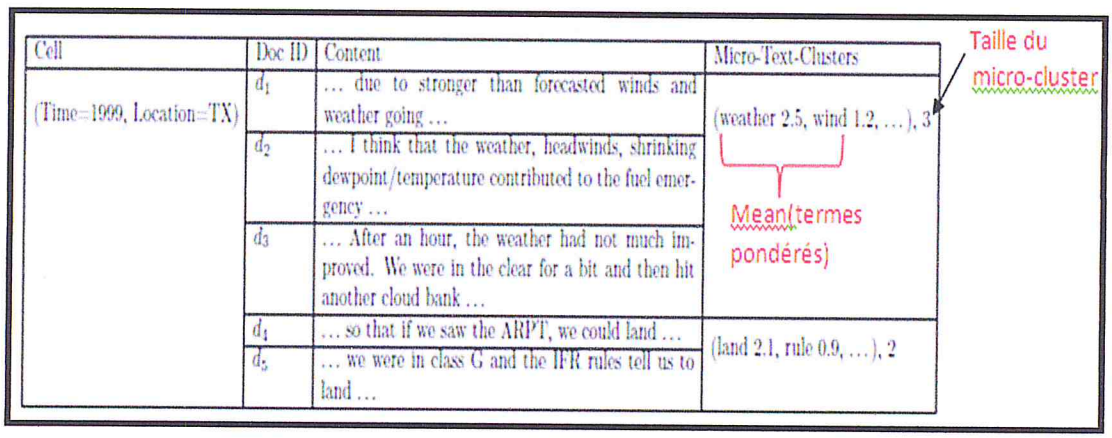

Figure 04 : Exemple d'un MixTexCube[Zhan et al,11] .

2.4. Modèle multidimensionnel d'objet complexes [Bou,11]

Le modèle multidimensionnel d'objet complexe est un modèle conçu pour répondre à la problématique d'analyse des données complexes, il est basé sur le paradigme objet, un choix justifié par la capacité des modèles orientés objets de représenter les objets de l'univers et de capter la sémantique qu'ils véhiculent, notamment dans les liens avec les autres objets.

Le modèle d'objet complexe est un modèle à trois niveaux : le premier niveau représente le diagramme de classe détaillée des faits candidats et des dimensions candidates, le deuxième niveau représente un diagramme de package pour chaque classe du premier niveau, le troisième niveau représente un diagramme de package qui résulte de la projection d'un package du deuxième niveau comme étant un fait et de regrouper les niveaux dimensions résultants.

2.4.1. Concepts de base du modèle d'objet complexe

A. Objet Complexe : Entité abstraite ou concrète qu'il est possible d'analyser en tant que sujet ou axe d'observation. Il est définit par :

- Attribut simple : attribut de class en UML.
- Attribut complexe : composé d'un ou de plusieurs attributs simples ou complexes.

B. Relation complexe : Les liens entre les objets de haut niveau, ils définissent les axes d'analyses de certains objets par rapport à d'autres (héritage, association). Lorsqu'il s'agit d'une relation de composition entre objets, l'objet composant est définit comme étant un attribut complexe de l'objet composite.

C. Hiérarchie attributs : relations intra-objets complexes. Il s'agit de relations qui organisent certains attributs de l'objet en hiérarchie.

D. Hiérarchie d'objets : relations inter-objets complexes. Permet d'effectuer des opérations d'agrégations entre les objets complexes. Elle définit un ordre partiel entre certains objets du monde réel selon leur degré de granularité.

E. **Schéma multidimensionnel** : réunit : (A) l'ensemble des objets complexes, (B) l'ensemble des relations complexes, (C) l'ensemble des hiérarchies d'attributs et (D) l'ensemble des hiérarchies d'objets.

La figure suivante nous représente le méta modèle d'objet complexe

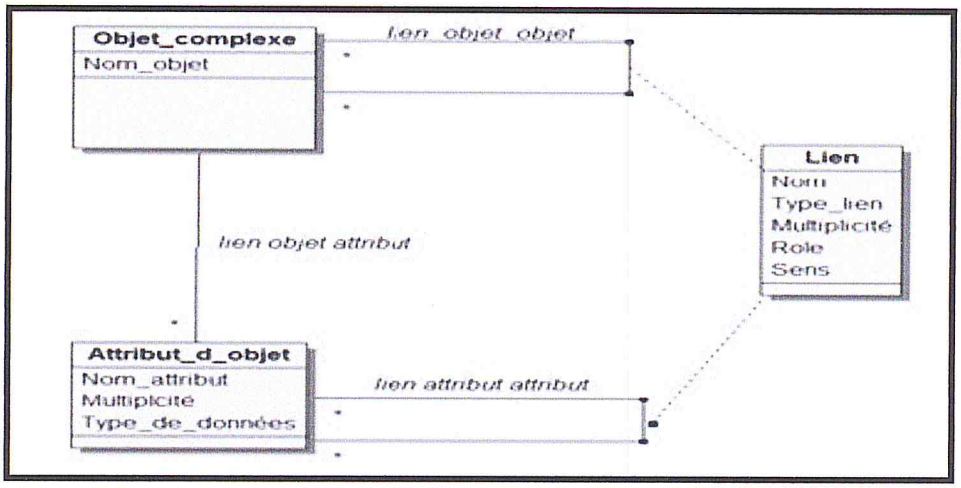

Figure 05: Méta modèle d'objet complexe [Bou,11] .

3. Etude Comparative [Att et al,13]

La modélisation des données textuelles dans un but d'analyse implique de nombreux problèmes notamment en ce qui concerne la prise en compte de leur structure et de leur sémantique d'une part et la flexibilité d'analyse d'autre part. Aussi les données textuelles comportent des mesures non numériques auxquelles il est nécessaire de définir de nouvelles fonctions d'agrégation.

3.1. Critères de comparaison [Att,13]

3.1.1. Aspect structurel

La modélisation des données textuelles dans un but d'analyse peut considérer le document texte comme étant une donnée élémentaire. L'objectif consiste alors de structurer et de stocker les documents dans une base de documents textes et de les préparer à l'analyse, sans prendre en compte de la structure interne des documents.

Toutefois, cette approche de modélisation ne répond pas à toutes les exigences d'un décideur, tel que l'analyse des sections sportives d'un ensemble de journaux.

Ce type d'analyse n'est pas supporté par cette approche car la structure interne des documents divise le document en plusieurs niveaux hiérarchiques, ce qui permet une analyse sur différents niveaux de granularité qui n'est pas prise en considération. Ainsi nous définissons un modèle qui prend en compte l'aspect structurel des documents, et permettant une analyse multidimensionnelle sur de différents niveaux structurels.

3.1.2. Aspect sémantique

L'extraction et la représentation de la sémantique véhiculée dans les données textuelles présentent une problématique déjà traitée dans la littérature dans les domaines d'extraction de connaissances et de la recherche d'information. Tandis que dans les entrepôts de données, la prise en compte de cet aspect important dans la modélisation multidimensionnelle est une nouvelle problématique. Répondre à cette problématique reviens à trouver une manière d'incorporer la sémantique des données textuelles et de la modéliser au sein d'un cube de données.

3.1.3. Flexibilité d'analyse

Dans les systèmes décisionnels classiques un fait représente un sujet d'analyse prédéfini. La définition d'un fait rend la spécification d'analyses peu flexible, car le décideur se voit contraint d'employer ces faits comme sujets. La flexibilité d'analyse est apparue comme un nouveau besoin exprimé par les décideurs. Elle réside dans le fait où le sujet d'analyse n'est pas prédéfini au préalable mais choisi au moment de l'analyse. Dans le domaine de l'analyse des données textuelles, nous percevons que le problème de flexibilité est assez complexe. Ainsi nous posons cette problématique autrement, lors d'une analyse textuelle, le contenu sémantique de ces données peut être vu comme étant une mesure d'analyse (K-top keyword, Topic).

Comme il peut être considéré comme étant un axe d'analyse. Donc assurer une bonne flexibilité revient à donner à ce contenu sémantique un double rôle.

3.1.4. Mesure textuelle

La modélisation reposant sur les concepts de fait et de dimension associés à des indicateurs numériques permet des analyses simples de documents textes.

Ces analyses reposent principalement sur le comptage de documents. Une bonne analyse de contenu des données textuelles doit prendre en compte les mesures textuelles.

3.1.5. Opérateur OLAP spécifiques aux données textuelles

Les opérateurs OLAP appliqués aux données simples ne sont pas adaptés aux données textuelles. Les fonctions d'agrégation numériques telles que somme, moyenne s'appliquent bien sur des données numériques, mais ne permettant pas d'agréger les données textuelles. Donc définir de nouveaux opérateurs OLAP s'appliquant sur les données textuelles s'avère nécessaire.

Malgré que les modèles extensifs ont permis d'effectuer des analyses multidimensionnelles sur les données textuelles, nous constatons qu'ils sont toujours limités et ne traitent que quelques aspects. De plus, ces modèles ne sont pas génériques et ne permettant pas de représenter n'importe quelle donnée de complexité liée à l'analyse de ce type de données, telle que la sémantique qui a été représentée par une dimension. Les autres aspects comme la prise en compte de la structure des données textuelles ainsi que la flexibilité d'analyse sur ces derniers, restent toujours non traités textuellement. Par contre, les modèles à nouveaux concepts ont permis de traiter d'autres problèmes d'analyse textuelle telle que la prise en compte de la structure ainsi que l'analyse du contenu des documents textes grâce à l'utilisation d'une mesure textuelle. Tandis que la flexibilité d'analyse reste toujours limitée, bien que le sujet d'analyse Fait n'est pas prédéfini au préalable dans les deux modèles (modèle en galaxie et à objets complexes). Nous constatons que le problème de flexibilité est assez complexe. Le contenu sémantique des données textuelles peut être vu comme étant une mesure d'analyse K-top keyword,Topic, comme il peut être considéré comme étant un axe d'analyse (hiérarchie de thèmes), les travaux actuels ne traitent pas ce double rôle.

Dans notre approche de modélisation nous nous intéressons en particulier aux trois aspects suivant : la prise en compte de la structure des données textuelles et de leur sémantique d'une part et la flexibilité d'analyse d'autre part.

Chapitre II
Méthodes d'analyses des données textuelles

1. Introduction

Les entrepôts de données classiques sont alimentés à partir des bases de données structurées, tandis que dans les entrepôts textes on parle d'une alimentation à partir d'un ensemble de documents textuels non structurés. Afin d'assurer cette alimentation on doit faire référence à des méthodes de recherches et d'extractions de l'information tel que la méthode TFIDF et des méthodes d'extraction des topic.

Alors dans ce chapitre nous allonsprésenter un ensemble de méthodes qui ont été utilisé dans le domaine des entrepôts textes.

2. Méthodes utilisées dans les entrepôts textes:

2.1. Méthode tfidf[Ger,83]

Term frequency-Inverse Document Frequency en anglaisest une méthode de pondération souvent utilisée en recherche d'information et en particulier dans la fouille de texte et dans les modèles à nouveaux concepts. Afin de permettre l'analyse des documents textes, Tournier a proposé deux fonctions d'agrégation pour les données textuelles : AVG-KW qui permet de regrouper des mots clefs en des mots clef plus généraux, à travers une ontologie de domaine et TOP-KWk, qui retourne une liste des termes les plus significatifs. Les termes sont pondérés par la méthode Tf Idf, les K termes avec les plus grands poids sont retournés.

Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus. Des variantes de la formule originale sont souvent utilisées dans des moteurs de recherche pour apprécier la pertinence d'un document en fonction des critères de recherche de l'utilisateur.

La justification théorique repose sur l'observation empirique de la fréquence des mots dans un texte qui est donnée par la Loi de Zipf. Si une requête contient le terme T, un document a d'autant plus de chances d'y répondre qu'il contient ce terme : la fréquence du terme au sein du document (TF) est grande. Néanmoins, si le terme T est lui-même très fréquent au sein du corpus, c'est-à-dire qu'il est présent dans de nombreux documents (e.g. Les articles définis- le, la, les), il est en fait peu discriminant. C'est pourquoi le schéma propose d'augmenter la pertinence d'un terme en fonction de sa rareté au sein du corpus (fréquence du terme dans le corpus IDF élevée). Ainsi, la présence d'un terme rare de la requête dans le contenu d'un document fait croître le « score » de ce dernier.

2.1.1. Concepts de la méthode [Ber et Laf, 99]

La méthode est basée sur les calculs suivants qui sont la fréquence du terme et la fréquence inverse du document.

La fréquence d'un terme (term frequency) est simplement le nombre d'occurrences de ce terme dans le document considéré (on parle de "fréquence" par abus de langage).Des variantes ont été proposées (dont une application du logarithme sur le tf simple pour amortir les écarts).

La fréquence inverse de document (inverse document frequency) est, quant à elle, une mesure de l'importance du terme dans l'ensemble du corpus. Dans le schéma TF-IDF, elle vise à donner un poids plus important aux termes les moins fréquents, considérés comme plus discriminants. Elle consiste à calculer le logarithme de l'inverse de la proportion des documents du corpus qui contiennent le terme :

$$\boldsymbol{idf_i = \log \frac{|D|}{|\{d_j = t_j \in d_j\}|}}$$

Où

- $|D|$: nombre total de documents dans le corpus
- $|\{d_j = t_j \in d_j\}|$: nombre de documents où le terme t_i apparaît (c'est-à-dire$n_{i,j} \neq 0$)).

Finalement, le poids s'obtient en multipliant les deux mesures : $\boldsymbol{tfidf_{i,j} = tf_{i,j}.\, idf_i}$

2.1.2. Avantages [Ral, 06]

- Facile à implémenter.
- Donne d'assez bons résultats (pour la langue Anglaise notamment) ;
- On trouve plusieurs variantes pour les formules initiales.

En pratique, la méthode de représentation la plus utilisée pondère l'importance d'un terme à l'intérieur d'un document et son importance dans un corpus en représentation creuse.

2.2.3. Limitations [Ral, 06]

- Basé uniquement sur des fréquences ce qui peut induire à des faux positifs,
- N'a pas le même degré d'efficacité pour toutes les langues (notamment les langues très fléchies comme le Polonais par exemple).

2.2. Extraction de topic (sujet)

2.2.1. Définition d'un topic

Afin de construire un entrepôt texte Zhang et al ont proposé un modèle nommé Topic Cube qui étend le cube de données traditionnel en intégrant une hiérarchie de thèmes 'Topics' comme étant une dimension d'analyse.

En informatique, un topic correspond à un sujet défini dans un forum. Chez DTPC (District Tourism Promotion Councils), la catégorie « Matériel » par exemple peut abriter comme topic "Problème carte graphique" il permet d'attirer rapidement toutes les personnes désirant faire progresser les recherches ou voulant posséder des informations sur les problèmes de cartes graphiques. L'un des objectif de notre projet et d'étudier l'opération d'extraction de topic ou de sujet dans laquelle, il est question d'identifier, d'abord, les sujets importants dans un corpus puis dessiner les relations entre eux.

Certaines recherches procèdent par la segmentation en divisant simplement le corpus selon le sujet lors d'un changement. D'autres recherches tentent d'extraire les opinions positives et négatives en analysant les commentaires parus dans des discussions de forums sur tel ou tel produit par exemple. C'est pourquoi, une

indexation sémantique et à prévoir pour éviter ces erreurs et aider à réaliser les objectifs primaires à savoir l'extraction des sujets. En effet, plusieurs méthodes existent pour cette opération et nous allons dans ce qui suit tenter de parler de quelques une d'entre elles.

2.2.2. Méthodes d'extractions des topics

Pour extraire des topics plusieurs méthodes sont vu le jour et ces méthodes sont tous basés sur des calculs mathématiques soit sur le calcule matriciel en se basant des théorèmes qui sont utilisés sur des matrices ou sur les statistiques ces derniers sont assez utiliser et donne de très bon résultat, pour cela nous allons détailler quelque méthodes qui on était mise en œuvre.

2.2.2.1. Analyse sémantique latente [Roc, 06]

Analyse sémantique latente (LSA) est bien connue comme une technique qui répond en partie à ces questions. L'idée principale est de cartographier les vecteurs de comptage de grande dimension, telles que celles qui se posent dans les représentations d'espace vectoriel de documents textes, dans une moindre représentation en trois dimensions dans un soi-disant espace sémantique latente.

Comme son nom l'indique, le but de LSA est de trouver un mappage de données qui fournit des informations au-delà du niveau lexical et révèle les relations sémantiques entre les entités d'intérêt. En raison de sa généralité, LSA s'est avéré être un outil d'analyse précieux avec une large gamme d'applications. Pourtant, son fondement théorique reste dans une large mesure insatisfaisant et incomplet. Toutefois, étant donné que l'application la plus importante de LSA se situe dans l'analyse et la recherche des documents textes.

La LSA utilise une matrice qui décrit l'occurrence de certains termes dans les documents. C'est une matrice creuse dont les lignes correspondent aux « termes » et dont les colonnes correspondent aux « documents ». Les « termes » sont généralement des mots tronqués ou ramenés à leur radical, issus de l'ensemble du corpus. On a donc le nombre d'apparition d'un mot dans chaque document, et pour tous les mots. Ce nombre est normalisé en utilisant la pondération *tf-idf*, combinaison

de deux techniques : un coefficient de la matrice est d'autant plus grand qu'il apparaît beaucoup dans un document, et qu'il est rare pour les mettre en avant.

Cette matrice est courante dans les modèles sémantiques standards, comme le modèle vectoriel, quoique sa forme matricielle ne soit pas systématique, étant donné qu'on ne se sert que rarement des propriétés mathématiques des matrices. La LSA transforme la matrice des occurrences en une « relation » entre les termes et des « concepts », et une relation entre ces concepts et les documents. On peut donc relier des documents entre eux.

Etapes d'applications du LSA [Dee et Al, 90]

Cette organisation entre termes et concepts est généralement employée pour :

- La comparaison de documents dans l'espace des concepts (classification et catégorisation de documents, partitionnement de données).
- La recherche de documents similaires entre différentes langues, en ayant accès à un dictionnaire de documents multilingues.
- La recherche de relations entre les termes (résolution de synonymie et de polysémie).
- Etant donné une requête, traduire les termes de la requête dans l'espace des concepts, pour retrouver des documents liés sémantiquement (recherche d'information).
- Trouver la meilleure similarité entre petits groupes de termes, de façon sémantique (c'est-à-dire dans le contexte d'un corpus de connaissance), comme par exemple dans la modélisation de la réponse aux questionnaires à choix multiples (QCM).

La résolution de la synonymie et de la polysémie est un enjeu majeur en traitement automatique des langues :

- Deux synonymes décrivent une même idée, un moteur de recherche pourrait ainsi trouver des documents pertinents mais ne contenant pas le terme exact de la recherche.

- La polysémie d'un mot fait qu'il a plusieurs sens selon le contexte on pourrait de même éviter des documents contenant le mot recherché, mais dans une acception qui ne correspond pas à ce que l'on désire ou au domaine considéré.

2.2.2.2. Analyse sémantique latente probabiliste [Hof, 01]

L'analyse sémantique latente probabiliste (de l'anglais, Probabilistic latent semantic analysis : PLSA), aussi appelée indexation sémantique latente probabiliste, est une méthode de traitement automatique des langues inspirée de LSA. Elle améliore cette dernière en incluant un modèle statistique particulier. La PLSA possède des applications dans le filtrage et la recherche d'information, le traitement des langues naturelles, l'apprentissage automatique et les domaines associés. Elle fut introduite en 1999 par Thomas Hofmann, et possède des liens avec la factorisation de matrices positives. Comparée à l'analyse sémantique latente simple, qui découle de l'algèbre linéaire pour réduire les matrices des occurrences (au moyen d'une décomposition en valeurs singulières), l'approche probabiliste emploie un mélange de décompositions issues de l'analyse des classes latentes. On obtient ainsi une approche plus souple, fondée sur les statistiques.

Il a été montré que l'analyse sémantique latente probabiliste souffre parfois de sur apprentissage, le nombre de paramètres croissant linéairement avec celui des documents. Bien que PLSA soit un modèle génératif des documents de la collection, elle modélise effectivement directement la densité jointe P (mot, document), elle ne permet pas de générer de nouveaux documents, et en ce sens n'est pas un « vrai » modèle génératif. Cette limitation est levée par l'Allocation de Dirichlet latente (LDA).

Chaque document d'une collection D est représenté par une distribution de probabilité sur les K valeurs de la variable thématique latente $j \in A= \{j1,\ldots,jk\}$ où chaque valeur de j correspond à une distribution de probabilité sur l'ensemble des mots de la collection. Dans le processus génératif correspondant à ce modèle un document est d'abord choisi suivant la probabilité P(d), ensuite une thématique est générée avec une probabilité P (j,d), et finalement un mot w est émis suivant la probabilité P(w,j). Dans le cas où les partitions de documents sont confondues aux thématiques, l'algorithme PLSA peut être utilisé comme un

algorithme de clustering de documents. Le point de départ de l'analyse sémantique latente probabiliste est un modèle statistique qui a été appelé modèle d'aspect. Le modèle est un modèle aspect variable latente pour la co-occurrente de données qui associe une variable de classe non observée à chaque observation. Ce modèle est défini par le mélange :

$$P(w,d) = \sum_c P(c)P(d|c)P(w|c) = P(d)\sum_c P(c|d)P(w|c)$$

Comme quasi-totalité des modèles statistiques de variables latentes du modèle aspect introduit une hypothèse d'indépendance conditionnelle, à savoir que d et w sont indépendants conditionnées par l'état de la variable latente associée (la représentation graphique du modèle correspondante est représentée sur la figure 06 :

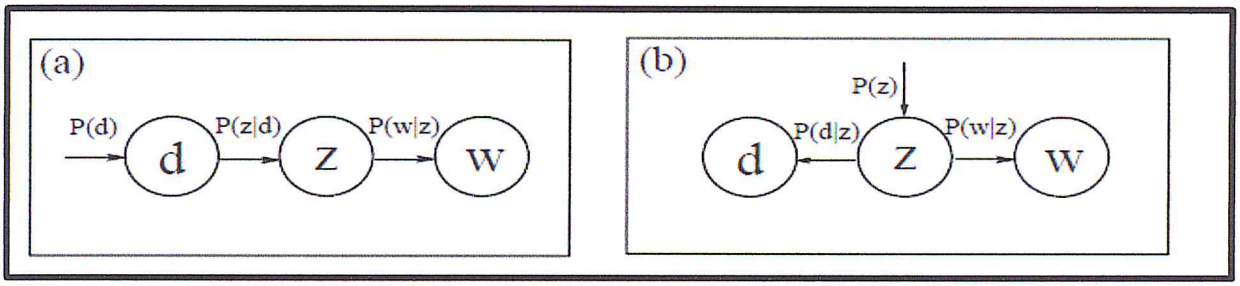

Figure 06 : automate de PLSA détaillé [Hof, 01].

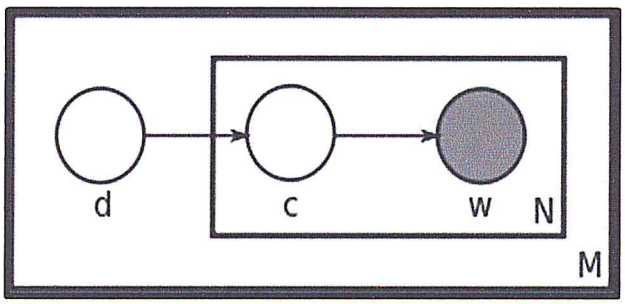

Figure 07: représentation générale de l'automate PLSA [Hof, 01].

D : est la variable de document, c'est un sujet tiré de la distribution de ce document sujet, P (c | d), et w est un mot élaboré à partir de la distribution des mots de cette fiche, P (w | c). Le d et w sont des variables observables, le sujet c'est une variable latente.

La procédure standard pour l'estimation du maximum de vraisemblance dans les modèles à variables latentes est l'Espérance-Maximisation (EM). EM alterne deux étapes couplées:

- Une attente (E) où l'étape probabilités a posteriori estcalculée pour les variables latentes.

- Une maximisation (M) l'étape, où des paramètres sont mises à jour. Calculs standards donnent l'équation E-étape.

2.2.2.3. Allocation de Dirichlet latente (LDA) [D. Ble and M. Jor, 03]

L'allocation de Dirichlet latente (de l'anglais Latent Dirichlet Allocation) ou LDAest un modèle génératif probabiliste permettant d'expliquer des ensembles d'observations, par le moyen de groupes non observés, eux-mêmes définis par des similarités de données.

Par exemple, si les observations sont les mots collectés dans un document textuel, LDA suppose que chaque document est un mélange d'un petit nombre de sujets ou thèmes (topics), et que la création de chaque mot est attribuable (probabilités) à l'un des sujets du document. LDA est un exemple de « modèle de sujet » et fut présenté initialement comme un modèle graphique pour l'analyse de sujets, par David Blei, Andrew Ng et Michael Jordan en 2002. Les applications de LDA sont nombreuses, notamment en fouille de données et en traitement automatique du langage naturel.

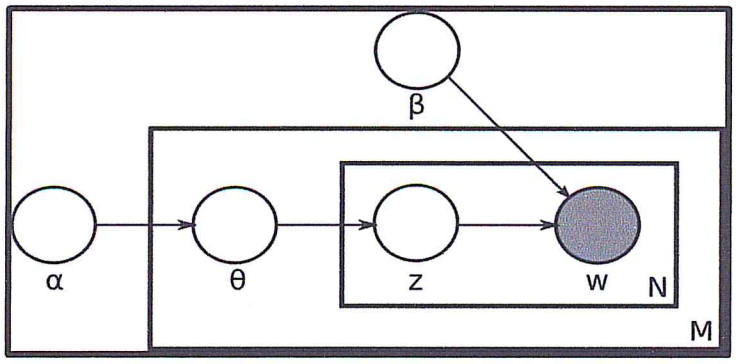

Figure 08 : Automate de l'Allocation de Dirichlet Latente [D. Ble et al, 03].

Les variables aléatoires :

- Mot est représenté par une variable aléatoire multinomialew.
- Le sujet est représenté par une variable aléatoire multinomiale z.

- Document est représenté comme variable aléatoire θ.

On a examiné brièvement la représentation graphique de l'LDA via simplex sujets:

- Chaque cornet du simplexe correspond à un sujet - une composante du vecteur *z*.
- Un document est modélisé comme un point du simplexe - une distribution multimodale sur les sujets.
- Un corpus est modélisé comme une distribution de Dirichlet sur le recto.

On présente dans ce qui suit, un scénario type qui montre l'utilisation de LDA dans les moteurs de recherche :

1) Grâce à la LDA, on extrait des thématiques d'un grand corpus de documents. On va avoir quelques milliers ou plus de topics. On prend ensuite une sélection de mots qui paraissent intéressants pour la ségrégation de documents. Chaque couple (mot, topic) est caractérisé par une quantité qui est la probabilité avec laquelle un mot d'un texte est employé avec une signification qui le rattache au topic.

2) Tous ces couples sont mis dans une matrice MOT ×TOPIC. Chaque mot est alors défini par le vecteur qui donne toutes les probabilités d'appartenance aux divers topics. Quand on a une requête de plusieurs mots, on peut moyenner les vecteurs pour avoir la probabilité d'appartenance de la requête aux topics.

3) Quand on a deux textes (ou un texte et une requête), on peut utiliser le cosinus de Salton, ou la distance de kullback Leibler entre les vecteurs associés pour mesurer la similarité.

3. Etude Comparative

Après avoir vu les différentes modèles d'extraction de topic, nous avons décidé de mener une petite étude comparative que nous avons résumé dans le tableau suivant :

	Avantage	Inconvénient
LSA	✓ Examine des problèmes de taille raisonnable (d'ordre 100-2000 documents) avec (5000-7000 termes). ✓ Peut-être modélisé dans un espace déterminé pour approximer les relations entre termes et documents. ✓ Extrait les similarités entre mot-mot, texte-texte ou mot-texte.	• L'ordre des mots et la syntaxe ne sont pas pris en compte • Le problème de polysémie et le rapprochement du vocabulaire des différents documents. • LSA s'adapte bien avec le problème de synonymie mais offre seulement une solution partielle au problème de polysémie.
PLSA	✓ La PLSA est une méthode générative qui se base sur l'introduction de variables cachées ou encore appelées variables latentes. ✓ Le modèle PLSA est intéressant car c'est le premier qui introduit la notion de thématique au sein des documents.	• PLSA réside dans sa théorie statistique induite par le modèle d'aspect. • La génération du document se fait par la probabilité d'appartenance du mot au topic, et non par la probabilité qu'un document contienne un

		mot d'un topic donné.
LDA	✓ LDA offrait la possibilité d'extraire des ensembles de mots fréquemment groupés dans des textes. ✓ Le modèle LDA a été conçu pour éviter que la distribution de probabilités qui sert au choix du topic soit dépendante des documents connus précédemment. ✓ Le LDA est l'un des concepts les plus prometteurs.	

4. Conclusion

L'extraction automatique d'information et plus généralement la problématique de la structuration et de la description automatique de l'information textuelle devient une fonctionnalité importante des systèmes d'informations, parce qu'il ne peut plus être question d'indexer à la main des volumes sans cesse croissants d'informations et parce que la masse d'information proposée à chaque lecteur devient telle qu'on a besoin d'outils d'aide à la lecture rapide.

Nous avons, dans ce chapitre, détaillé les méthodes d'analyse de données textes tel que la méthode tfidf et l'extraction des topic et on a constaté que la méthode LDA est la plus efficace. L'intérêt du modèle LDA est sur l'aspect génératif et d'inférer les thématiques à partir d'un corpus de documents.

Partie II

Conception et Implémentation

Chapitre I

Vers un modèle d'entrepôt texte

1. Introduction

La modélisation multidimensionnelle est aujourd'hui reconnue comme reflétant le mieux la vision des décideurs sur les données à analyser.Cependant, les modèles multidimensionnels classiques ont été pensés pour traiter des données numériques ou symboliques mais échouent dès lors qu'il s'agit de données textuelles. Pour réponde à cette problématique plusieurs travaux ont été élaboré. Ces travaux ont été classé selon **[Att et al,13]** en deux familles de modèles : (i) modèles extensifs et (ii) modèles à nouveaux concepts. D'après notre étude de ces différents travaux et en se basant sur l'étude comparative présentée dans **[Att et al,13]** nous avons choisis d'instancier le modèle d'objet complexe avec des données textuelles, notre choix se justifie par le fait que le modèle d'objets complexe est parmi les modèles les plus adaptés à l'analyse des données textuelles car il permet de faire des analyses sur de différents niveau de granularités textuelles et il offre aussi une bonne flexibilité d'analyse. Dans ce chapitre, nous allons présenter notre instanciation du modèle d'objet complexe avec des données textes, en définissons les différents concepts de base liés à notre modèle.

2. Du modèle d'objets complexes au modèle d'objets textes

2.1. Concepts de base

Nous avons présenté dans la partie état de l'art, les concepts de base liés au modèle d'objets complexes en détail, nous présentons dans cette partie l'adaptation des définitions de ces concepts par rapport aux données textuelles :

2.1.1. Objet texte

C'est un objet qui hérite de la classe objet complexe avec des spécifications propres à lui. Il peut être une entité abstraite ou concrète analysable en tant que sujet ou axe d'observation. Il est définit par :

- Attribut simple : attribut de class en UML ;
- Attribut complexe: composé d'un ou de plusieurs attributs simples ou complexes.

2.1.2. Objet Topic

C'est un cas particulier d'un objet complexe, son rôle est de représenter la sémantique liée aux objets textuels.Il est définit par :

- Attribut simple : attribut de class en UML ;
- Attribut complexe: composé d'un ou de plusieurs attributs simples ou complexes.

La figure suivante illustre la structure d'un objet texte représentant une publicationscientifique avec un exemple concretde publication, limité par quelques attributs :

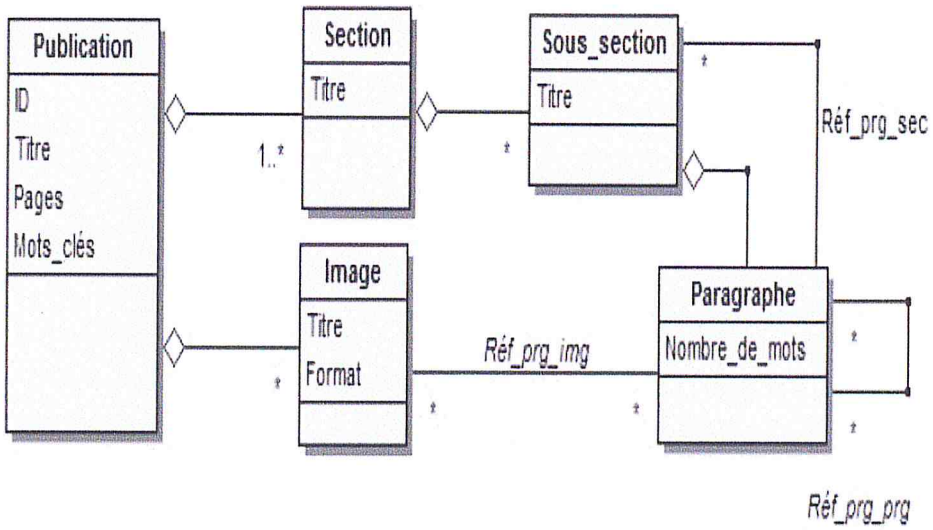

Figure 9 : Exemple d'un objet texte

2.1.3. Objet métadonnées

C'est un cas particulier d'un objet complexe Objet métadonnée contient des données qui décrivent un objet texte. Il est définit par

- Attribut simple : attribut de class en UML ;
- Attribut complexe: composé d'un ou de plusieurs attributs simples ou complexes.

2.1.4. Relation complexe

Représente les liens entre les objets de haut niveau, ils définissent les axes d'analyse de certains objets par rapport à d'autres (héritage, association). Lorsqu'il s'agit d'une relation de composition entre objets, l'objet composant est définit comme étant un attribut complexe de l'objet composite. La figure suivante nous illustre un exemple d'une relation complexe qui relie l'objet texte « Publication » et l'objet métadonnées « Auteur »

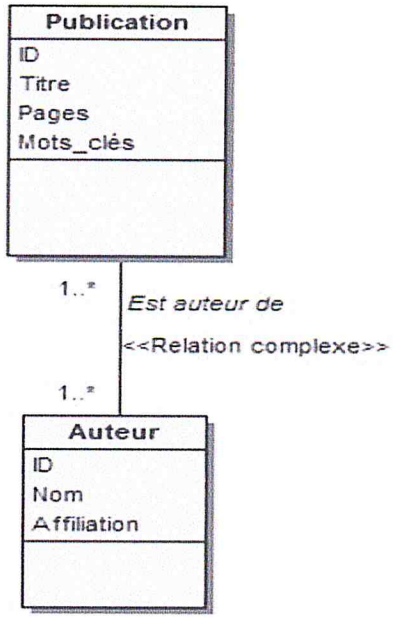

Figure 10 : Exemple d'un objet texte

2.1.5. Hiérarchie attributs

Relations intra-objets complexes. Il s'agit de relations qui organisent Certains attributs de l'objet en hiérarchies. La figure suivante nous illustre un exemple d'une hiérarchie d'attributs.

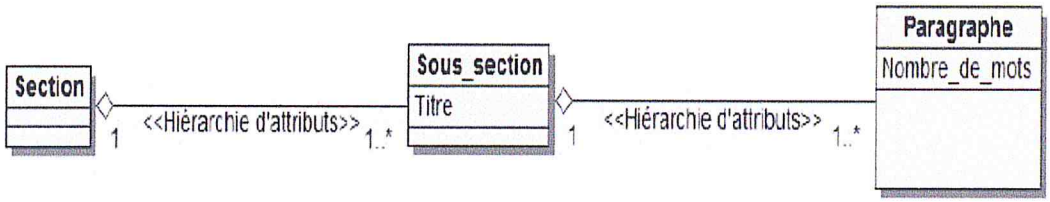

Figure 11. Exemple de hiérarchie d'attributs

2.1.6. Hiérarchie d'objets

Une hiérarchie d'objets est similaire à une hiérarchie d'attributs mais elle estdéfinie entre plusieurs objets textes plutôt qu'entre les attributs d'un seul objet. Elle permet d'effectuer des opérations d'agrégation entre les objets textes. Unehiérarchie d'objets définit un ordre partiel entre certains objets du monde réel selonleur degré de granularité.

2.2. Modèle à trois niveaux

Dans notre instanciation du modèle d'objets complexes, nous retenons les trois niveaux présentés par ce dernier. Le modèle d'objet texte résultant est un modèle à trois niveaux tel que : le premier niveau est représenté par un diagramme de classe détaillé des faits candidats et des dimensions candidates. Dans le deuxième niveau, les classes décrivant le même objet texte sont regroupées en un seul package, pour fournir à la fin un diagramme de packages décrivant des objets textes. Le troisième niveau est représenté par un diagramme de packages qui résulte de la projection d'un package objet complexe du deuxième niveau comme étant un objet fait et de lui associer un ensemble d'objets dimensions décrites par des objets textes liés à l'objet fait par des relations complexes. Chaque objet texte peut être défini grâce à l'opérateur de projection cubique comme étant un axe ou un sujet d'analyse.

Donc l'objet fait n'est pas prédéfini au préalable ce qui offre une bonne flexibilité d'analyse.Ce modèle permet aussi une analyse sur de différents niveaux de granularité de chaque objet texte. La figure suivante nous illustre les trois niveaux présents sur notre instanciation du modèle d'objets complexes, le premier niveau est représenté par la figure c, le deuxième par la figure b et le troisième par la figure a :

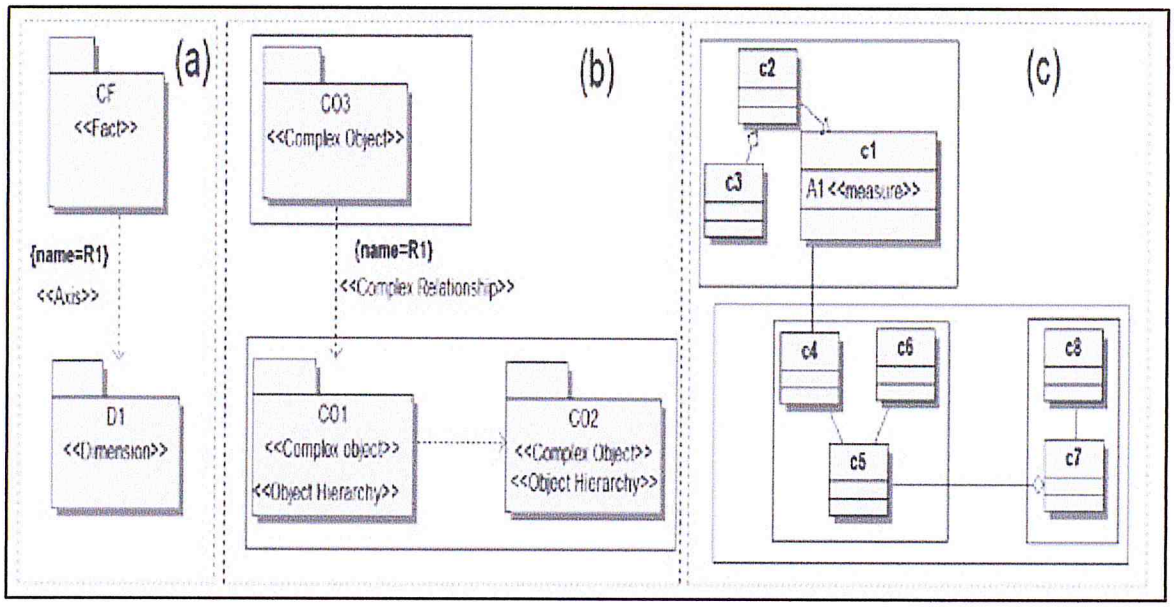

Figure 12: Modèles à trois niveaux

3. Conclusion

Nous avons présenté dans ce chapitre notre adaptation du modèle d'objets complexes à l'analyse des données textuelles. Nous avons définit l'ensemble des concepts que nous allons utiliser pour créer notre entrepôt de textes.

Dans ce chapitre, nous avons présenté notre instanciation du modèle d'objet complexe avec les données textes, et nous avons définis les concepts de bases du modèle.

Chapitre II

Modélisation d'entrepôt

UML est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG). La dernière version diffusée par l'omg est UML 2.4.1 depuis aout 2011.

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

- Activité d'un objet/logiciel
- Acteurs
- Processus
- Schéma de base de données
- Composants logiciels
- Réutilisation de composants

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie de code, par exemple Java, à partir des divers documents réalisés.

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système :

- Vue logique : c'est la définition du système vu de l'intérieur. Elle explique comment peuvent être satisfaits les besoins des acteurs.
- Vue d'implémentation : cette vue définit les dépendances entre les modules.
- Vue des processus : c'est la vue temporelle et technique, qui met en œuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation, etc.
- Vue de déploiement : cette vue décrit la position géographique et l'architecture physique de chaque élément du système .

2.2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation présenté au-dessus permet de recenser les grandes fonctionnalités fournit par notre système, et montre l'interaction entre l'administrateur et les principaux fonctionnalités offertes par notre système.

2.2.1. Diagramme de cas d'utilisation « Création d'un entrepôt texte »

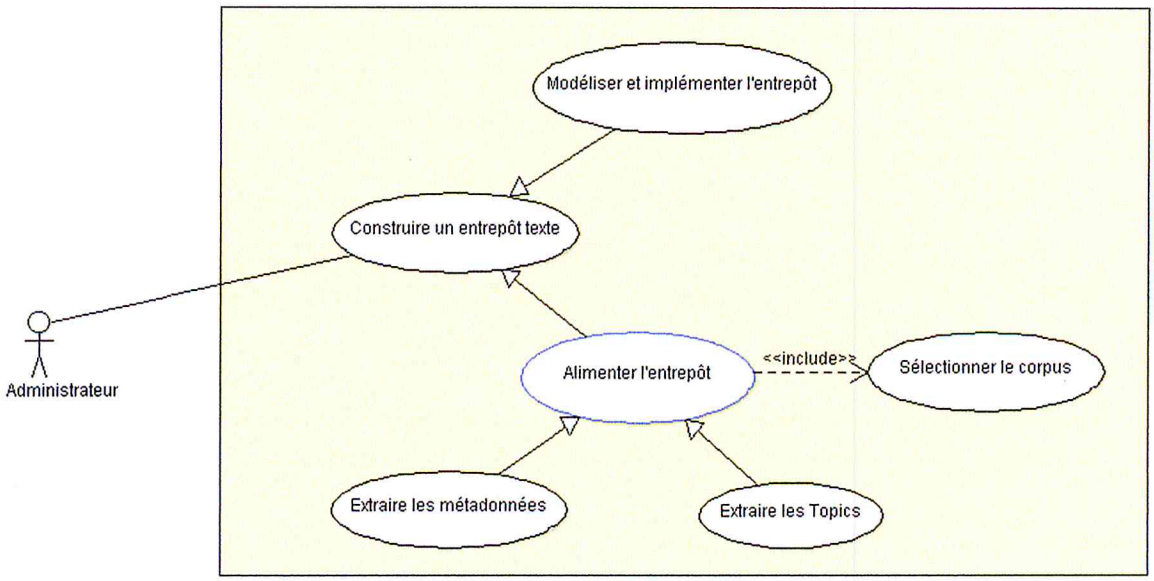

Figure 13 :Diagramme de cas d'utilisation 1« Création d'un entrepôt texte »

Description des acteurs

Nom de l'acteur	Rôle
Administrateur	Acteur principale qui prend en charge toutes les phases du processus de construction de l'entrepôt de texte

Description textuelle des cas d'utilisation

Cas d'utilisation	Description
Construire un entrepôts texte	Afin de permettre une analyse multidimensionnelle des documents textuelles, l'administrateur doit procéder à créer un entrepôt texte .
Modéliser et implémenter l'entrepôt	L'administrateur modélise son entrepôt texte en se basant sur les concepts de base liés au modèle d'entrepôt texte et implémente son entrepôt sur un serveur de base de données.
Alimenter l'entrepôt	Ce cas d'utilisation représente la phase de chargement des données dans l'entrepôt. Dans cette phase on fait appel à des méthodes de recherche d'information.
Extraire les métadonnées	Ce cas d'utilisation représente un cas spécifique du cas « alimenter l'entrepôt ». Les différentes informations qui décrivant les caractéristiques d'un document sont extraites afin d'alimenter l'entrepôt.
Extraire les Topics	Ce cas d'utilisation représente un cas spécifique du cas « alimenter l'entrepôt ». Les topics (sujet) présents dans chaque document sont extraits en faisons appel à la méthode LDA
Sélectionner le corpus	Sélectionner le chemin du corpus que nous souhaitons analyser.

2.2.2. Diagramme de cas d'utilisation « Construction de cube texte »

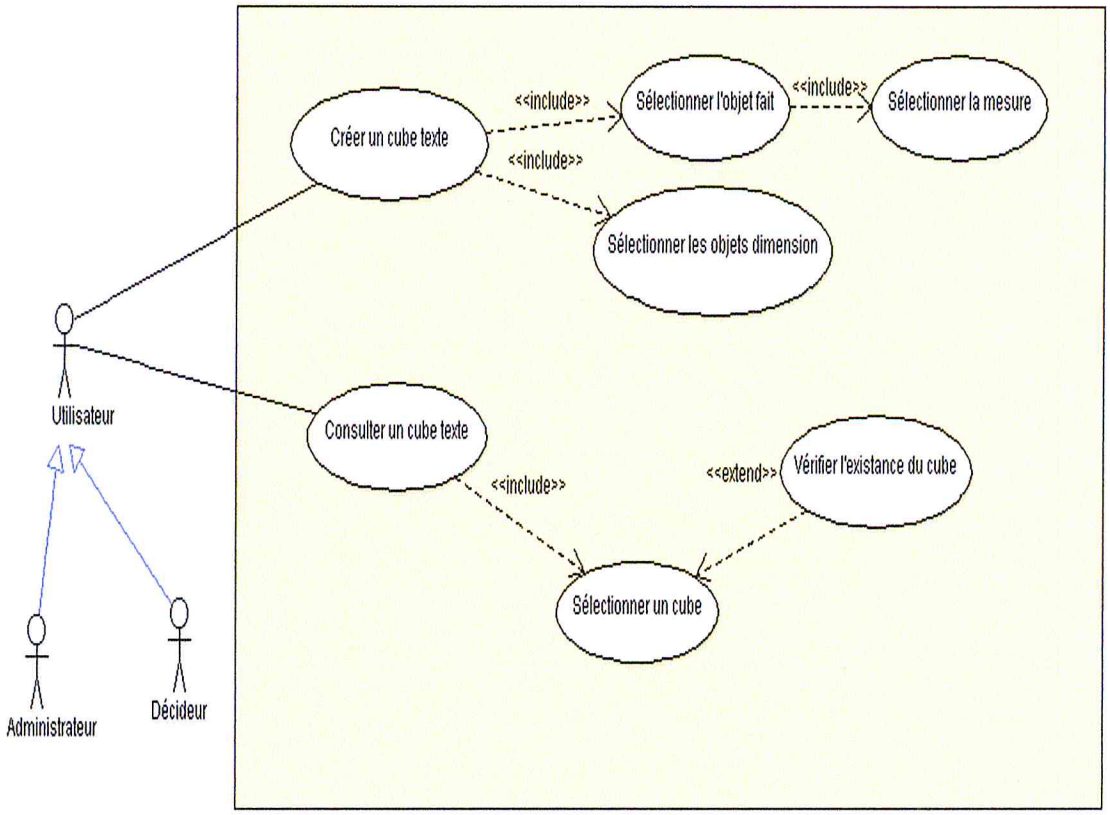

Figure 14 :Diagramme de cas d'utilisation 2 « Construction de cube texte »

Description des acteurs

Nom de l'acteur	**Rôle**
Administrateur	Acteur principale qui prend en charge toutes les phases du processus de création du cube texte
Décideur	Acteur qui souhaite effectuer des analyses sur l'entrepôt

Description textuelle des cas d'utilisation

Cas d'utilisation	Description
Cree un cube texte	Ce cas d'utilisation décrit l'action de la construction d'un cube texte à partir de l'entrepôt.
Consulter un cube texte	Consulter un cube de texte existe.
Selectionner l'objet fait	Pour créer la table de fait, l'utilisateur doit sélectionner à partir l'entrepôt un objet auquel il va associer le rôle de « l'objet fait »
Selectionner les objets dimentions	l'utilisateur doit sélectionner à partir de l'entrepôt l'ensemble des dimensions
Selectionnée les mesures	l'utilisateur doit sélectionner pour chaque objet fait une mesure d'analyse
Selectionnée un cube	Sélectionner le cube de texte que l'utilisateur souhaite consulter
Verifié l'existance du cube	Vérifier si le cube que l'utilisateur souhaite consulter est déjà présent

2.3. Diagramme de classe

Afin de présenter le premier niveau qui décrit notre entrepôt d'objets textes, nous avons utilisé le diagramme de classe UML. Dans ce premier niveau, chauque objet texte est décrit par une classe UML. Ce diagramme de classe détaille les faits candidats et des dimensions candidates.

Notre entrepôt de texte dans le revues de presse est decrit à ce niveau par le diagramme suivant :

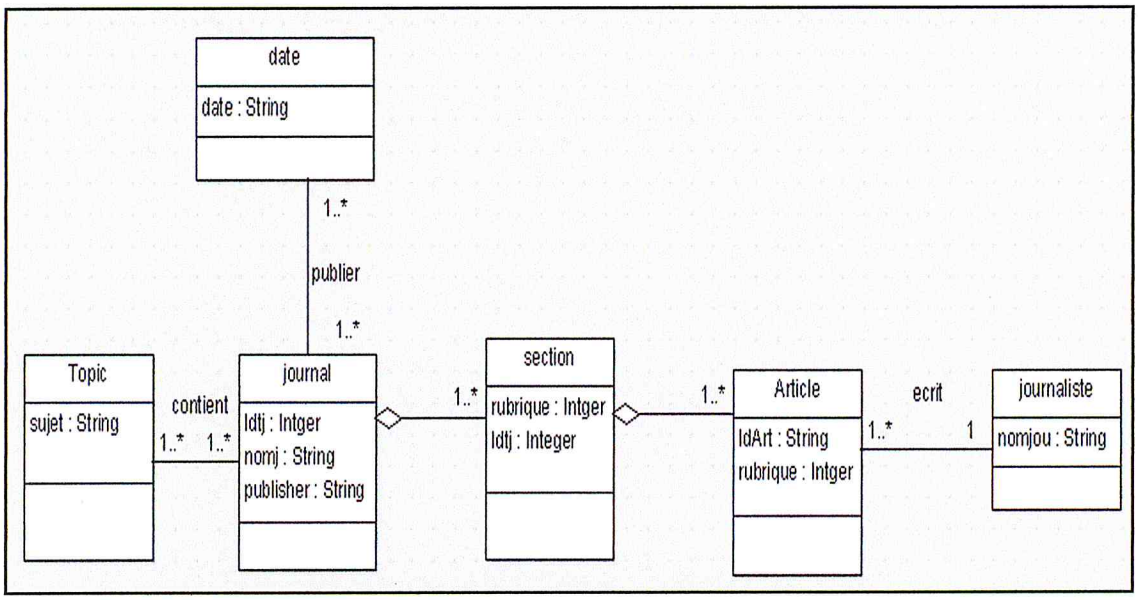

Figure15 :Diagramme de classe de l'exemple

2.4. Diagramme de package [w1]

Un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, le diagramme de paquetage sert à représenter les dépendances entre paquetages, c'est-à-dire les dépendances entre ensembles de définitions.

Dans ce niveau, les classes décrivant le même objet complexe sont regroupées en un seul package, pour fournir à la fin un diagramme de packages décrivant des objets complexes.

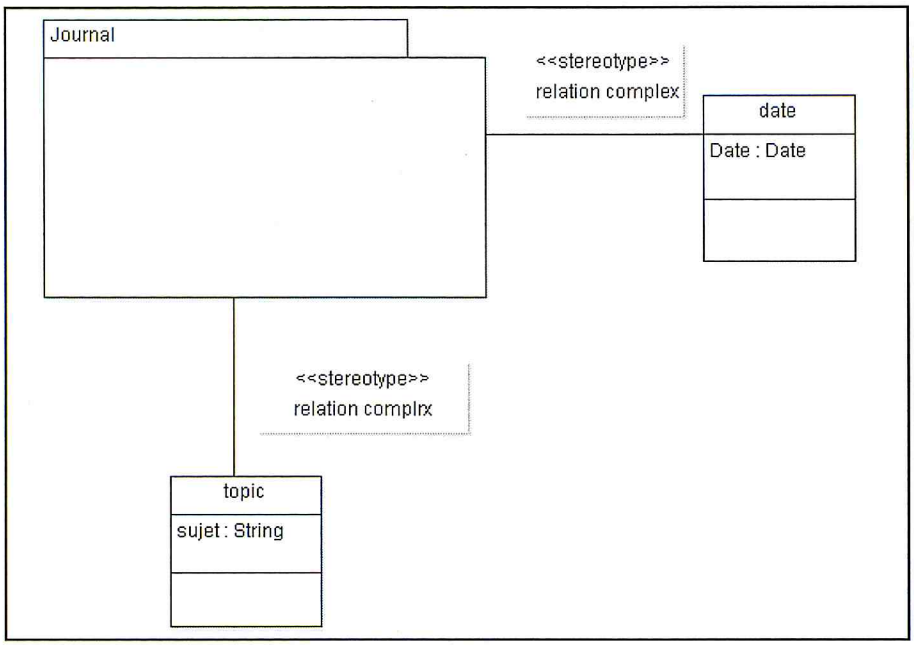

Figure 16:Diagramme de package de l'exemple.

2.4.1. Modèle d'objets textes

Afin de présenter le deuxième niveau qui décrit notre entrepôt d'objets textes, nous avons utilisé le diagramme de package UML. Dans ce deuxième niveau, les classes décrivant le même objet texte seront regroupé dans un seul package, qui va représenter les objets texte .

Notre entrepot de texte dans le revues de presse est decrit à ce niveau par le diagramme de package suivant :

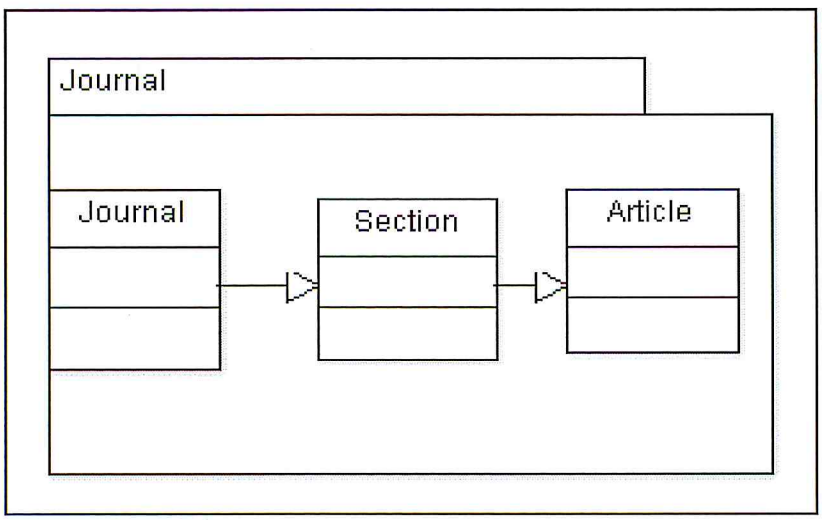

Figure 17:Diagramme de package journal

2.4.2. Modèle multidimensionnelle

Afin de présenter le troisième niveau qui décrit notre entrepôt d'objets textes, nous avons utilisé le diagramme de package UML. Dans ce deuxième niveau. Chauqe objet fait et objet dimension est décrit par un package UML.

Notre entrepot de texte dans le revues de presse est decrit à ce niveau par le diagramme de package suivant :

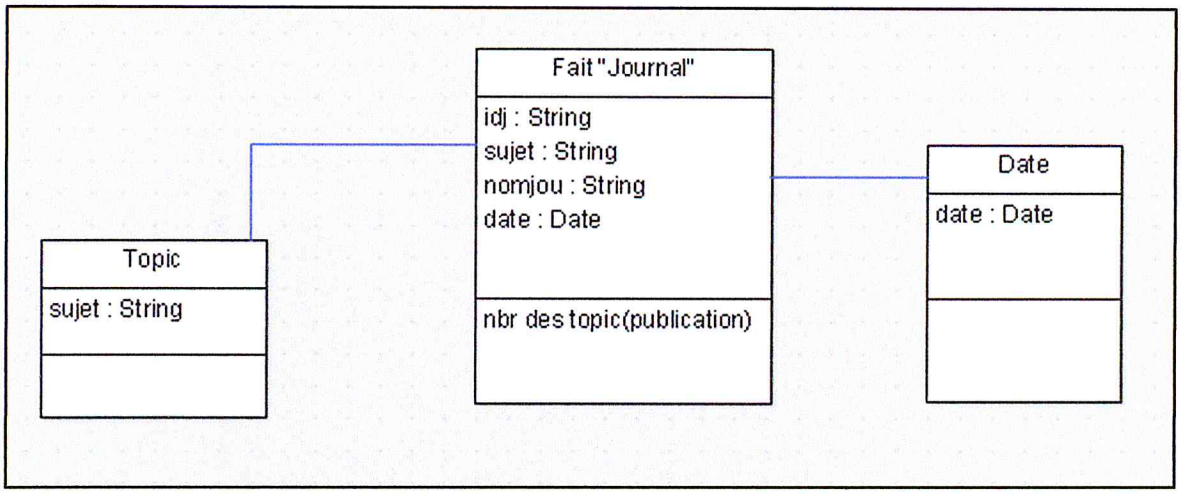

Figure 18:Diagramme de niveau multi dimensionnel de journal.

3. Conclusion

L'ensemble des diagrammes présentés dans ce chapitre nous ont été très utiles pour mieux comprendre le but de notre travail ainsi que de l'aborder d'une manière correcte. Nous avons présenté une solution d'entrepôt texte adapté à l'analyse des revues de presse, nous avons définis aussi les différents concepts du modèle.

Chapitre III

Implémentation

1. Introduction

Dans cette partie, nous allons introduire les aspects techniques liés à l'implémentation de notre solution logicielle, en expliquant les différents outils utilisés dans l'élaboration du système, à savoir l'IDE utilisé pour le développement ainsi que le SGBDR utilisé.

Plus loin dans ce chapitre, nous allons décrire les buts désirés de l'application et aussi les différentes interfaces qui la compose et les différents résultats qui sont générés.

2. Outils utilisés

Après avoir présenté les besoins fonctionnels et techniques et les grandes lignes concernant la modélisation de la solution logicielle que nous avons mis au point, nous allons à présent parler de la réalisation de cette dernière. Nous présenterons alors, dans un premier lieu, l'environnement de développement (langages et outils) ensuite, le diagramme d'accessibilité général du système et enfin, nous présenterons quelques captures d'écran.

2.1. Langage de programmation Java [W3]

Le langage Java est un langage de programmation informatique orienté objet créé par JamesGosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'UNIX, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications... C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à

la fois les plus déroutants, tel que l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.)

Java permet de développer des applications client-serveur. Côtéclient, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

2.2. IDE eclipse[W4]

Eclipse IDE est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plug-in (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

Plusieurs logiciels commerciaux sont basés sur ce logiciel libre, comme par exemple IBM Lotus Notes 8, IBM Symphony ou Websphere Studio Application Développer.

Le projet Eclipse est organisé en un ensemble cohérent de projets logiciels distincts, sa spécificité tenant à son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi) : toutes les fonctionnalités de

l'atelier logiciel doivent être développées en tant que plug-in bâti autour de l'IDE Eclipse Platform.

Eclipse recouvre donc notamment également à cet effet tout un Frameworkde développement logiciel fournissant des briques logicielles à partir desquelles développer tous ces outils. C'est la raison pour laquelle Eclipse est présenté dans la littérature tout autant comme un EDI ou comme un Framework.

2.3. MySQL [W5]

MYSQL est un Système de gestion de bases de données relationnelles (SGBDR) sous licence GNU très utilisé pour mettre en ligne des bases de données. Il permet d'entreposer des données de manière structurée (Base, Table, Champs, Enregistrements). Le noyau de ce système permet d'accéder à l'information entreposée via un langage spécifique le SQL.

3. APIs utilisées

Durent notre travaille nous avons utilisé des APIs, qui sont des packages java sous forme de jar, qui ont des fonctions pré implémente, les différentes APIs utilisées sont citées comme suit avec leurs détailles et leurs butes .

3.1. API Mallet [W7]

MALLET est un package basé sur Java pour le traitement statistique du langage naturel, classification du document, le regroupement, la modélisation du sujet, l'extraction de l'information, et d'autres applications machines à apprendre à texte.

MALLET inclut des outils sophistiqués pour la classification de documents : une grande variété d'algorithmes (y compris Naïve Bayes, entropie maximale, et les arbres de décision), et le code pour évaluer la performance de classificateur à l'aide de plusieurs indicateurs couramment utilisés, cette API nous a aidée dans la partie de fouilles de données exactement dans l'extraction du sujet à partir d'un ensemble de documents textes .

3.2. API Textwise [W6]

TextWise a été à la pointe de traitement du langage naturel et d'analyse sémantique depuis près de vingt ans. Fondé en 1994 pour fournir des solutions uniques pour le renseignement américain, la finance et les communautés de propriétés intellectuelles, TextWise a été élargi pour soutenir un large éventail d'industries axées sur l'information, y compris CRM, e-commerce, les médias sociaux et le marketing.

En utilisant cette api web car elle fonction avec une connexion web on peut avoir des intitulés pour chaque sujet extrait et cella selon des contextes réelles en prenant en charge même les nouveaux mots apparu dans la langue anglaise.

3.3. API StandFordCore [W9]

StanfordCore fournit un ensemble d'outils d'analyses du langage naturel, leurs parties du discours, qu'ils soient noms de sociétés, les gens, ...etc, normaliser les dates, heures et numériques quantités, et marquer la structure des phrases en termes de phrases et les dépendances de mots, et indiquent quelles phrases nominales se réfèrent aux mêmes entités. StanfordCore est un cadre intégré, ce qui rend très facile à appliquer un tas d'outils d'analyses de la langue à un morceau de texte. A partir de texte, vous pouvez exécuter tous les outils sur elle avec seulement deux lignes de code. Ses analyses fournissent les blocs de construction fondamentaux pour les applications de compréhensions du texte spécifiques à un domaine de niveau supérieur. Le but de ce projet est de permettre aux gens d'obtenir rapidement les annotations linguistiques complets de textes en langage naturel. Il est conçu pour être très flexible et extensible. Avec une seule option, vous pouvez changer les outils qui devraient être activés et qui doivent être désactivés.

3.4. API Apache Http Client [W10]

Le projet HttpClient communes est maintenant en fin de vie, et ne sont plus en cours d'élaboration. Il a été remplacé par le HttpComponents Apache projet dans son HttpClient et HttpCore modules, qui offrent une meilleure performance et une

plus grande flexibilité. L'Hyper-Text Transfer Protocol (HTTP) est sans doute le protocole le plus important utilisé sur l'internet aujourd'hui. Les services Web, les appareils compatibles réseaux et la croissance de l'informatique de réseau continuent d'étendre le rôle du protocole HTTP au-delà des navigateurs Web pilotées par l'utilisateur, tout en augmentant le nombre d'applications qui nécessitent un soutien HTTP.

Bien que le java.net paquet fourni les fonctionnalités de base pour accéder aux ressources via HTTP, il ne fournit pas toute la flexibilité ou la fonctionnalité requise par de nombreuses applications. Le Jakarta Commons *HttpClient* composante vise à combler cette lacune en fournissant une mise à jour efficace et riche en fonctionnalités paquet mise en œuvre du côté client des normes et recommandations HTTP les plus récentes.

Il y a beaucoup de projets qui utilisent *HttpClient* pour fournir les fonctionnalités HTTP de base. Certains d'entre eux sont open source avec des pages du projet que vous pouvez trouver sur le web tandis que d'autres sont fermés source que vous n'auriez jamais voir ou entendre parler. La licence Open Source Apache offre une flexibilité maximale pour la source et la réutilisation binaire.

3.5. API Jexel [W11]

Excel API Java est une mature, open source Java API permettant aux développeurs de lire, d'écrire et Modifié Excel dynamiquement. Maintenant les développeurs Java peuvent lire feuilles de calcul Excel, les modifier avec une API simple et pratique, et écrire les modifications apportées à un flux de sortie (par exemple disque, HTTP, base de données, ou n'importe quelle prise).

Tout système d'exploitation qui peut exécuter une machine virtuelle Java (ie, pas seulement Windows) permet d'offrir des tableurs Excel. L'API peut être appelée depuis une servlet, donnant ainsi accès à des feuilles de calcul Excel sur internet et les applications Web intranet.

4. Interfaces

4.1. Interface de l'application

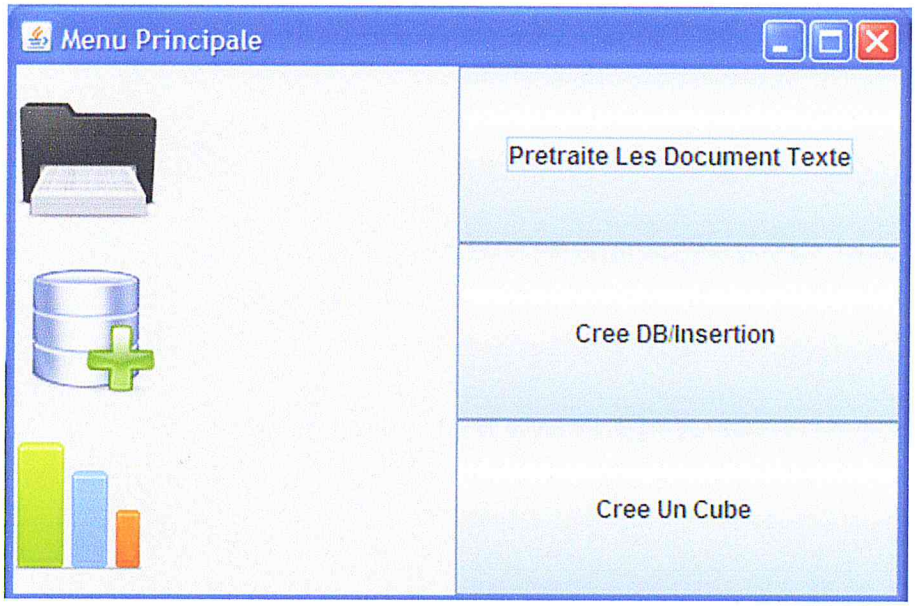

Figure 19 : Menu principale

Le menu principal qui va aider l'utilisateur a appliqué les options de son choix pour avoir une analyse voulu.

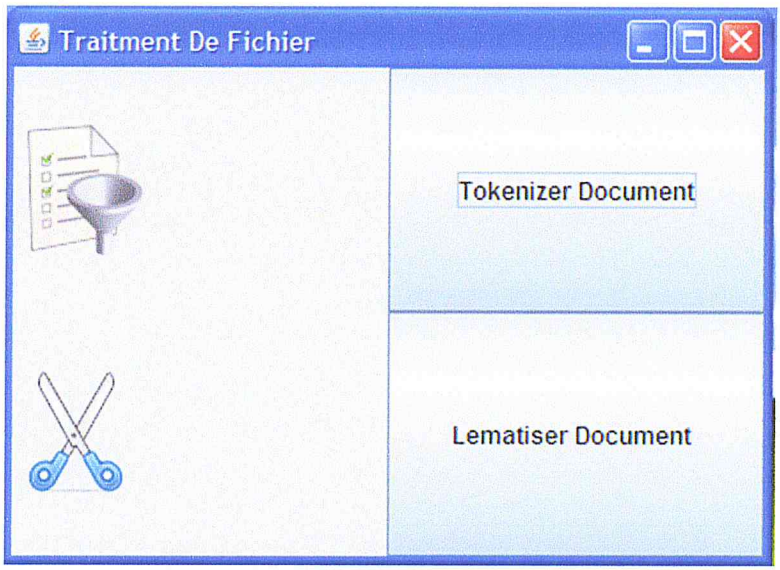

Figure 20 : Traitement de fichier.

Cette interface représente l'interface de traitement de fichier, elle se compose de deux boutons l'un pour Tokenization et l'autre pour lemmatiser le document.

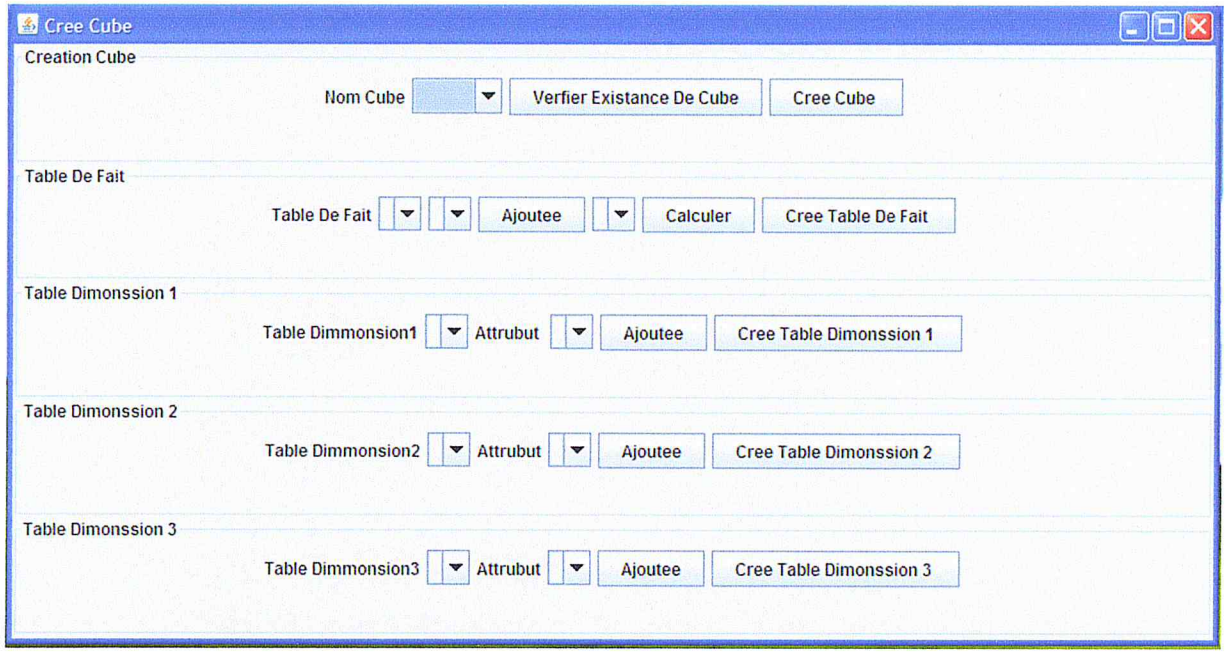

Figure 21 : Création du cube

Notre application offre aussi une interface de création du cube, l'utilisateur aura la possibilité de crée le cube il suffirait juste d'entrer le nom du cube, il va sélectionner la table de fait ensuite les tables de dimensions, et ajouter les attributs.

5. Conclusion

Dans cette étape de notre projet, nous avons présenté les différents outils utilisés pour implémenter notre modèle ainsi les différents interfaces implémentées afin de construire l'entrepôt de donnée texte, et nous avons effectués des analyses multidimensionnelles sur les données textes.

Conclusion Générale

Conclusion

L'objectif de ce travail était deconstruire un entrepôt textuel,qui nous permettra d'effectuer des analyses multidimensionnelles sur les données textes afin de prendre profit des informations qu'elles contiennent.

Durant ce projet, nous avons étudié les différentes techniques de l'intelligence artificielle et du traitement automatique de la langue ce qui a permis de choisir les meilleures approches pour concevoir et développer une application qui permet d'effectuer des analyses multidimensionnelles sur des données textuelles de l'entrepôt de données.

Ce projet nous a donné l'occasion d'acquérir de nouvelles connaissances dans le système décisionnelle et l'extraction des données textes, avoir de l'expérience en programmation java et se familiariser avec l'environnement java et MySQL.

En dépit de certaines difficultés que nous avons rencontrées, dont la nécessité de trouver un corpus pour réaliser notre travaille et aussi des méthodes efficaces pour permettre une très bonne extraction de sujet de façon optimale qui prenait un temps considérable ; et de choisir un modèle d'entrepôt textuelle ; nous avons tout de même pu réaliser le travail qui nous a été confié et ainsi réussi à atteindre notre but. Cependant, il restera toujours des améliorations à apporter, pour une meilleure performance.

Références bibliographique

Bibliographie

- [Att et al,13]:Attaf sarah et Benblidia nadjia,2013Modélisation Multidimensionnelle des données textuelles: où en sommes-nous?
- [Ber et Laf, 99] Berger, A &Lafferty, J. (1999) « Information Retrieval as Statistical Translation »
- [Bou,11] : Boukraa, D., O. Boussaid, F. Bentayeb, et D. Zegour (2011). Modèle multidimensionnel d'objets complexes. du modèle d'objets aux cubes d'objets complexes. Ingénierie des Systèmes d'Information 16.
- [CYR, 11] : CYRILLE HERBY 2011 Environnement java : Apprenez a programmé en java ISBN : 978-2-9535278-3-4 éditions livre du zéro
- [D. Ble and M. Jor, 03]:D. Blei, A. Ng, and M. Jordan (2003). Latent Dirichlet allocation. Journal of Machine Learning Research.
- [Dee et Al,90]: Deerwester et Al, 1990 Indexing by latent semantic analysis.
- [Ger,83]: Gerard salton.« Introduction to modern information retrieval », 1983
- [Gri, 12] :Chantal Gribaumont2012 Administrez vos bases de données avec MySQL.
- [Hof, 01]: T. Hofmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis.Machine Learning, 42(1/2):177–196, 2001.
- [Kim,96] : Kimball, 1996 The data warehouse toolkit: Practical Techniques for Building Dimensional Data Warehouses,. John Wiley and Sons.
- [Lan et Lah ,98] :Landauer, T. K., Foltz, P. W., & Laham, D. (1998).Introduction to Latent Semantic Analysis.
- [Lau, 08] : Laugie en 2008.java et Eclips : développez une application java.Ediciones ENI.268p.
- [Mul et Gae, 05] : Muller et Gaetner en 2005.Modelisation objet avec UML.514p.
- [Ral, 06] : Liva Ralaivola « Représentations vectorielles de textes » 19 décembre 2006.
- [Roc, 06] : Roche M., Chauché J., "LSA : les limites d'une approche statistique" FDC'2006.

- [Tou,07] : Tournier,2007 Analyse en ligne (OLAP) de documents. Thèse de doctorat, Université Toulouse III . Paul Sabatier.
- [Vol,10]:SvitlanaVolkova."Independent Project Final Report CIS 798"
- [Zha et al,09] : Zhang, D., C. Zhai, et J. Han (2009). Topic cube: Topic modeling for olap on multidimensional text databases. SDM '09: Proceedings of the 2009 SIAM International Conference on Data Mining, Sparks, NV, USA, 1124–1135.
- [Zhan et al,11] : Zhang, D., C. Zhai, et J. Han (2011). Mitexcube:microtextcluster cube for online analysis of text cells. The NASA Conference on Intelligent Data Understanding (CIDU), 204–218.

- [W1] :http://laurent-audibert.developpez.com/Cours-UML/html/index.html
- [W2]:http://grim.developpez.com/articles/concepts/bi- intro/#LV-C-2
- [W3] :www.mentalworks.fr/glossaire-web/J/definitions-internet.html
- [W4] :perso.limsi.fr/martin/ens/s3/java/Eclipse.pdf
- [W5] :Sun acquires MySQL [archive], blogs.mysql.com
- [W6] :http://www.textwise.com/api/documentation/introduction
- [W7] :http://programminghistorian.org/lessons/topic-modeling-and-mallet
- [W8] : Wikipédia.

- [W9]: http://nlp.stanford.edu/software/stanford-corenlp-full-2013-04-04.zip
- [W10]:http://hc.apache.org/httpclient-3.x/
- [W11]:http://jexcelapi.sourceforge.net/

Annexe

I. La méthode TF-IDF « exemple »[W8] :

Afin de bien assimilé le fonctionnement de la méthode tf-idf, nous avons décidez d'en présenter un exemple concret. Soit les trois documents représentésdans le tableau suivant :

Document 1	**Document 2**	**Document 3**
Son nom est célébré par le bocage qui frémit, et par le ruisseau qui murmure, les vents l'emportent jusqu'à l'arc céleste, l'arc de grâce et de consolation que sa main tendit dans les nuages.	À peine distinguait-on deux buts à l'extrémité de la carrière : des chênes ombrageaient l'un, autour de l'autre des palmiers se dessinaient dans l'éclat du soir.	Ah ! le beau temps de mes travaux poétiques ! les beaux jours que j'ai passés près de toi ! Les premiers, inépuisables de joie, de paix et de liberté ; les derniers, empreints d'une mélancolie qui eut bien aussi ses charmes.

L'exemple porte sur le document 1 (soit d_1) et le terme analysé est « qui » (soit t_1 = qui). La ponctuation et l'apostrophesontignorées.

- Calcul de la valeur*tf*

$$tf_{1,1} = \frac{n_{1,1}}{\sum_k n_{k,1}} = \frac{2}{38}$$

Détails du calcul : la plupart des termes apparaissent une fois (21 termes), *l* apparaît 3 fois et *arc*, *de*, *et*, *le*, *les*, *par* et *qui* (2 fois). Le dénominateur est donc 3 + 7*2 + 21 = 38. Cette somme est le nombre de mots dans le document.

- Calcul de la valeur*idf*

Le terme « qui » n'apparaît pas dans le deuxième document. Ainsi :

$$idf_1 = \log \frac{|D|}{|\{d_j : t_1 \in d_j\}|} = \log \frac{3}{2}$$

- **Calcul du Poids final**

On obtient :

$$tfidf_{1,} = \frac{2}{38} . \log \frac{3}{2} \approx 0{,}0092$$

Pour les autresdocuments:

$$tfidf_{1,2} = 0. \log \frac{3}{2} = 0$$

$$tfidf_{1,3} = \frac{1}{40} . \log \frac{3}{2} \approx 0{,}0044$$

Ainsi, le premier document apparaît ainsi comme « le plus pertinent ».

II. LSA « exemple » : [Lan et Lah ,98]

La première étape dans LSA est de représenter le texte sous forme d'une matrice :
A= [aij], i : 1..m et j : 1..n

Les lignes sont représentées par les termes ou mots et les colonnes représentent les contextes (paragraphe, document, phrase). Chaque cellule contient une fréquence aij qui représente le nombre d'occurrences du terme i dans le contexte j.
Une fois la matrice est établie, on procède à une normalisation qui consiste à introduire une transformation de la forme Log entropie ou TF-IDF.
La technique de log-entropie est une technique de pondération du terme, s'explique par le fait qu'un mot apporte de l'information s'il réduit l'entropie, de plus cette étape permet de pondérer chaque mot et d'estimer son importance dans le contexte.
La formeglobale de cette transformation est :

$$X(i,j) = L(i,j) * G(i)$$

$$Telque: \quad L(i,j) = \log_2(TF(i,j) + 1)$$

$$G(i) = 1 - E(i)$$

$$E(i) estl'entropiedutermei: \quad E(i) = -\sum_{j=1}^{m} \frac{P_{ij} \log_2(P_{ij})}{\log_2(n)}$$

$$P_{ij} = \frac{TF(i,j)}{\sum_j TF(i,j)}$$

Une autre technique de normalisation particulièrement utilisée est la règle TF-IDF « TermFrequency –Inverse Documents Frequency), dans ce cas la présence des mots communes dans tous les contextes ajoute du bruit, générant une faible distinction des contextes. Pour cela l'introduction d'un correctif qui relativise les poids des mots fréquents dans tous les classes. Pour ce faire des poids faibles sont affectés aux termes fréquents dans de nombreux documents. Ceci permet de privilégier les termes caractérisant un document. La forme générale est:

$$X(i,j) = TF(i,j) * IDF(i)$$

$$telque: \quad IDF(i) = \log_2\left(\frac{n}{DF(i)} + 1\right)$$

n: est le nombre de documents, DF(i): est le nombre de documents qui contiennent le terme i, TF (i,j) : est le nombre d'occurrence du mot i dans le document j.

Dans l'auteur précise l'intérêt de la règle TF-IDF, surtout l'effet de la pondération et son impact sur les résultats obtenus.

Une fois la normalisation effectuée, on obtient une matrice A' qui va être décomposée

en valeurs propres singulières « SVD : Singular Values Decomposition», c'est à- dire on obtient une décomposition de la forme suivante :

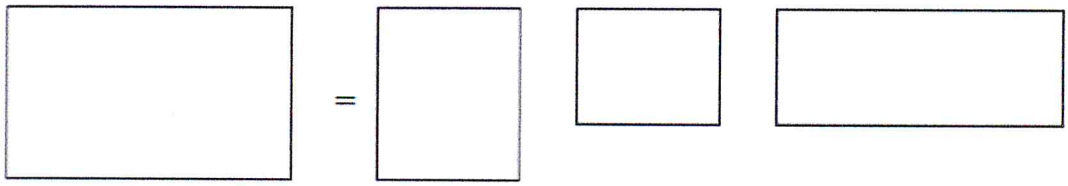

Fig.7 Decompositon de A′ en SVD

On obtient les caractéristiques suivantes : D est une matrice diagonale, U et Vt matrices orthogonales:

$U.U^t = IetV.V^T$

L'étape qui suit est la diagonalisation, qui va se limiter de prendre que les k premières valeurs propres de la matrice D, c'est à dire qu'on va éliminer toute la partie restante de D et ceux qui lui correspond dans U et Vt, (voir Fig.). On précise que les valeurs propres sont triées par ordre décroissant.

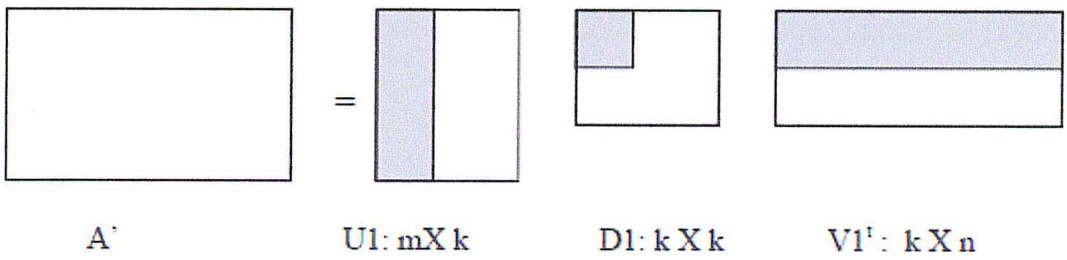

Fig.8 Diagonalisation de A′ et prise en compte des k premieres valeurs propres

Dans, la réduction d'ordre par le choix de k valeurs propres est critique, idéalement le but est d'avoir une valeur de k assez large pour couvrir toutes les données. En contrepartie, elle doit être petite sous contrainte que l'erreur commise soit moins importante, c'est ce qu'on appelle le compromis biais–variance. Toujours d'après, l'analyse LSA est de complexité d'ordre (mkn) (m: est le nombre de mots, k: le nombre de valeurs propres pris en compte et n: est le nombre de documents).

L'étape de la diagonalisation est suivie par l'approximation, cette dernière consiste

à réévaluer de nouveau le produit U1.D1.V1t et on obtient une matrice A'' qui représente l'image approximative de la matrice originale A. Cette étape est très importante, elle sert de base pour tirer la sémantique à base de similarité. La particularité de LSA d'après, est une technique statistique automatique pour l'extraction et l'inférence des relations entre les mots ou documents indépendamment des techniques tel que : dictionnaire, réseau sémantique, règles de connaissance, analyse syntaxique ou morphologique. Néanmoins, on peut les intégrer pour améliorer la méthode LSA.

La mesure de similarité permet de tracer les concepts ou grouper les mots et classifier les textes suivant un degré de rapprochement. Pour ce faire, on utilise des fonctions de similarité comme: le cosinus , la mesure de Spearman, la distance euclidienne, etc. (voir Table. 3). On suppose deux vecteurs xi et xj(mots ou documents), alors le cosinus (xi,xj) varie entre ⌊-1,1⌋, si la valeur retournée est égale à –1 (respectivement +1), on dit qu'il s'agit d'une corrélation négatif parfaite (respectivement corrélation positive parfaite) . Une autre mesure est celle de Spearman caractérisée par la formule suivante:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)\sqrt{\frac{\sum_{i=1}^{n}(y_i-\bar{y})^2}{n-1}}\sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}}}$$

La mesure de similarité permet de tracer les concepts ou grouper les mots et classifier les textes suivant un degré de rapprochement. Pour ce faire, on utilise des fonctions de similarité comme: le cosinus, la mesure de Spearman, la distance euclidienne, etc. (voir Table. 3). On suppose deux vecteurs xi et xj(mots ou documents), alors le cosinus (xi ,xj) varie entre [-1,1], si la valeur retournée est égale à –1 (respectivement +1), on dit qu'il s'agit d'une corrélation négatif parfaite (respectivement corrélation positive parfaite) . Une autre mesure est celle de Spearman caractérisée par la formule suivante:

$$\boldsymbol{euclidean}(x,y) = \left(1 \Big/ \left(1 + \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\right)\right)$$

Dans ce cas, la similarité liée à l'équation (20) se calcule suivant l'inverse de la distance euclidienne, on dit que deux vecteurs x, y sont similaires si euclidien(x,y) est proche 1.

Un autre modèle pour calculer la similarité est appelé le modèle géométrique, on suppose qu'on a la matrice approximative X=A''=U1.D1.V1t, alors pour comparer :

- deux termes, on calcule *XXt=U1.D12.U1t*
- deux documents, on calcule *XtX=V1.D12.V1t*
- un terme et un document, on calcule *X= U1.D1.V1t*

Nous dressons le tableau suivant qui contienent les différentes mesures de similaritéappliquées à la classification de textes.

Tableau.3 Représentation des différentes mesures de similarité

Type de mesure	Nom de similarité	Formule
Mesure géométrique	Cosinus (x,y)	$\frac{x*y}{\|x\|*\|y\|}$
Mesure ensembliste	Jaccard (x,y)	$\frac{x*y}{\|x\|^2+\|y\|^2-x*y}$
Mesure ensembliste	Overlap (x,y)	$\frac{x*y}{\min(\|x\|^2,\|y\|^2)}$
Mesure ensembliste	Dice	$\frac{2*x*y}{\|x\|^2+\|y\|^2}$
Mesure géométrique	euclidean	$\frac{1}{1+\|x-y\|}$

Annexe

Mesure ensembliste	Ochiai	$\frac{x^{*}y}{\sqrt{\|x\|^{2}\|y\|^{2}}}$
Mesure distributionnelle	Kullback-Leibler Notée KL	$\sum_{k=0}^{k=n} x_k \log(\frac{x_k}{y_k})$
Mesure distributionnelle	Jensen-Shannon Notée JS	KL(x, (x+y)/2)+KL(y, (x+y)/2)
Mesure distributionnelle	City Block	$\frac{1}{1+\sum(x_i-y_i)}$
Mesure distributionnelle	maximum	$\frac{1}{1+\max(x-y)}$

Nous illustrons la phase de similarité par l'exemple suivant dont les résultats sont extraites. Supposons que nous avons les phrases suivantes avec les termes:
Human, computer, survey, system, response, time, user, interface, trees, graph et minors. Cette base est constituée des titres de cinq articles sur l'interaction homme machine et quatre sur la théorie des graphes (voir tableau suivant).

	Documents
c1	Human machine interface for ABC computer applications
c2	A survey of user opinion of computer system response time
c3	The EPS user interface management system
c4	System and human system engineering testing of EPS
c5	Relation of user perceived response time to error measurement
m1	The generation of random, binary, ordered trees
m2	The intersection graph of paths in trees
m3	Graph minors IV : Widths of trees and weel-quasi-ordering
m4	Graph minors : A survey

Annexe

Tableau 4 Matrice mots x documents brute (avant LSA)

	C1	C2	C3	C4	C5	M1	M2	M3	M4
Human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
Computer	1	1	0	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	0	1	1	2	0	0	0	0	0
Response	0	1	0	0	1	0	0	0	0
Time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
Survey	0	0	0	0	0	0	0	0	1
Trees	0	0	0	0	0	1	1	1	0
Graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Tableau 5 Matrice reconstruite après l'application de la décomposition SVD (Apres LSA)

	C1	C2	C3	C4	C5	M1	M2	M3	M4
Human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
Computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
User	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
System	0.45	1.23	1.05	2.27	0.56	-0.07	-0..15	-0.21	-0.05
Response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
Time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
Survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
Trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
Graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

Si on trace le coefficient de corrélation de Spearman entre human et user pour les deux tableaux, on trouve qu'il passe de –0.38 à 0.94. On peut dire que les mots human et user qui apparaissent dans des contextes différents avant LSA, figurent dans des documents de significations similaires après LSA

Annexe

L'API TextWise:[W6]

L'API TextWise a été conçu pour permettre aux développeurs d'exploiter la puissance de cette technologie de base à la fois sémantique et de la technologie prédécesseur. Cette API simple permet à n'importe qui d'extraire des concepts clés sémantiques détaillées dans un document de n'importe quelle taille. La technologie est hébergée à distance sur le Cloud TextWise sémantique. Ce service de cloud computing basée sur allège le fardeau sur votre réseau informatique tout en offrant l'accès à la puissance de notre technologie sémantique.
L'API TextWise permet d'accéder aux trois mêmes services sémantiques qui sont intégrés dans ces solutions d'entreprise de propriété:

Assorti

Catégorisation

Marquage Concept

L'API TextWise fournit un appel de service match qui analyse le texte d'entrée ou document fourni dans l'appel et renvoie une liste ordonnée de la pertinence des documents similaires d'une collection de documents. Au lieu de simplement compter les termes ou mots-clés dans le texte, Gist sémantique génère une analyse sophistiquée conceptuelle du document et utilise que pour calculer la similarité entre la saisie de texte et les documents de la collection cible.
TextWise recherche sémantique offre d'une seconde correspondance, à haut débit et l'évolutivité des index contenant des milliers à des dizaines de millions de documents. En outre, le système permet en temps réel et batch incrémental mise à jour des index personnalisés.
Le service d'appariement peut être utilisé avec un large éventail de textes, y compris des mots isolés, des phrases, mots-listes (métadonnées, par exemple), des passages courts, les étiquettes d'image, des pages Web ou des documents en texte intégral (ex. articles techniques).
Vous pouvez faire correspondre le texte contre cette nouvelle disposition du public, blog, image, vidéo, Wikipedia et les indices d'Amazon, ou vous pouvez créer un index personnalisé contenant vos documents exclusifs à faire correspondre (licence supplémentaire nécessaire).

Annexe

MotsConcept

L'appel API TextWise concept de service génère des balises lisibles par machine et affichable pour faciliter la navigation et le contenu des ressources. Chaque balise est accompagné par un poids qui décrit la pertinence de ce terme pour le contenu, de sorte que vous pouvez voir en un coup d'œil ce qu'un document donné est axé sur. Concept balises sont générées par l'obtention d'une liste de mots qui apparaissent à la fois dans le dictionnaire sémantique et le contenu du texte pour les plus pondérés des dimensions sémantiques du texte.

Résultats de texte Wise api sur 20 sujets :

Sujet	Catégorie	Pondération pour catégorie
1	Arts/Music/Rock	0.17599796
	Recreation/Models/Railroad	0.14384997
2	Science/Earth Sciences/Paleontology	0.27423918
3	Sports/Cricket/ICC	0.5841924
4	Society/History/Nineteenth Century	0.3912339
	Society/History	0.3255767
5	Society/Issues/Science_and_Technology	0.17114273
	Society/Ethnicity/Asian	0.1469123
6	Recreation/Pets/Dogs Science/Astronomy/Solar System	0.19841228
		0.16646887
7	Arts/Performing_Arts/Acting	0.21038671
	Arts/Movies/Filmmaking	0.16889691
8	Science/Biology/Animalia	0.13978969
	Business/Electronics_and_Electrical/Electromechanical	0.11782749
9	Science/Social Sciences/Psychology	0.33061334
10	Society/History/Nineteenth Century	0.23883699
11	Arts/Literature/Horror	0.54526293
12	Science/Biology/Animalia	0.25423098

	Science/Physics/Relativity	0.23090316
13	Sports/Basketball/Women	0.14144085
	Society/People/Women	0.10395986
14	Society/Law/General_Practice	0.18998052
	Society/Government/US_Presidents	0.15951909
15	Arts/People	0.20546003
	Arts/Literature/Drama	0.19643293
16	Society/History/Twentieth_Century	0.1907061
	Business/Employment/Recruitment_and_Staffing	0.18985473
17	Arts/People	0.13658208
	Business/Agriculture_and_Forestry/Forestry	0.10704183
18	Arts/Performing_Arts/Acting	0.37039322
19	Society/History	0.36399275
20	Science/Astronomy/Solar_System	0.35097924

Et on cas ou on lui donne le texte complet la catégorie sont :

- Arts/Literature/Horror
  0.13276315
- Business/Industrial_Goods_and_Services/Packaging
  0.11654533.

L'application Mallet [W7]

I. Définition de l'application Mallet

MALLET est un package basé sur Java pour le traitement statistique du langage naturel, classification de documents, clustering, modélisation de sujet, l'extraction d'information, et une autre machine d'apprentissage applications au texte.

MALLET inclut des outils sophistiqués pour la classification des documents, MALLET inclut des outils pour l'étiquetage de séquence pour des applications telles que l'extraction des entités nommés .des algorithmes incluent modèles de Markov cachés, maximum d'entropie modèles de Markov et champs aléatoires conditionnels.

Ces méthodes sont mises en œuvre dans un système extensible pour transducteurs à états finis.

II. Mallet et la modélisation sujet

Un modèle de sujet offre un moyen simple d'analyser de grandes quantités de texte sans titre. Un "sujet" se compose d'un groupe de mots qui se produisent fréquemment ensemble. Pour modéliser les sujet on utilisant des indices contextuels, mais le modèle sujet peut se basée sur les mots avec des significations similaires et de distinguer entre les utilisations de mots avec des significations multiples.

Dans notre cas mallet utilise la méthode LDA qui a était décrierprécédemment dans le chapitre état de l'art.

II. Étape de création du modèle sujet et instruction

II.1 Importation des documents texte:

Une fois MALLET téléchargé et installé, l'étape suivante consiste à importer des fichiers texte au format interne de Mallet. Les instructions suivantes supposent que les documents devant être utilisé comme entrée pour le modèle de sujet sont dans des fichiers séparés, dans un répertoire qui ne contient pas d'autres fichiers..

Allez dans le répertoire MALLET et exécutez la commande

bin\mallet import-dir --input pathway\to\the\directory\with\the\files --output tutorial.mallet --keep-sequence --remove-stopwords

un fichier appelé*tutorial.mallet* a était crée par cette instruction Ce fichier contient maintenant toutes vos données, dans un format *MALLET*qu'onpeut travailler avec.

Si on travaille avec de très grandes collections de fichiers ou de fichiers - en effet, de très grandes – on peut rencontrer des problèmes avec l'espace de tas, la mémoire de travail de notre ordinateur.cesproblèmes peuvent ce produire pendant la séquence de l'importation, Par défaut, *MALLET* permet de 1Go de mémoire à utiliser lors de la séquenced' importation. Si notre système a plus de mémoire, on peut essayer

d'augmenter la mémoire allouée à *la machine virtuelle Java*. Pour cella, vous devez modifier le code dans le *maillet* fichier trouvé. Ouvrez le *Mallet.bat* fichier (*C: \ Mallet \ bin \ mallet.bat*).

Trouvez la ligne suivante: MÉMOIRE = 1g

On peut ensuite modifier les valeurs vers le haut 1g - à 2g, 4g, voir plus en fonction de RAM notre système, qu'on peut trouver en lisant les informations système de la machine. On enregistre les modifications. On devrait maintenant être en mesure d'éviter l'erreur. Si ce n'est pas le cas, on augmente la valeur à nouveau.

II.2 Création du modèle sujet

Cette commande :

bin\mallet train-topics --input tutorial.mallet

Ouvre notre fichier *tutorial.mallet* fichier et exécute la routine de modèle sujet (l'opération de création du modèle sujet) et en utilisant uniquement les paramètres par défaut. Comme il parcourt la routine, cette opération essaye de trouver la meilleure division des mots en thèmes, la fenêtre d'invite de commandes se remplit de sortie de chaque course. Lorsque cela est fait, vous pouvez défiler vers le haut pour voir ce qu'il était sorti.

L'ordinateur imprime les mots clés, les mots qui aident à définir un sujet statistiquement significatif, par la routine. Dans la figure, le premier sujet il affiche pourrait ressembler à ceci (vos mots clés peut paraître un peu différente):

1 0 5 test cricket Australian hill acting England northern leading ended innings record runs scored run team batsman played society English

Si vous êtes un fan de cricket, vous reconnaîtrez que tous ces mots pourraient être utilisés pour décrire un match de cricket. Ce dont nous parlons ici d'un sujet lié au de cricket. Si vous allez à *C: \ maillet \-échantillons de données \ web \ fr \ hill.txt* , vous verrez que ce fichier est une brève biographie du célèbre joueur de cricket australien

Hill, Clem. Le *0* et le *5* , nous allons parler plus tard dans la leçon. Notez que *MALLET* comprend un élément de hasard, de sorte que les listes de mots clés sera différent à chaque fois que le programme est exécuté, même si sur le même ensemble de données.

Mais quand on retourne *MALLET* répertoire et on tape dir. ont voit qu'il n'y a pas de fichier de sortie. Alors que nous avons réussi à créer un modèle de sujet, nous n'avons pas enregistré la sortie.

II.3 Création des sauvegarde

La commande suivante

bin\mallet train-topics --input tutorial.mallet --num-topics 20 --output-state topic-state.gz --output-topic-keys tutorial_keys.txt --output-doc-topics tutorial_compostion.txt

- Ouvrevotre *tutorial.mallet* fichier
- forme 20 sujets
- affiche tous les mots dans votre corpus de documents et le sujet auquel il appartient
- émet un document texte vous montrant ce que les mots clés sont meilleures pour chaque sujet (*tutorial_keys.txt*)
- et génère un fichier texte indiquant la répartition, en pourcentage, de chaque sujet dans chaque fichier texte original que vous avez importé (*tutorial_composition.txt*).

Nous fichiers seront délivrées en sortie au bas de la liste des fichiers et répertoires dans le *répertoire C: \ Mallet* . Ouvrir *tutorial_keys.txt* dans un traitement de texte Vous êtes présenté avec une série de paragraphes. Le premier paragraphe est *sujet 0* , le deuxième alinéa est sujet 1, le troisième alinéa est *sujet 2* , etc (La sortie commence à compter à 0 au lieu de 1, donc si vous lui demandez de déterminer 20 thèmes, la liste se déroulera entre 0 à 19). Le deuxième nombre dans chaque paragraphe est le Dirichlet de paramètre pour le sujet (Dirichlet parameter). Ceci est lié à une option que on n'a pas de