République Algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université SAAD DAHLAB Blida

Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études
pour l'obtention du diplôme de Master en Informatique

Spécialité : **Ingénierie du Logiciel**

Thème

Mise en œuvre d'un outil d'analyse de données textuelles

Réalisé par :
- **BOUKHERS Rachid**
- **MERAGA Omar**

Promotrice : **Mme BENBLIDIA N.**
Encadreur : **Mme ATTAF S.**

Présenté le 27 Octobre 2014 devant le jury composé de :

- **Mme BOUSTIA N.** (Présidente)
- **Mme BENBLIDIA N.** (Promotrice)
- **Mme REZOUG N.** (Examinatrice)
- **Mme AMEUR K.** (Examinatrice)

Année universitaire : **2013 / 2014**

Remerciement

Avant tout, nous remercions le bon dieu le tout puissant qui nous a donné la force et le courage pour réaliser ce travail.

Et la prière sur notre guide le prophète Mohammed la plus chère personne à notre âme, qui la prière soit sur lui jusqu'à l'éternité.

Nous exprimons nos remerciements à notre promotrice Mme BENBLIDIA N. pour son aide et sa compréhension, et à notre encadreur Mme ATTAF S. pour avoir consacrer du temps pour superviser nos efforts, afin de bien mener ce modeste travail.

Nous réservons une mention particulière à Mr CHAOUA Nabil, pour son aide et ses précieux conseils.

Nous adressons nos remerciements les plus respectueux au jury qui a bien voulu accepter de juger notre travail.

Nous tenons à remercier vivement nos enseignants qui ont contribué à notre formation durant tout notre cycle d'étude, ainsi qu'à toutes les personnes qui nous ont soutenues et tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Résumé :

Ce projet s'inscrit dans le cadre de mise en œuvre d'un outil d'analyse de données textuelles. Ce travail a été fait durant un stage pratique au niveau du département d'informatique à l'université SAAD DAHLAB Blida.

Administrer des données texte, afin de pouvoir prendre profit des informations qu'elles contiennent, est devenu très essentiel, à cause de leurs volumes importants et la quantité d'information qu'elles puissent contenir. Afin de permettre l'analyse de ce type de données, il est devenu plus que nécessaire d'adapter un modèle de représentation de données, permettant de décrire ces données textuelles de façon suffisamment formelle, pour qu'elles puissent être prêtes à l'analyse.

Plusieurs modèles ont été proposés, ils servent à la représentation multidimensionnelle des données textuelles, on peut les classer en deux familles de modèles : modèles extensifs, et modèles à nouveaux concepts. Un nouveau modèle appelé MSMTO a été proposé, c'est un modèle à nouveaux concepts, il est puissant puisque il prend en compte la sémantique des données textuelles, il offre aussi la flexibilité, en tenant compte du contenu sémantique de données textuelles comme une mesure, un fait ou même une dimension.

Notre choix s'est porté sur l'implémentation d'un outil d'analyse de données texte, basé sur le modèle MSMTO.

Pour extraire des sujets (Topics) à partir d'un corpus de documents, on a opté l'approche statistique LDA (Latent Dirichlet Allocation). En ce qui concerne la sémantique, on a utilisé l'API puissant de TextWise. On a choisi aussi le format XML pour stocker les informations extraites (Les hiérarchies sémantiques).

Pour le développement on a utilisé le langage Java (Java SE), et concernant le SGBD Natif, on a choisi le logiciel eXist-db.

Abstract :

This project is part of implementation of a textual data analysis tool.
This work was done during an internship at the Department of Computer Science at SAAD DAHLAB Blida University.

Administer text data, in order to take advantage of the information they contain, has become very important because of their large volumes and the amount of information they may contain. To enable the analysis of such data, it has become necessary to adapt a model of data representation for describing the textual data in a sufficiently formal way, so they can be ready for analysis.

Several models have been proposed, they are used for multidimensional representation of textual data, they can be classified into two families of models: Extensive models, new concepts models. A new model called MSMTO has been proposed, it's a model of new concepts, it's powerful because it takes into account the semantics of text data, it also offers the flexibility, taking into account the semantic content of data text as a measure, fact or even a dimension.

Our choice fell on the implementation of a tool of analysis of text data, based on the MSMTO model.

To extract topics from a corpus of documents, we chose the statistical approach LDA (Latent Dirichlet Allocation). Regarding semantics, we used the powerful TextWise API. We also chose the XML format to store the extracted information (semantic hierarchies).

For development we used the Java (Java SE) language, and for the native DBMS, we chose the eXist-db software.

Table des matières

• Liste des tableaux

➢ Chapitre 3 : Modèles d'entrepôt de données textuelles

• Liste des figures

➢ Chapitre 2 : Modèles de sujets (Topic models)

➢ Chapitre 3 : Modèles d'entrepôt de données textuelles

➢ Chapitre 4 : Implémentation

Introduction

1) Contexte général :

A l'heure actuelle, avec l'accroissement continu du volume de l'information, Les données à analyser ne sont plus seulement numériques ou symboliques, mais sous différents formats (texte, image, son, vidéo…etc.), provenant de sources différentes. De cela est né un nouveau besoin, celui de l'analyse de données dite « Complexes ». Notre projet s'inscrit dans le cadre de l'analyse d'un type particulier des données complexes, « Les données textes ».

De nos jours, la documentation électronique fait partie intégrante de l'information et stratégie de communication de toute organisation moderne. En fait, il est entendu que plus de 80% [13] des données nécessaires pour les opérations des organisations sont encapsulés dans des documents, et pas seulement dans les bases de données opérationnelles. Ces données textuelles restent hors de portée des systèmes d'aide à la décision, ce qui indique que la plupart des informations restent inaccessible.

Les systèmes d'aide à la décision classiques utilisés dans l'analyse des données simples, ont déjà donné de bons résultats. Néanmoins, ces systèmes ne sont pas adaptés pour l'analyse des documents textuels, qui met en valeur la nécessité de créer de nouveaux modèles multidimensionnels pour les données textuelles. Le stockage de ce type de données reste aujourd'hui comme l'une des difficultés majeures qui impliquent de nombreux défis en ce qui concerne leur modélisation et intégration d'une part, et l'analyse de l'autre part. Les entrepôts de texte ont émergé comme une nouvelle solution pour l'analyse de données textuelles. La nature complexe de ces données nécessite un traitement particulier, en tenant compte de leur sémantique. Dans la littérature, les méthodes de recherche d'information et Datamining ont donné de bons résultats pour l'exploration de données textuelles. L'idée clé derrière les entrepôts de texte est de relier les techniques de Datamining et de récupération de l'information, avec des techniques OLAP. Les modèles d'entrepôt de texte existants, tels que le cube de sujet (Topic cube) [13] et le cube de texte (Text cube) [9], ont proposé une première solution à ce problème en intégrant une dimension sémantique. Cependant, il reste encore inexploité, chaque modèle propose un type sémantique spécifique (termes, sujets ...) ce qui pourrait limiter les possibilités d'analyse.

La flexibilité représente un autre défi dans l'analyse de données textuelles. Dans les systèmes d'information décisionnels classiques, un fait représente un sujet d'analyse prédéfini. Une définition de fait dès le départ réduit la flexibilité d'analyse pour le décideur, il est obligé d'utiliser ces faits comme sujets. Afin d'assurer une plus grande flexibilité, certains travaux ont proposé de supprimer la notion de fait, pour éviter de contraindre l'analyse par des faits prédéfinies. Un modèle appelé MSMTO [chapitre 3] a abordé le problème de la flexibilité d'analyse en supposons que le contenu sémantique des données textuelles peut jouer le rôle d'une mesure analytique ainsi que d'un fait. En d'autres termes, un objet est considéré comme une mesure pour un ensemble donné de données et comme un axe pour l'analyse d'un autre ensemble de données. À notre connaissance, ce double rôle, afin de garantir une bonne flexibilité d'analyse, n'est pas utilisé par un travail existant.

L'approche de modélisation que propose le modèle MSMTO [chapitre 3] se concentre particulièrement sur les aspects suivants : (i) l'inclusion de la structure de données textuelles, (ii) l'inclusion des aspects sémantiques des données textuelles, et enfin (iii) la flexibilité d'analyse. Ce modèle est basé sur le paradigme objet, un choix justifié par la capacité de représentation des modèles orientés objet. Le but de ce modèle est de concevoir un modèle d'entrepôt qui peut représenter des données textuelles et faire de l'analyse multidimensionnelle basée sur l'information contenue dans ces entrepôts, offrant une bonne flexibilité d'analyse.

Notre projet consiste à implémenter une application d'analyse de données textuelles, basée sur le modèle MSMTO [chapitre 3].

2) Objectifs et besoins :

L'objectif du travail est le développement d'une solution logicielle permettant aux utilisateurs, d'analyser des données texte, en assurant :

- ✓ La construction d'un entrepôt adapté aux données textuelles prenant en considération l'aspect sémantique de ces dernières.
- ✓ Le développement d'un processus ETL assurant l'alimentation d'un entrepôt texte.
- ✓ Le développement des opérateurs de construction de cubes textes.

Chapitre 1

Introduction à la fouille de textes (Text Mining)

Introduction :

Concernant ce chapitre, on va donner une définition générale de la fouille de textes, on va expliquer aussi la procédure suivie pour la mettre en œuvre en citant les deux approches connues : l'approche statistique et l'approche sémantique. L'utilité et quelques applications de la fouille de textes sont aussi montrées dans ce chapitre.

1) Définition :

La fouille de textes ou "l'extraction de connaissances" dans les textes est une spécialisation de la fouille de données et fait partie du domaine de l'intelligence artificielle. Cette technique est souvent désignée sous l'anglicisme Text Mining.

C'est un ensemble de traitements informatiques consistant à extraire des connaissances selon un critère de nouveauté ou de similarité dans des textes produits par des humains pour des humains. Dans la pratique, cela revient à mettre en algorithmes un modèle simplifié des théories linguistiques dans des systèmes informatiques d'apprentissage et de statistiques.

Les disciplines impliquées sont donc la linguistique calculatoire, l'ingénierie du langage, l'apprentissage artificiel, les statistiques et bien sûr l'informatique.

Les outils de Text Mining ont pour vocation d'automatiser la structuration des documents peu ou faiblement structurés. Ainsi, à partir d'un document texte, un outil de Text Mining va générer de l'information sur le contenu du document. Cette information n'était pas présente, ou explicite, dans le document sous sa forme initiale, elle va être rajoutée, et donc enrichir le document. [W1] [W4]

2) Comment ça marche ?

Il y a quelques règles de base que les outils de Text Mining se doivent de respecter dans leur traitement. Ces règles de base sont plus ou moins chronologiquement les suivantes :

- D'abord le logiciel doit reconnaître les unités de la langue que sont les mots (Tokenisation).
- Ensuite il doit savoir interpréter et prendre en compte la ponctuation et la mise en page (retour à la ligne, paragraphe, etc.).
- Puis les formes lexicales et grammaticales, qui peuvent énormément varier selon que la langue est l'anglais, l'arabe ou le chinois.
- Ensuite, il y a une phase de lemmatisation : elle consiste à identifier les différentes flexions d'un terme, ou déclinaisons d'un verbe.

L'ensemble des phases précédentes relèvent de ce qu'on appelle l'analyse linguistique, au sortir de laquelle nous avons un document que le logiciel de Text Mining a transformé. Si le document initial était fait pour les yeux de l'humain, le document après traitement est fait pour un traitement par les machines. [W4]

Deux approches, qui ne sont pas antinomiques, peuvent ensuite être envisagées :

- une approche statistique
- une approche sémantique

2.1) L'approche statistique :

Elle consiste à ne voir le document que via le prisme du nombre et des chiffres. Ainsi l'outil statistique de Text Mining produit des informations sur le nombre d'occurrence d'un terme, le nombre de cooccurrence de plusieurs termes, la fréquence d'apparition d'un terme dans un document ou un corpus.

Il peut encore produire ce que l'on appelle des « vecteurs de sens », qui sont des « concepts » statistiques de cooccurrence de termes qui permettent de classer et/ou de catégoriser tout un corpus. [W4]

2.1.1) Les avantages de l'approche statistique :

Le principal réside dans son très faible coût d'entretien eut égard au véritable service que cela peut apporter, à condition que le volume du corpus documentaire soit significatif, voire très important. [W4]

2.2.1) Les avantages de l'approche sémantique :

On peut paramétrer le moteur de Text Mining pour coller à la spécificité du corpus documentaire en exploitant l'ensemble des référentiels du domaine ou de l'organisation. On peut également modéliser des connaissances métiers spécifiques pour effectuer des traitements de Text Mining qui répondent à un besoin bien identifié. La pertinence des résultats obtenus est beaucoup plus fine et généralement meilleure que dans l'approche statistique (la notion de « meilleur » étant toutefois toujours relative). [W4]

2.2.2) Les désavantages de l'approche sémantique :

Le coût d'exploitation et de maintenance est très fort. Cela demande des ressources matérielles, budgétaires et humaines significatives. De plus si le corpus est important, le temps de traitement requis peut être très long et peut représenter un frein à la démarche. [W4]

3) A quoi cela peut bien servir ?

- ✓ à classifier automatiquement des documents
- ✓ à avoir un aperçu du contenu d'un document sans le lire
- ✓ à alimenter automatiquement des bases de données
- ✓ à faire de la veille sur des corpus documentaires importants
- ✓ à enrichir l'index d'un moteur de recherche pour améliorer la consultation des documents

Bref, plusieurs usages et plusieurs services peuvent découler des solutions de Text Mining. [W4]

4) Quelques applications :

Recherche d'information :

Les moteurs de recherche tels Google, Exalead ou Yahoo! sont des applications très connues de fouille de textes sur de grandes masses de données. Notons toutefois que les moteurs de recherche ne se basent pas uniquement sur le texte pour l'indexer, mais également sur la façon dont les pages sont mises en valeurs les unes par rapport

aux autres. L'algorithme utilisé par Google est PageRank, et il est courant de voir HITS dans le milieu académique. [W1]

Applications biomédicales :

Un exemple d'application biomédicale de fouille de textes est PubGene, qui combine la fouille de textes et la visualisation des résultats sous forme de réseaux graphiques. Un autre exemple d'utilisation d'ontologies avec la fouille de textes est GoPubMed.org. [W1]

Filtrage des communications :

Beaucoup de gestionnaires de courriers électroniques sont maintenant livrés avec un filtre anti-spam. Il existe aussi des logiciels anti-spam qui s'interfacent entre le serveur de courrier et votre gestionnaire de courrier. [W1]

Applications de sécurité :

Le système mondial d'interception des communications privées et publiques Echelon est un exemple d'utilisation militaire et économique de la fouille de textes.
En 2007, la division de lutte anticriminelle d'Europol a acquis un système d'analyse afin de lutter plus efficacement contre le crime organisé. Ce système intègre parmi les technologies les plus avancées dans le domaine de la fouille et d'analyse de textes. Grâce à ce projet, Europol a accompli des progrès très significatifs dans la poursuite de ces objectifs. [W1]

Intelligence économique :

Les méthodes de fouilles de texte contribuent au processus d'Intelligence économique : cartographie des relations, détection de relations explicites entre des acteurs (concessions de licences, fusions / acquisitions, ...) [W1]

Marketing :

Les techniques de la fouille de texte sont très utilisées pour analyser les comportements d'internautes : parcours de visite, critères favorisant le déclenchement d'un achat, efficacité de campagnes publicitaires, analyse du sentiment... [W1]

5) Exemple (indexation de textes) :

La fouille de texte peut consister en l'indexation d'un ensemble de textes par rapport aux mots qu'ils contiennent. On peut ensuite interroger l'index ainsi créé pour connaître les similarités entre une requête et notre liste de textes.

L'algorithme d'indexation se décrit comme suit :

1. On indexe le texte par rapport aux mots qui le composent
2. On effectue l'index inversé (on indexe les mots contenus par rapport aux textes les contenant)
3. Au moment de traiter une requête, on teste la similarité de cette requête avec notre index inversé
4. Cela nous retourne les textes similaires avec notre requête, et pour chaque texte, un rang

Les applications sont multiples : d'une simple indexation pour les moteurs de recherche à l'extraction de connaissances dans des documents non structurés.

D'autres techniques, comme la lemmatisation, permettent d'améliorer notre indexation, en perdant néanmoins une partie du sens. [W1]

Conclusion :

Enfin, on a appris beaucoup de concepts en ce qui concerne la fouille de textes. Après les différentes étapes d'un processus de fouille de textes, on a le choix d'appliquer soit une approche statistique soit une approche sémantique pour la catégorisation des documents. Ces deux approches ne sont pas antinomiques, alors, on peut les utiliser simultanément, comme est le cas de notre application. Les domaines d'application de la fouille de textes sont multiples.

Introduction :

Dans ce chapitre, on va montrer c'est quoi un modèle de sujets. On va citer et expliquer d'une manière générale quelques modèles de sujets (LSA, PLSA), et détailler le modèle LDA, puisque c'est le modèle le plus récent, et c'est lui qui fait partie de l'implémentation de notre application.

1) Définition :

Dans le domaine de l'apprentissage automatique et du traitement automatique du langage naturel (TALN), un modèle de sujet (topic model) est un type de modèle probabiliste pour découvrir les "sujets" abstraits qui se produisent dans une collection de documents. Intuitivement, un document est sur un sujet particulier, on attendrait que les mots particuliers paraissent dans le document plus ou moins fréquemment : "chien" et "os" paraîtront souvent plus dans les documents au sujet de chiens, "chat" et "miaulement" paraîtront dans les documents au sujet de chats, et "le" et "est" paraîtront également dans les deux. Un document concerne typiquement des sujets multiples dans des proportions différentes ; donc, dans un document qui est 10% au sujet de chats et 90% au sujet de chiens, il y aurait probablement approximativement 9 fois plus des mots qui concernent les chiens que des mots qui concernent les chats. Un modèle du sujet capture cette intuition dans une structure mathématique qui permet à examiner un ensemble de documents et découvrir, basé sur les statistiques des mots dans chacun, que les sujets peuvent être et ce que la balance de sujets de chaque document est.

Bien que les modèles du sujet aient été décrits et ont rendu effectif en premier dans le contexte du traitement automatique du langage naturel (TALN), ils ont des applications dans d'autres champs tel que La bio-informatique. [W1]

2) Histoire :

Un premier modèle du sujet a été décrit par Papadimitriou, Raghavan, Tamaki et Vempala en 1998. Un autre, appelé : l'Indexation sémantique latente probabiliste (PLSI pour : Probabilistic Latent Semantic Indexing), a été créé par Thomas Hofmann en 1999. Allocation de Dirichlet latente (LDA pour : Latent Dirichlet Allocation), peut-être le modèle du sujet le plus commun actuellement en usage, est une généralisation de

PLSI développée par David Blei, Andrew Ng et Michael I. Jordan en 2002, permettre aux documents d'avoir un mélange de sujets. D'autres modèles du sujet sont généralement des extensions sur LDA, tel qu'allocation Pachinko (Pachinko allocation) qui améliore sur LDA en modelant des corrélations entre sujets en plus des corrélations du mot qui constituent des sujets. [W1]

3) Analyse sémantique latente (LSA) :

L'analyse sémantique latente (LSA, de l'anglais : Latent semantic analysis) ou indexation sémantique latente (ou LSI, de l'anglais : Latent semantic indexation) est un procédé de traitement des langues naturelles, dans le cadre de la sémantique vectorielle. La LSA fut brevetée en 1988 et publiée en 1990.

Elle permet d'établir des relations entre un ensemble de documents et les termes qu'ils contiennent, en construisant des « concepts » liés aux documents et aux termes. [W1]

4) Analyse sémantique latente probabiliste (PLSA) :

L'analyse sémantique latente probabiliste (de l'anglais, Probabilistic latent semantic analysis : PLSA), aussi appelée indexation sémantique latente probabiliste (PLSI), est une méthode de traitement automatique des langues inspirée de l'analyse sémantique latente.

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 surapprentissage, 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). [W1]

5) LDA :

5.1) Introduction :

Le modèle Latent Dirichlet Allocation (LDA) [2] est un modèle probabiliste génératif qui permet de décrire des collections de documents de texte ou d'autres types de données discrètes. LDA fait partie d'une catégorie de modèles appelés "topic models", qui cherchent à découvrir des structures thématiques cachées dans des vastes archives de documents. Ceci permet d'obtenir des méthodes efficaces pour le traitement et l'organisation des documents de ces archives : organisation automatique des documents par sujet, recherche, compréhension et analyse du texte, ou même résumer des textes. Aujourd'hui, ce genre de méthodes s'utilisent fréquemment dans le web, par exemple pour analyser des ensembles d'articles d'actualité, les regrouper par sujet, faire de la recommandation d'articles, etc. Des modèles de ce type peuvent également s'utiliser sur des images, en utilisant des "mots visuels", par exemple pour regrouper des images par catégorie (voir L. Fei-Fei et P. Perona [3]), ou encore pour les problèmes de filtrage collaboratif (collaborative filtering), par exemple la recommandation de films, en assimilant un utilisateur et les films qu'il a vus à un document et les mots qu'il contient.

Le LDA est un modèle Bayesien hiérarchique à 3 couches (voir Fig. 2.2 pour une représentation graphique) : chaque document est modélisé par un mélange de topics (thèmes) qui génère ensuite chaque mot du document. La structure des documents du corpus peut être déterminée par des techniques d'inférence approchée basées sur des méthodes variationnelles ou des méthodes de Gibbs sampling. Les paramètres des distributions peuvent être estimés par l'algorithme EM. La librairie lda-c de D. Blei (qui utilise des méthodes variationnelles pour l'inférence et l'estimation de paramètres)

- Un corpus est une collection de D documents, $D = (w_1, \ldots . w_D)$.
- Les variables $z_{d,n}$ représentent le topic choisi pour le mot $w_{d,n}$.
- Les paramètres θ_d représentent la distribution de topics du document d.
- α et η définissent les distributions à priori sur θ et β respectivement, où β_k décrit la distribution du topic k.

Processus de génération : Le processus génératif suivi par LDA pour un document w est le suivant (voir le modèle graphique de Fig. 2.2) :

1. Choisir $\theta \sim \text{Dirichlet}(\alpha)$.
2. Pour chaque mot w_n :
   - Choisir un topic $z_n \sim \text{Multinomial}\ (\theta)$
   - Choisir un mot $w_n \sim \text{Multinomail}(\beta_k)$, avec $k = z_n$.

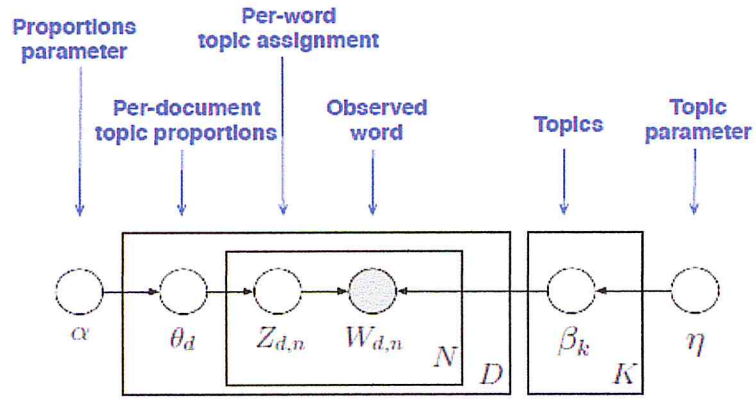

Fig. 2.2 : Représentation de LDA sous forme de modèle graphique [1]

Les boîtes représentent des répliques du modèle qu'elles contiennent (par exemple, il y a une boîte N pour chaque document D).

Loi de Dirichlet : La loi de Dirichlet permet de tirer une variable θ telle que $\forall i, \theta_i \geq 0$ et $\sum_{i=1}^{k} \theta_i = 1$ (θ est dans le (k − 1)-simplexe). Sa densité est de la forme :

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} a_i)}{\prod_{i=1}^{k} \Gamma(a_i)} \theta_1^{\alpha_1 - 1} \ldots \theta_k^{\alpha_k - 1}$$

avec $\alpha \in \mathbb{R}^k, \alpha_i > 0$ et $\Gamma(x)$ la fonction Gamma. Cette distribution permet donc d'obtenir une distribution multinomiale de paramètre θ, correspondant pour LDA au mélange de topics d'un document w. Fig. 2.3 montre la densité d'une telle distribution pour 3 topics. Chaque sommet du triangle correspond à un topic, et chaque point du triangle représente donc une pondération des 3 topics. Le paramètre $\alpha = \alpha_1 + \cdots + \alpha_k$ contrôle l'homogénéité des θ_i : lorsque α est grand, les θ_i sont proches et homogènes, lorsque α est petit, la plupart des θ_i sont proches de 0 sauf quelques-uns. Dans les cas extrêmes, tous les θ_i sont égaux ($\alpha \to \infty$) ou tous les θ_i sont nuls sauf un($\alpha \to 0$).

L'un des avantages de la loi de Dirichlet est qu'elle est conjuguée à la loi multinomiale, c'est à dire que si $z_1, \ldots, z_N$ sont des variables multinomiales de paramètre θ, alors la variable $\theta|z_1, \ldots, z_N$ donnée par $p(\theta|z_1, \ldots, z_N) \propto p(z_1, \ldots, z_n|\theta)p(\theta|\alpha)$ suit également une loi de Dirichlet. Ceci permettra de simplifier les calculs au moment de l'inférence.

Probabilité jointe et loi marginale : Etant donnés les paramètres α et β, la probabilité jointe du mélange de topics θ, des N topics **z** et de N mots **w** est donnée par :

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta)p(\omega_n|\beta_{z_n})$$

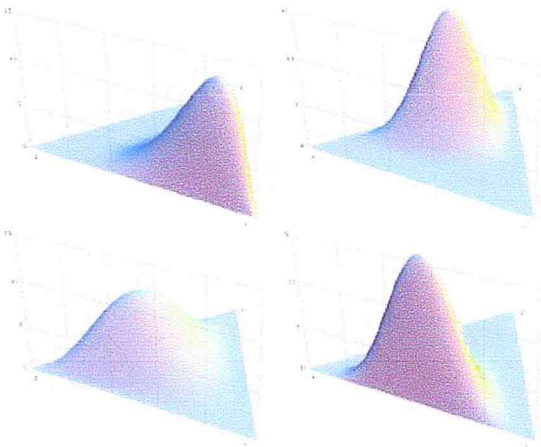

Fig. 2.3 : Fonction de densité de loi de Dirichlet à 2 dimensions (k = 3 topics) sur le triangle [Wikipédia]

La loi marginale d'un document **w** est alors :

$$p(\omega|\alpha,\beta) = \int p(\theta|\alpha)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\theta)p(\omega_n|\beta_{z_n})\right)d\theta$$

(1)

et il suffit de prendre le produit de cette quantité pour chaque document **w** du corpus pour obtenir la probabilité de ce corpus.

5.3) Inférence et estimation de paramètres :

Nous avons décrit le modèle du LDA au paragraphe précédent et montré son utilité pour l'analyse de corpus de documents. Mais les variables et paramètres du modèle ne sont pas connus initialement, et il faut essayer de les apprendre à partir des données observables, c'est à dire les mots des documents.

Dans la représentation graphique de Fig. 2.2, on peut voir que les seules variables observées sont les mots $w_{d,n,}$, alors que toutes les autres variables sont cachées. Etant donnés les paramètres α et β, le rôle de l'inférence est de déterminer les variables cachées θ et z_n d'un document **w**, étant donnée la liste des mots w_n du document. Les principales méthodes d'inférence (approchée) pour LDA sont les méthodes de sampling (notamment le collapsed Gibbs sampling) et les méthodes variationnelles (particulièrement les méthodes mean-field, qui peuvent se faire en batch ou en ligne). On peut ensuite faire recours à ce procédé d'inférence pour estimer les paramètres α, β et η du modèle grâce à l'algorithme EM.

5.3.1) Inférence :

Le problème principal de l'inférence pour LDA est celui de déterminer la distribution à posteriori des variables cachées étant donné le document (et les paramètres α et β) :

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)}$$

Cette distribution est malheureusement très difficile à calculer, comme on peut déjà le remarquer sur l'expression du dénominateur donnée à l'équation 1 (qui peut s'exprimer en fonction des paramètres du modèle). Une inférence exacte utilisant la distribution à posteriori des paramètres est donc inenvisageable. Mais il existe plusieurs méthodes d'inférence approchée qui peuvent être utilisées pour LDA, utilisant par exemple l'approximation variationnelle ou le Markov chain Monte Carlo.

5.3.1.1) Markov chain Monte Carlo et Gibbs sampling :

Les méthodes Markov chain Monte Carlo (MCMC) consistent à construire une chaîne de Markov sur les variables cachées dont la loi stationnaire est la distribution à posteriori cherchée. Pour rappel, une chaîne de Markov est définie par une loi de transition $p(\theta^{t+1}, z^{t+1}|\theta^t, z^t)$ d'un état (θ^t, z^t) au suivant(θ^{t+1}, z^{t+1}), et peut converger vers une loi stationnaire qui est laissée invariante par la transition. Une fois l'état stationnaire atteint, les échantillons donnés par la chaîne de Markov suivent cette loi stationnaire qui est la distribution voulue.

Le Gibbs sampling est une méthode MCMC où la transition de la chaîne est donnée par la loi conditionnelle d'une variable cachée étant données les observations et l'état courant des autres variables cachées. Pour LDA, on peut successivement obtenir des échantillons $\theta|w, z$ puis $z_n|\theta, w$.
Le collapsed Gibbs sampling échantillonne sur les topics seulement, depuis $z_n|z_{-n}, w$ z, en intégrant les θ.

5.3.1.2) Inférence variationnelle :

L'inférence variationnelle utilise l'optimisation plutôt que l'échantillonnage. L'idée est de commencer par établir une distribution sur les variables cachées avec des paramètres libres (les paramètres variationnels), puis d'optimiser les paramètres variationnels pour que la distribution converge vers la distribution à posteriori souhaitée.

5.3.1.3) Estimation des paramètres :

Pour estimer les paramètres α et β, on peut utiliser la méthode empirical Bayes (ou maximum de vraisemblance marginale), qui consiste à chercher des paramètres α et β qui maximisent la log-vraisemblance (marginale) des données :

$$\ell(\alpha, \beta) = \sum_{d=1}^{D} \log p(w_d|\alpha, \beta)$$

On a vu (équation 1) que le calcul de $p(w_d|\alpha, \beta)$ est inenvisageable en pratique, mais l'inférence variationnelle permet d'obtenir la borne inférieure de la log-vraisemblance qui peut être exploitée par un algorithme Espérance-Maximisation appelé variational EM, qui utilise en plus les paramètres variationnels de la méthode décrite au paragraphe 3.1.2. L'algorithme itère les deux étapes suivantes : l'étape E qui effectue une inférence variationnelle avec les paramètres α et β courants pour calculer la log-vraisemblance, et l'étape M qui maximise en α et β la borne inférieure de la log-vraisemblance calculée.

5.4) Résultats :

Les résultats de Fig. 2.4 ont été obtenus en utilisant la librairie C de David Blei lda-c sur un corpus de 2246 articles d'Associated Press. Chaque colonne représente un thème (topic) découvert, et les mots y sont classés par probabilité décroissante. On peut voir par exemple que la première colonne traite de présidents et de politique, la troisième de faits policiers et la septième de marchés financiers.

Un autre exemple est donné dans Fig. 2.5, qui montre quelques thèmes découverts par LDA sur un corpus de 17000 articles scientifiques du magazine Science.

bush	i	police	soviet	percent	mecham	stock
dukakis	think	shot	gorbachev	year	keating	market
campaign	dont	man	president	rate	senators	index
jackson	people	arrested	summit	last	lincoln	stocks
president	like	two	reagan	report	deconcini	trading
democratic	get	people	bush	prices	meeting	million
convention	going	city	union	increase	john	dow
presidential	just	shooting	europe	month	barry	issues
republican	say	night	gorbachevs	rose	like	rose
new	know	killed	moscow	price	time	volume
vice	im	yearold	leader	production	office	shares
george	thats	officers	mikhail	department	years	jones
bentsen	see	death	nato	months	nixon	exchange
primary	go	car	meeting	annual	wine	new
sen	things	authorities	world	average	bishops	average
michael	make	found	new	government	church	wall
bushs	back	monday	leaders	expected	five	american
reagan	got	wounded	foreign	new	made	big
told	says	men	visit	inflation	senate	prices
state	am	officer	american	january	told	board

bill	communist	court	dollar	budget	israel	police
senate	party	case	cents	bush	israeli	people
house	korea	supreme	late	billion	jewish	students
legislation	south	ruling	gold	congress	arab	demonstrators
sen	korean	judge	lower	spending	palestinian	protesters
measure	north	state	cent	deficit	peace	killed
vote	first	appeals	higher	president	palestinians	government
rep	solidarity	decision	futures	plan	plo	today
congress	walesa	rights	yen	security	minister	two
president	war	federal	bid	cuts	west	protest
committee	two	courts	ounce	new	east	violence
law	president	law	london	fiscal	gaza	building
amendment	leader	appeal	pound	cut	occupied	state
voted	government	order	trading	administration	bank	security
new	people	lawyers	new	programs	jews	injured
debate	talks	attorney	prices	social	shamir	condition
approved	th	ruled	troy	bushs	middle	protests
republican	congress	souter	fell	house	territories	capital
veto	union	civil	francs	federal	meeting	student
year	years	lawyer	york	tax	peres	moslem

Fig. 2.4 : Les 20 premiers mots de quelques topics sur les 100 obtenus avec la librairie lda-c sur des articles d'Associated Press

human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Fig. 2.5 : 4 topics obtenus sur des articles scientifiques

Conclusion :

Finalement, on a appris c'est quoi un modèle de sujets. Aussi, on a bien constaté que le modèle LDA est un modèle probabiliste, donc il se base sur les mathématiques. Aujourd'hui, LDA est le modèle le plus utilisé, puisque il est puissant, et surtout, il a corrigé les erreurs trouvées dans les modèles précédents.

Chapitre 3

Modèles d'entrepôt de données textuelles

Introduction :

Ce chapitre est divisé en deux parties :

- La première (où en sommes-nous ?) présente un ensemble de modèles spécifiques à l'analyse multidimensionnelle des données textuelles, catégorisés en deux familles de modèles : (i) modèles extensifs et (ii) modèles à nouveaux concepts. Cette partie présente aussi une étude comparative des modèles présentés selon quelques critères de comparaison.
- La deuxième (le modèle MSMTO) présente un nouveau modèle puissant qui appartient à l'ensemble des modèles proposant de nouveaux concepts, c'est le modèle MSMTO.

1) Où en sommes-nous ?

Dans le domaine de l'analyse des données textuelles, de différents modèles multidimensionnels d'entrepôts de textes ont été élaborés. Nous distinguons deux familles de modèles : (i) modèles extensifs basés sur les concepts de base des modèles d'entrepôts classiques et (ii) modèles à nouveaux concepts proposant une nouvelle manière de percevoir la complexité des données textuelles. Nous présentons dans cette partie un ensemble de modèles dédiés à l'analyse multidimensionnelle des données textuelles, ainsi qu'une étude comparative de ces modèles. [4]

1.1) Modèles Multidimensionnels d'analyse des données textuelles :

1.1.1) Modèles extensifs :

Les modèles d'entrepôts extensifs sont de nouveaux modèles multidimensionnels, dédiés à l'analyse des données textuelles. Ces modèles proposent des extensions du modèle d'entrepôt classique en se basant sur les deux concepts de base : fait et dimension. Parmi ces travaux nous citons le moteur multidimensionnel de recherche d'information (MIRE) Lee et al. (2002), qui est basé sur un modèle multidimensionnel de données textuelles. Leur approche consiste à construire un modèle multidimensionnel dédié aux données textuelles et permettre aux utilisateurs de faire des recherches à l'aide des techniques OLAP. Le système MIRE peut répondre à une requête multidimensionnelle tel que 'Trouver tous les documents contenant les termes modélisation multidimensionnelle, qui sont publiés en France pendant l'année 2009'. MIRE est une approche qui permet de construire un système de recherche

d'information sur les bases des systèmes OLAP, où une table de faits contient la mesure (les mots apparaissant dans le document), et les tables de dimensions contiennent des données structurées d'une manière hiérarchique. MIRE intègre un index inversé pour les données textuelles et des méthodes d'accès multidimensionnelles. Ces méthodes sont utilisées pour traiter les dimensions, et peuvent fournir des fonctionnalités d'OLAP tel que le drill down et le roll up sur les dimensions. Pour faire face à des problèmes d'évolutivité, MIRE construit un index inversé et utilise une structure multidimensionnelle d'accès unique modified kdb tree pour accéder aux données multidimensionnelles. La requête 'Trouver tous les documents contenant les termes modélisation multidimensionnelle qui sont publiés en France pendant l'année 2009 ' est traitée en deux phases. Les dimensions TEMPS et LOCALISATION sont retrouvées à partir de la structure d'accès multidimensionnelle. L'ensemble des documents retournés de cette structure seront enregistrés dans une collection de documents intermédiaire. A partir de l'index inversé, les documents qui contiennent les termes 'modélisation multidimensionnelle' seront enregistrés dans un autre ensemble intermédiaire. Un ensemble final de documents pondérés peut être obtenu par la fusion des deux ensembles intermédiaires obtenus précédemment.

Mothe et al. (2003) ont proposé un modèle basé sur un schéma en étoile nommé Docube, qui permet de produire des vues globales de grands corpus de documents, en utilisant la classification. Son élément de base est l'utilisation du concept hiérarchie afin de structurer les collections de documents, chaque hiérarchie correspond à une facette de documents 'dimension d'analyse' pour laquelle les utilisateurs peuvent être intéressés. Quelques exemples de dimensions sont : auteur, affiliation...etc. Ces dimensions ne sont pas différentes de celles utilisées dans les systèmes OLAP classiques. Tandis que le contenu de la table fait est différent. Celle-ci contient un lien qui associe une ligne de cette dernière à chaque document. Ce lien peut être pondéré par rapport au degré d'association du document avec les valeurs de la table de dimension. Ce poids est obtenu par l'application de la méthode de classification vector voting. Le lien est représenté par le fichier Doc.Ref qui correspond à l'identifiant du document ou à son URL. DocCube fournit deux ou trois dimensions de visualisation de sorte que l'utilisateur peut visuellement savoir combien de documents sont reliés les uns aux autres dans l'espace multidimensionnel et accéder directement à leurs contenus. Ils ont proposé aussi une fonction score(Dd) qui retourne les tops

documents, ces scores sont calculés par la moyenne des poids associés aux documents.

Tseng et al. (2006) ont proposé un entrepôt de documents pour l'analyse multidimensionnelle de documents textes. Dans leur modèle, une dimension est représentée par une structure d'arbre de m niveaux, utilisée pour représenter les relations hiérarchiques d'un ensemble de mots clefs issus des documents à analyser, des catégories de documents et des métadonnées tels que le titre, l'auteur, la date...etc.(chaque document est représenté par un ensemble de mots clefs). Toutefois, ils n'ont pas mentionné la façon dont les métadonnées et les mots-clés pourraient être organisés sous forme hiérarchique. Ils utilisent comme mesure d'analyse une mesure numérique qui consiste à calculer le nombre de documents. Par exemple 'calculer le nombre de documents traitant la modélisation multidimensionnelle, entre l'année 2006 et 2013'. Nous pouvons à travers leur modèle, sélectionner un cube de documents à partir de l'entrepôt de documents pour permettre aux utilisateurs de naviguer dans les documents par un forage vers le haut (roll-up) et un forage vers le bas (drill-down) le long de certaines dimensions de différentes granularités.

Lin et al. (2008) ont proposé un cube de textes nommé TextCube dans lequel une dimension textuelle est représentée par une hiérarchie de termes. Cette hiérarchie spécifie les relations sémantiques entre les termes textuels extraits des documents, ce qui permet une navigation sémantique dans les données textuelles grâce aux deux opérateurs qui lui sont associés : pull-up and push-down. Ils définissent aussi dans leur cube, deux mesures d'agrégation adaptées aux données textuelles, fréquence des termes (term frequency TF) et l'index inversés (inverted index IV).

Zhang et al. (2009) 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, Fig. 3.1 nous illustre un exemple d'une hiérarchie de topics (thèmes) extraits des rapports sur les anomalies enregistrées lors des vols. La racine représente l'agrégation de tous les thèmes (tous ce qui représente une anomalie), le niveau suivant comporte certaines anomalies générales, comme Anomaly Altitude Diviation. Un nœud enfant représente un événement spécialisé de l'événement

représenté dans le nœud père, par exemple, Undershoot et Overshoot sont deux anomalies spécifiques à l'événement Anomaly Altitude Deviaton.

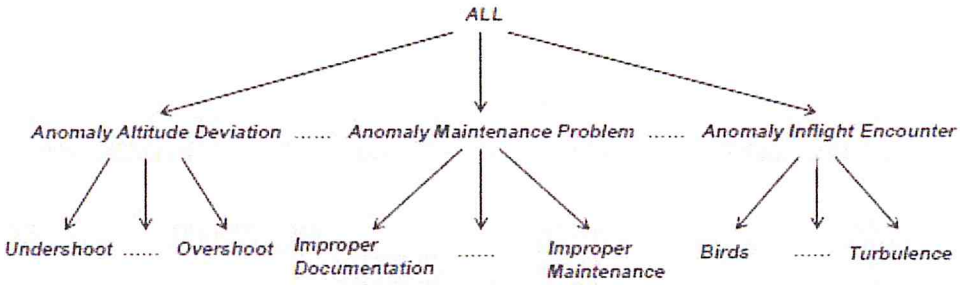

Fig. 3.1 : Exemple d'un arbre d'hiérarchie de thèmes "cas d'anomalies" Zhang et al. (2009)

Topic Cube propose deux mesures probabilistes : la distribution d'un mot dans un thème (word distribution of a topic p (wi)) et la couverture d'un thème par les documents (topic coverage by documents p (topic:j)). La couverture d'un topic est la probabilité qu'un document dj couvre le topic. Ainsi, nous pouvons facilement prédire quel est le sujet dominant dans l'ensemble des documents en agrégeant la couverture sur tous les documents dans l'ensemble. Fig. 3.2 décrit le schéma en étoile d'un Topic Cube

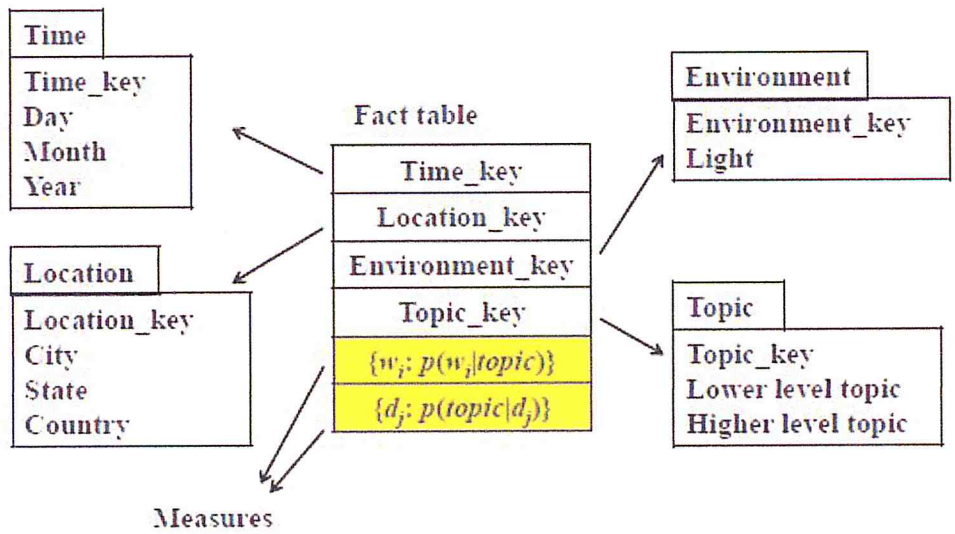

Fig. 3.2 : schéma en étoile d'un Topic Cube Zhang et al. (2009)

Bautista et al. (2010) ont proposé un modèle multidimensionnel qui prend en charge les informations textuelles dans un entrepôt de données, en introduisant une dimension textuelle AP-Dimension obtenue par la transformation des données textuelles en une structure sémantique AP-Structure. Cette structure est basée sur les items fréquents nommés AP-sets (apriori sets) obtenus par l'application de l'algorithme apriori sur les attributs textuels d'une base de données transactionnelle, Bautista et al. (2006). Le cube de données résultant est un cube de données classique qui intègre une dimension textuelle, tandis que la mesure définie pour ce cube reste une mesure numérique.

Zhang et al. (2011) ont proposé MicroTextClusterCube, un modèle basé sur un schéma en étoile. Ils ont proposé d'introduire une nouvelle mesure d'analyse (meani, sizei) qui eprésente respectivement le vecteur mean (un vecteur de termes pondérés) et la taille d'un micro-cluster, où un micro-cluster est une cellule texte qui permet de compresser les documents similaires (chaque cellule texte contient un certain nombre de documents). Cette compression (en microcluster) permet de retenir des informations sémantiques essentielles sur les cellules textuelles.

Les travaux dans cette catégorie ont permis d'étendre les modèles d'entrepôts classiques pour assurer l'analyse des données textuelles. Ils ont basé leurs modèles sur les deux concepts : fait et dimension. Ces modèles ont pu traiter la sémantique des données textuelles à travers l'intégration d'une dimension sémantique. Tandis que l'aspect structurel n'a pas été pris en compte, ce qui ne permet pas de faire des analyses sur de différents niveaux structurels. [4]

1.1.2) Modèles à nouveaux concepts :

La complexité des données textuelles et la limite des modèles classiques ont poussés les chercheurs à proposer de nouveaux modèles basés sur de nouveaux concepts. Parmi ces travaux nous citons Khrouf et Dupuy (2001) qui ont proposé un modèle d'entrepôt de documents basé sur le paradigme objet. Leur modèle est basé sur deux concepts : la structure logique générique et la structure logique spécifique. La première décrit les structures logiques communes à un ensemble de documents. Elle regroupe ainsi toute une classe de documents ayant des structures logiques

identiques ou similaires. La deuxième correspond à une spécialisation de la structure logique générique, elle est unique et correspond à un et un seul document. Leur processus d'analyse consiste à gérer à partir de l'entrepôt, le schéma du magasin de documents désiré. Ce processus se compose de quatre étapes : (1) le choix du type d'analyse qui permet à l'utilisateur de choisir un type d'analyse, il s'agit de décider de travailler sur des documents ayant des structures similaires ou différentes ou même sur un seul document. (2) la sélection des composants d'analyse, qui consiste à sélectionner les faits, les mesures ainsi que les dimensions d'analyse. (3) le filtrage qui permet à l'utilisateur d'affiner ses analyses en sélectionnant des valeurs précises. (4) la visualisation qui consiste à restituer le schéma du magasin des documents selon une représentation graphique facilitant les analyses multidimensionnelles. Khrouf et Dupuy (2005).

Tounier (2007), a proposé le modèle en Galaxie défini par un nouveau concept galaxie. Une galaxie est définie comme étant un regroupement de dimensions liées entre elles par un ou plusieurs nœuds centraux ; chaque nœud modélise les dimensions compatibles pour une même analyse. Son modèle est basé sur la généralisation du concept de constellation de Kimball (1996). Cette approche consiste à décrire un schéma multidimensionnel par l'unique concept de dimension où la notion de fait est supprimée. Afin de permettre l'analyse des documents textes, Tournier a introduit un nouveau concept hiérarchie structurelle de dimension documentaire qui permet de faire des analyses OLAP sur différents niveaux hiérarchiques des documents XML(section, paragraphe,...), il a aussi 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.

Dans leur modèle multidimensionnel d'objets complexes, Boukraa et al. (2011) se sont basés sur le paradigme objet grâce auquel il est possible de représenter les objets de l'univers et de capter la sémantique qu'ils véhiculent, notamment dans les liens avec les autres objets. Ainsi ils modélisent le monde réel par un ensemble d'objets complexes qui décrivent les entités de ce dernier. Le modèle d'objets complexes est un modèle à trois niveaux : 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 complexe sont regroupées en un seul package, pour fournir à la fin un diagramme de packages décrivant des objets complexes. 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 complexes liés à l'objet fait par des relations complexes. Chaque objet complexe dans leur modèle peut être défini grâce à leur 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. Leur modèle permet aussi une analyse sur de différents niveaux de granularité de chaque objet complexe. Fig. 3.3 nous illustre les trois niveaux présents sur le modèle d'objet complexe, le premier niveau est représenté par la figure c, le deuxième par la figure b et le troisième par la figure a :

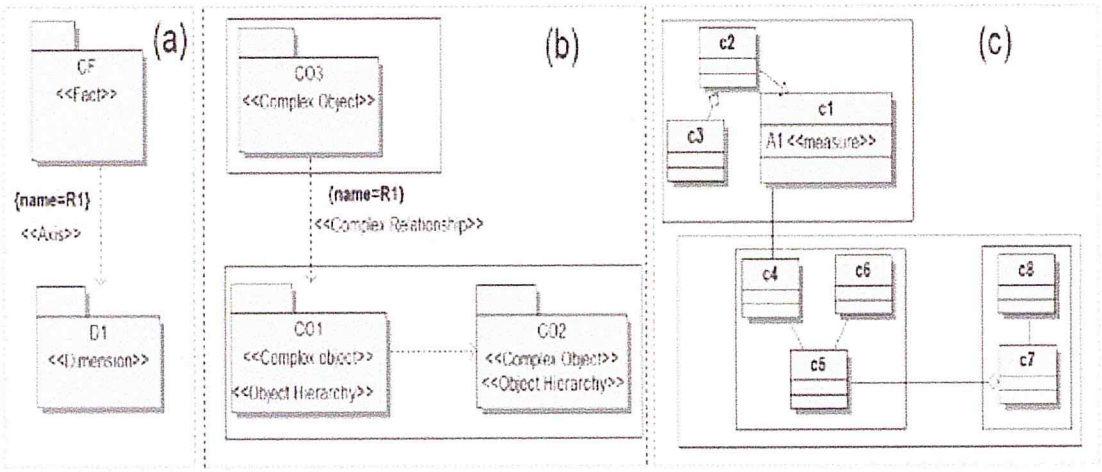

Fig. 3.3 : Modèle multidimensionnel d'objets complexe à trois niveaux Boukraa (2013)

Les travaux dans cette catégorie ont proposé de nouveaux concepts pour répondre à la complexité des données textuelles. Leurs modèles prennent en compte l'aspect structurel des documents en considérant le document textuel comme étant un ensemble de mots ayant déjà une certaine structure, ce qui permet une analyse sur différents niveaux structurels. La sémantique des données textuelles est prise en

compte grâce à l'utilisation d'une ontologie dans certains modèles, toutefois elle reste toujours peu exploitée. L'ensemble de ces modèles ne permettent pas une analyse sur de différents niveaux sémantiques. [4]

1.2) Étude comparative :

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. Nous présentons dans cette partie une étude comparative entre les différents modèles cités auparavant. Nous comparons l'ensemble de ces modèles par rapport à la prise en compte des cinq aspects suivants, qui nous ont permis de les étudier :

1.2.1) L'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 la structure interne des documents. Toutefois, cette approche de modélisation ne répond pas à toutes les exigences d'un décideur, tel que l'analyser 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 qui divise le document en plusieurs niveaux hiérarchiques, ce qui permet une analyse sur de différents niveaux de granularité, n'est pas prise en considération. Ainsi ils définissent un modèle qui prend en compte l'aspect structurel des documents, et permettant une analyse multidimensionnelle sur de différents niveaux structurels.

1.2.2) L'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.

1.2.3) La 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, Tounier (2007). 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.

1.2.4) 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.

1.2.5) 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.

Nous présentons dans le tableau ci-dessous, une étude comparative des modèles présentés dans les sections précédentes.

Modèles d'entrepôts de textes	Familles de modèles		Mesure texte	Opérateurs OLAP		Aspect Sémantique	Aspect structurel	Flexibilité d'analyse
	Modèles extensifs	Modèles à nouveaux concepts		Fonctions d'agrégation	Opérateurs de navigation			
E.documents Khrouf et Dupay (2001)		X	-	-	-	-	X	Bonne flexibilité
Mire Lee et al. (2002)	X		X	-	Drill down et Roll-up	-	-	Non flexible
DocCube Mothe et al. (2003)	X		-	Score(DJ	Drill down et Roll-up	X	-	Non flexible
D.cube Tseng et al. (2006)	X		-	Count	Drill down et Roll-up	-	-	Non flexible
Galaxie Tounier (2007)		X	X	AVG-KW, Top-KW	Drill down et Roll-up	X	X	Bonne flexibilité
TexCube Lin et al. (2008)	X		X	-	pull-up et push-down	X	-	Non flexible
TopicCube Zhang et al. (2009)	X		X	-	Drill down et Roll-up	X	-	Non flexible
MMAP-structure Bautista et al. (2010)	X		-	-	-	X	-	Non flexible
M.Cube Zhang et al. (2011)	X		-	-	-	X	-	Non flexible
MMOC Boukraa et al. (2011)		X	X	utilisation du Top-KW	Drill down et Roll-up	-	X	Bonne flexibilité

Tab. 3.1 : Tableau comparatif

Bien 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 complexité liés à l'analyse de ce type de données, tel 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. De plus, ces modèles ne sont pas génériques et ne permettant pas de représenter n'importe quelle données textuelles. Par contre, les modèles à nouveaux concepts ont permis de traiter d'autres problèmes d'analyse textuelle tel 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 restent 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. [4]

2) Le modèle sémantique multidimensionnel des objets de texte (The Multidimensional Semantic Model of TextObjects) :

2.1) Travaux connexes :

La modélisation multidimensionnelle consiste à organiser les données afin que les applications OLAP soient efficientes et efficaces. Les modèles d'entrepôt existants offrent une structure pour une modélisation multidimensionnelle des données simples, mais ils ne sont pas adaptés pour les données textuelles. Pour résoudre ce problème, plusieurs études ont été mises au point. Ces études peuvent être regroupées en deux catégories [4]. La première catégorie comprend des modelés extensifs, qui ont proposé d'étendre les modèles d'entrepôt traditionnelles pour permettre l'analyse des données textuelles. Ils sont basés sur les deux concepts de base : fait et dimension. Certains de ces modèles ont proposé d'étendre le cube de données classique en intégrant une dimension sémantique, comme la hiérarchie des topics dans le cube des sujets (Topic cube) [13], et la hiérarchie des termes dans le cube de texte (Text cube) [9] Et une AP-structure sur la base des éléments fréquents nommés AP-ensembles

(AP-Sets) [5]. D'autres travaux ont basé leurs modèles sur la technique de classification ; deux œuvres majeures représentatives de ce groupe sont : cube de documents (Doc cube) [10] et microtextcluster [14]. Ces modèles ont été en mesure de traiter la sémantique de données textuelles en intégrant une dimension sémantique. L'aspect structurel n'a pas été pris en compte, donc, ces modèles ne supportent pas l'analyse à différents niveaux structurels, tels que l'analyse d'une section spécifique (ex : sports) dans un journal. Les modèles de cette catégorie sont basées sur le modèle d'entrepôt traditionnel, conduisant à un manque de flexibilité d'analyse. La deuxième catégorie comprend des modèles avec de nouveaux concepts, à plus tard, de nouveaux modèles basés sur de nouveaux concepts ont été proposés. Les deux modèles les plus connus sont : le modèle galaxie (galaxie model) [12] et le modèle des objets complexes (complex objects model) [7]. La première est basée sur la généralisation de la notion de constellation [8]. Le deuxième modèle est basé sur la notion d'objet complexe. Ce modèle permet une analyse à différents niveaux de granularité de chaque objet complexe, tandis que l'aspect sémantique de données complexes n'est pas pris en charge. Les modèles de cette catégorie considèrent les aspects structurels des documents qui permettent une analyse en différents niveaux structurels, tandis que l'aspect sémantique est encore inexploité. Ces modèles offrent une bonne flexibilité d'analyse, mais ils ne permettent pas de définir le contenu sémantique comme une mesure ou une dimension à la fois.

2.2) Modèle sémantique multidimensionnel des objets de texte :

Dans cette approche de modélisation, ils se sont concentrés sur trois aspects : (i) la structure des données textuelles, (ii) la sémantique des données textuelles et (iii) la flexibilité d'analyse. Ce modèle propose une analyse approfondie sur les différents niveaux sémantiques et structurels à travers deux types d'hiérarchies : la hiérarchie structurelle et la hiérarchie sémantique. Il fournit également une flexibilité d'analyse par le biais d'un opérateur pour la construction de cube de texte sémantique. [15]

2.2.1) Définition des concepts :

Dans cette section, nous allons présenter les concepts de base définis pour le modèle MSMTO : [15]

Text Object (TObjt) :

un objet de texte (Text Object) est une entité représentant un élément de texte (ex : document texte) qui peut être analysé comme un fait ou une dimension d'analyse.

Un objet de texte est défini par un ensemble d'attributs simple qui sont des attributs des classes UML. Les attributs de l'objet texte représentent les données extraites de documents textuels. [15]

Définition 1 : Un objet texte est noté $TObjt$ et est défini comme suit :

$$TObjt = (IDTObjt, SATObjt) \qquad (1)$$

Avec : $SATObjt = \{ATObjt_1, ATObjt_2, \dots, ATObjt_n / n \in \mathbb{N}\}$

Où : $IDTObjt$ représente l'identifiant de l'objet texte et $SATObjt$ représente l'ensemble de ses attributs. [15]

Semantic Content Object ($SCObjt$) :

Il s'agit d'une entité représentant le contenu sémantique d'un objet texte (Text Object). Il est obtenu par l'application d'une méthode d'extraction de sémantique sur l'objet texte cible. Des termes ou des sujets pertinents sont des exemples d'objet contenu sémantique ($SCObjt$). Cela est défini plus tard par un jeu d'attribut et une seule méthode d'agrégation sémantique, qui fonctionne sur elle afin de produire d'autres $SCObjt$ avec la plus grande granularité. Par exemple, Les objets termes de la classe représentent un $SCObjt$ de haut niveau de granularité. Il est obtenu par l'application de la fonction d'agrégation AVG KWD [12] sur les termes de $SCObjt$. Il convient de noter que $SCObjt$ de bas niveau de granularité est associé au niveau 0, dans l'exemple précédent les objets termes sont associés au niveau 0. [15]

Définition 2 : Un objet de contenu sémantique est noté $SCObjt$ et est défini comme suit :

$$SCObjt = (IDSCObjt, SASCObjt, MSCObjt) \quad (2)$$

Avec : $SASCObjt = \{ASCObjt_1, ASCObjt_2, \ldots., ASCObjt_n / n \in \mathbb{N}\}$

Où : $IDSCObjt$ représente l'identifiant de l'objet de contenu sémantique, $SASCObjt$ représente l'ensemble de ses attributs et $MSCObjt$ représente une fonction (méthode) d'agrégation. [15]

Fig. 3.4 montre le méta modèle sémantique multidimensionnel d'objets textuels décrit par un diagramme de classes UML. [15]

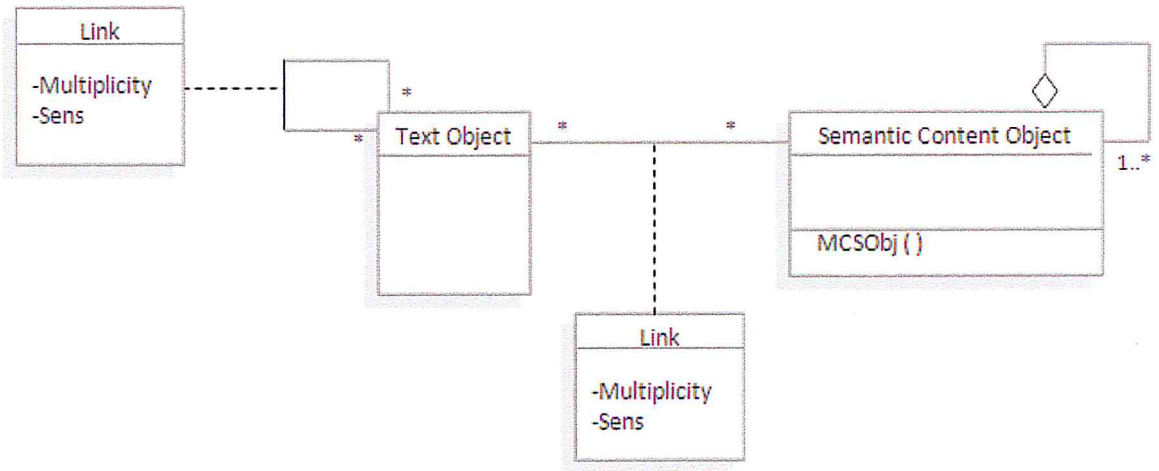

Fig. 3.4 : Le méta modèle sémantique des objets textuels

Ce modèle est décrit comme suit : [15]

- La classe Text Object représente l'objet du texte
- La classe Smantic Content Object représente l'objet de contenu sémantique
- L'association Link$(TObjt, TObjt)$ représente le lien entre différents objets textuels tels que : agrégation, association, composition ...
- L'association $Link(SCObjt; SCObjt)$ représente le lien d'agrégation entre les objets de contenus sémantiques.
- L'association $(TObjt; SCObjt)$ représente le lien qui associe chaque objet textuel à son objet de contenu sémantique.

Exemple :

Un exemple d'un objet de texte est un journal de presse. Un journal de presse peut être décrit par un ensemble d'attributs, tels que : nom du journal, type, éditeur ... etc. Un journal peut être décrit par un ensemble d'objets textuels comme : sections, articles. Les topics associés à des objets textuels (Journal, section, articles) sont un exemple d'objet de contenu sémantique $(SCObjt)$. Un exemple de lien $(SCObjt, SCObjt)$ est la relation d'agrégation entre l'objet contenu sémantique Topics et l'objet contenu sémantique Topics family. La fonction AVG KW [11] utilisée pour grouper les mots clés dans un seul plus général est un exemple d'une fonction d'agrégation appliquée à l'objet Terms. (Fig. 3.5). [15]

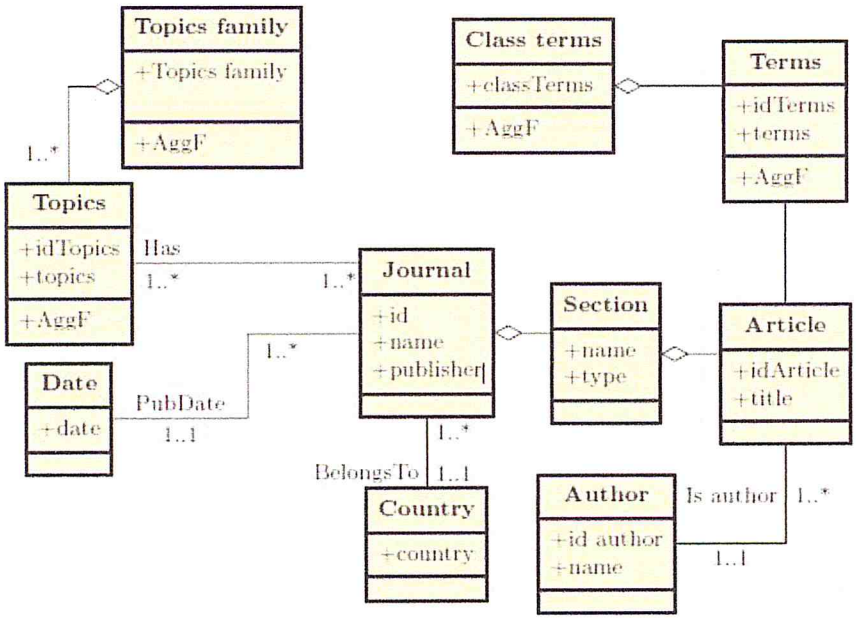

Fig. 3.5 : Représentation d'un journal par le modèle sémantique multidimensionnel des objets texte (MSMTO)

Relation complexe :

Cette relation modélise les liens entre les objets textuels au-delà ceux d'agrégation, généralisation et composition d'une part, et les liens entre les objets textuels et les objets de contenu sémantique de l'autre part. Sa complexité réside dans le fait qu'elle définit les axes d'analyse et les mesures analytiques de certains objets par rapports à d'autres. (Fig. 3.6) [15]

Définition 3 : Une relation complexe est noté R et est définie comme suit : [15]

$$R = (TObjt^R, Object^R) \quad (3)$$

Où : $Object \in \{TObjt, SCObjt\}$

Et $R \notin \{Aggregation, composition, generalisation\}$

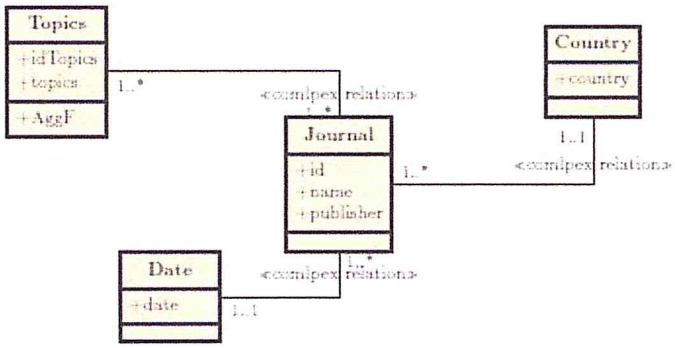

Fig. 3.6 : Un exemple de relation complexe pour l'objet Journal

Relation complexe étendue :

Elle modélise les liens non directs entre les objets, par exemple, le lien entre l'objet Article et l'objet Date de la Fig. 3.5. Elle définit également les axes d'analyse de certains objets par rapport à d'autres. [15]

Définition 4 : Une relation complexe étendue est notée ER et est définie comme suit : [15]

$$ER = (TObjt^{ER}_{Source}, TObjt^{ER}_{Cible}) \quad (4)$$

De telle sorte que : $Relation(TObjt^{Relation}_{Source}, TObjt^{Relation}_{x})$

Et : $R(Objt^R_x, TObjt^R_{Cible})$

Où : R est une relation complexe (voir (3))

$ER \notin \{Agregation, composition, generalisation\}$

Et : $Relation \in \{Agregation, composition, generalisation\}$

Hiérarchie structurelle :

Elle est définie entre plusieurs objets texte. Elle permet d'effectuer des opérations d'agrégation entre des objets textuels en fonction de leur structure. La hiérarchie structurelle définit un ordre partiel entre certains objets textuels en fonction de leur degré de granularité. (Fig. 3.7) [15]

Définition 5 : La hiérarchie structurelle est notée StH et est définie comme suit : [15]

$$StH = \{TObjt_1, TObjt_2, \ldots\ldots, TObjt_n / n \in \mathbb{N}\} \cup \{AllObjt\} \qquad (5)$$

Où : $AllObjt$ est un objet artificiel avec la granularité la plus basse.

Hiérarchie sémantique :

Une hiérarchie sémantique est définie entre plusieurs objets de contenu sémantique. Il s'agit d'un type particulier de hiérarchie qui établit des agrégations sémantiques entre les objets de contenu sémantique. ils définissent une fonction $\mathrm{SCObjtLevel}\,(\mathrm{SCObjt};\ \mathrm{SH})$, qui renvoie le niveau de chaque objet de contenu sémantique dans la hiérarchie sémantique. ils supposent que le niveau de l'objet le plus détaillé de la hiérarchie est affecté à 0, cet objet est associé à un objet de texte. (Fig. 3.8) [15]

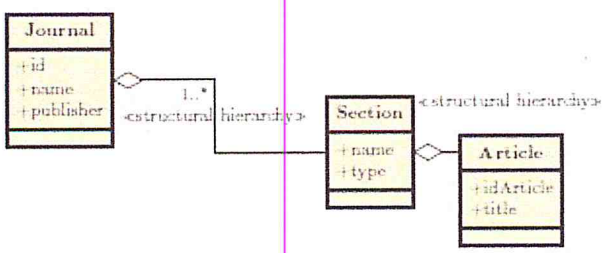

Fig. 3.7 : Un exemple d'hiérarchique structurelle

Définition 6 : Une hiérarchie sémantique est notée SH et est définie comme suit : [15]

$$\mathrm{SH} = \{\mathrm{SCObjt}_1, \mathrm{SCObjt}_2, \ldots\ldots, \mathrm{SCObjt}_n / \mathrm{n} \in \mathbb{N}\} \qquad (6)$$

De telle sorte que : $\mathrm{R}(\mathrm{TObjt}^{\mathrm{R}}, \mathrm{SCObject}^{\mathrm{R}}) \Rightarrow \mathrm{SCObjtLevel}(\mathrm{SCObjt}, \mathrm{SH}) = 0$

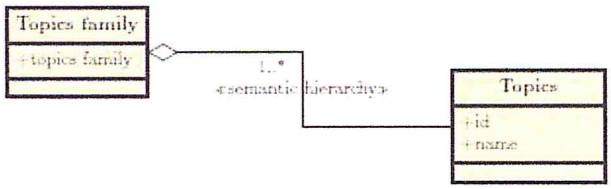

Fig. 3.8 : Un exemple de hiérarchie sémantique

2.2.2) Cube de texte sémantique : ST-Cube (Semantic Text Cube) :

Afin d'effectuer une analyse multidimensionnelle dans ce modèle, ils définissent un cube de texte sémantique comme suit: Un cube de texte sémantique est construit à la base de modèle sémantique multidimensionnel des objets textuels; il permet de modéliser un sujet d'analyse appelé FactObject défini par plusieurs dimensions d'objet textuel $\mathrm{TODim_i}$, $\mathrm{i} \in [1..*]$ et des dimensions de l'objet sémantique $\mathrm{SODim_j}, \mathrm{j} \in [1..*]$. La dimension objet textuel comprend des attributs $\mathrm{TA} = \langle \mathrm{ta_1}, \mathrm{ta_2}, \ldots\ldots, \mathrm{ta_*} \rangle$ et elle est associée à une hiérarchique structurelle composée d'objets textuels $\mathrm{StH} = \langle \mathrm{to_1}, \mathrm{to_2}, \ldots, \mathrm{to_*} \rangle$. Une dimension d'objet sémantique inclut des attributs $\mathrm{SA} = \langle \mathrm{sa_1}, \mathrm{sa_2}, \ldots\ldots, \mathrm{sa_*} \rangle$ et elle est associée à une hiérarchie sémantique, composée d'objets sémantiques $\mathrm{SO} = \langle \mathrm{so_1}, \mathrm{so_2}, .., \mathrm{so_*} \rangle$. Un cube de texte sémantique est défini par un ensemble d'objets de dimensions textuelles, objets de dimensions sémantiques, objet fait et un ensemble de mesure. Forent en bas et roulent le long des objets de dimensions textuelles permettra aux utilisateurs de faire des analyses sur différents niveaux hiérarchiques. Forent en bas et roulent le long des objets de dimensions sémantiques permettra une analyse sémantique à différents niveaux de granularité. Ce cube de texte sémantique peut sauvegarder différents types de mesures : mesure numérique, mesure textuelle et mesure sémantique. [15]

Opérateur de construction d'un cube de texte sémantique :

Le modèle proposé offre une bonne flexibilité d'analyse, non seulement en donnant aux utilisateurs la possibilité de définir le fait sur l'opérateur de la construction de cube, mais en donnant la possibilité de considérer le contenu sémantique de données textuelles comme : un axe d'analyse, un fait ou même une mesure pour un objet fait textuel. Le processus de création de cube de texte sémantique est de : (i) sélectionner un objet du modèle sémantique multidimensionnel des objets textuels et lui attribuer

le rôle de fait, cet objet sera appelé objet fait. (ii) Sélectionner un ensemble d'objets et assigner les le rôle des dimensions, ces objets seront appelés des objets de dimensions. L'objet fait peut-être : un objet de texte ou un objet de contenu sémantique qui nous donne deux cas : [15]

Cas 1 : L'objet fait peut être un objet de texte ou un objet de contenu sémantique ce qui nous donne deux possibilités :

- La mesure est un simple attribut de l'objet fait. Dans ce cas, l'objet contenu sémantique lié à l'objet textuel (choisi comme fait) par une relation complexe, sera considéré comme un axe d'analyse.

- La mesure est un objet contenu sémantique. Dans ce cas, la relation complexe entre l'objet textuel et l'objet de contenu sémantique (choisi comme mesure) n'est pas pris en charge lors de la sélection des objets de dimension.

Cas 2 : L'objet fait c'est un objet de contenu sémantique. Dans ce cas, la mesure sera un simple attribut de l'objet de contenu sémantique.

Pour la représentation graphique, ils utilisent le package des diagrammes UML pour représenter leur ST-Cube, où :

- Un package représentant un objet fait contient l'objet fait et la mesure d'analyse.

- Un package représentant un objet dimension contient l'objet dimension.

- Les liens entre les packages sont appelés relation complexe et relation complexe étendu. Fig. 3.9 montre un schéma multidimensionnel pour l'analyse d'un journal. Il est obtenu en appliquant l'opérateur de construction de ST-cube, et en sélectionnant l'objet journal comme un objet fait et l'attribut du journal comme une mesure. Dans ce cas, l'objet Topics sera considéré comme un objet dimension. Le schéma de la Fig. 3.10 présente une autre possibilité d'analyse, il en résulte en appliquant l'opérateur de construction de ST-Cube, définissant l'objet Articles comme un objet fait et l'objet Terms comme une mesure. L'objet fait résultant dans ce cas est représenté sur la Fig. 3.10. [15]

Algorithme de l'opérateur de Construction de ST-Cube :

L'algorithme prend en entrée une liste d'objets textuels et d'objets de contenu sémantique qui sera appelée $\text{Object}_{\text{List}} = \{\text{Obj}_1; \text{Obj}_2; \text{Obj}_3; \ldots.; \text{Obj}_n\}/n \in \mathbb{N}$. Il produit un cube de texte sémantique. L'algorithme utilise les fonctions et procédures suivantes : [15]

GetHDimension (DimensionObject) : est une fonction qui prend en entrée un objet de dimension et renvoie un ensemble d'objets qui ont une liaison directe avec elle de telle sorte que : $\text{Realtion}(\text{Object}^{\text{Relation}}; \text{DimensionObject}^{\text{Relation}})$ (voir (4)) [15]

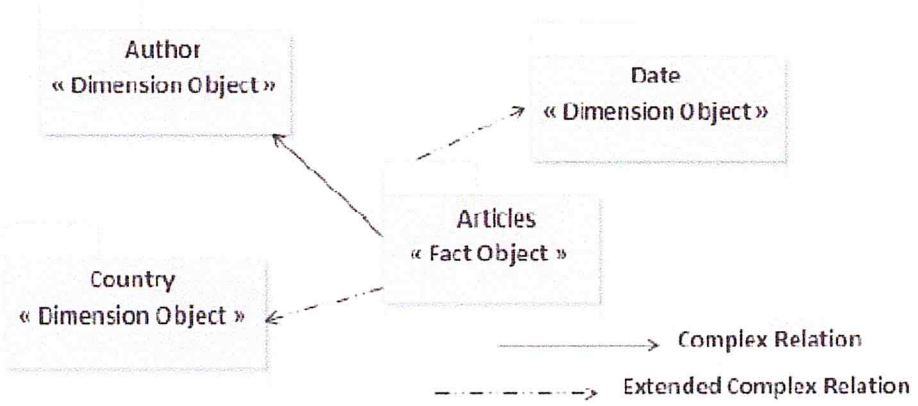

Fig. 3.9 : Un exemple pour un schéma multidimensionnel pour l'objet Articles

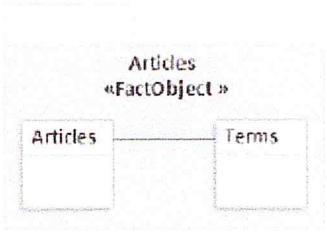

Fig. 3.10 : Un package détaillé représentant les articles de l'objet fait

GetDimensionSet(FactObject) :

Est une fonction qui prend en entrée un objet fait et renvoie l'ensemble des objets associés à cet objet à travers une relation complexe définie en (3) et une relation complexe étendue définie en (4). [15]

SetAgregationFunction(Function) :

Est une fonction qui attribue à chaque mesure analytique une fonction d'agrégation "Function". [15]

Select (mesure, AttributeSet) :

Est une fonction qui attribue à la mesure la valeur d'un attribut à partir de AttributeSet. [15]

Algorithm 1 ST-Cube Construction Part 1(Called functions)

```
1: function GETHDIMENSIONSET(DimensionObject)
2:     for each object From Object_List do
3:         if Relation(Object^Relation, DimensionObject^Relation) then
4:             HDimensionObject.add(Object);
5:         end if
6:     end for
7: end function
8: function GETDIMENSIONSET(Object)
9:     for each object From Object_List do
10:        if R(Object^R, DimensionObject^R) then
11:            DimensionObject.add(Object);
12:        end if
13:    end for
14: end function
```

Algorithm 1 ST-Cube Construction Part 2 (Main Program)

```
Input: Object_List={Obj1.Obj2.....Objn}
Output: Semantic TextCube
begin
FactObject:=SelectFact(Object_List):
if FactObject = TextObject then                ▷ The fact object is a text object
    Define measure ;
    if measure is TextObjectAttribute then
        DimensionSet:=GetDimensionSet(FactObject);
        for all DimensionObject in DimensionSet do
            GetHDimension(DimensionObject);
        end for
    else
        if measure is SemanticContentObject then
            object:=SemanticContentObject;
            measure:= Select(measure,SemanticContentObjectAttributes);
            DimensionSet:=GetDimensionSet(FactObject)-{object};
            for  all DimensionObject in DimensionSet do
                GetHDimension(DimensionObject);
            end for

        end if
    end if
else                                   ▷ the FactObject is a semanticContentObject
    measure:= Select(measure,SemanticContentObjectAttributes);
    DimensionSet:=GetDimensionSet(FactObject);
    for all DimensionObject in DimensionSet do
        GetHDimension(DimensionObject);
    end for
    measure.setAgreegationFunction(Function);
end if
End
```

Conclusion :

Dans la première partie de ce chapitre, la présentation d'un ensemble de modèles qui servent à la représentation multidimensionnelle des données textuelles, et la comparaison entre eux étaient faites. On a remarqué que ces modèles sont plus ou moins différents, on peut différencier entre eux selon qu'ils soient des modèles extensifs ou des modèles à nouveaux concepts, ou bien selon différents critères (la prise en compte de la sémantique des données textuelles, la flexibilité d'analyse, ...).

Pour la deuxième partie de ce chapitre, on a bien expliqué le modèle MSMTO. On peut remarquer facilement que ce modèle est puissant par rapport aux modèles présentés dans la première partie de ce chapitre, puisque : (1) Il est basé sur le paradigme objet. (2) Il intègre un nouveau concept (Objet du contenu sémantique) utilisé pour représenter et organiser la sémantique des données textuelles dans un format hiérarchique, afin de permettre une analyse sémantique à différents niveaux de granularité. (3) Il considère la composition interne des documents textuels comme une hiérarchie structurelle, qui permet à l'utilisateur d'effectuer des analyses sur des niveaux hiérarchiques différents. (4) Il offre également la flexibilité, en tenant compte du contenu sémantique de données textuelles comme une mesure, un fait ou même une dimension.

Chapitre 4

Implémentation

Introduction :

Ce chapitre est très important, puisque on y trouve tout ce qui concerne la mise en œuvre de notre application. On commence par la définition des différents outils qu'on a utilisés pour implémenter notre application. Ensuite, on montre les différentes fonctionnalités de l'application à travers des photos d'écran.

Il est nécessaire de rappeler que le modèle d'entreposage des données textuelles utilisé est le modèle MSMTO [chapitre 3]. Le langage de programmation utilisé pour le développement est le langage Java SE.

On doit dire que pour simplifier notre application, on n'a pas pris en compte un des avantages du modèle MSMTO [chapitre 3], c'est la prise en compte de l'aspect structurel des données textuelles, on a considéré que les documents de notre corpus contiennent seulement des articles de journaux.

1) Présentation de l'API TextWise :

1.1) Définition :

TextWise est un chef et innovateur dans les technologies sémantiques. Ses scientifiques, ingénieurs et analystes ont développé et amélioré la science de recherche sémantique et outils analytiques depuis 1994. Aujourd'hui, ses outils sont largement utilisés pour distiller les significations de documents complexes.

L'API de TextWise s'appuie sur la puissance de la technologie de Semantic Gist®. D'abord présenté au public au début de 2008, son héritage, la technologie de Trainable Semantic Vectors (TSV) a permis des applications d'utiliser le sens du texte pour alimenter des recherches de similarité très pertinentes entre le contenu de la requête et le contenu indexé. En 2012, TextWise introduit une technologie sémantique nettement améliorée, Semantic Gist®. Cette technologie, basée sur la modélisation du langage et les réseaux de neurones, a été intégrée de façon transparente dans l'API de TextWise.

Semantic Gist® propulse le concept de marquage (Concept Tagging) et la catégorisation en utilisant un API simple, mais puissant.

Premiers pas avec l'API est aussi simple que d'enregistrer avec TextWise.com. Une fois votre inscription est validée, vous recevrez un jeton de l'API (a token). En utilisant ce jeton, vous pouvez commencer à faire des appels à chacun des services définis ci-dessous et commencer à intégrer la technologie de Semantic Gist® dans vos applications immédiatement.

Les technologies sémantiques de TextWise propulsent trois services distincts, parmi lesquels le service de catégorisation qu'on a utilisé pour notre projet. [W2]

1.2) L'API de TextWise en action :

La catégorisation sémantique de TextWise produit une analyse chargée d'un poids d'étiquettes et sujets basée sur tout genre de texte. Voici un bref exemple de la technologie :

Exemple de texte d'entrée (les URLs sont aussi acceptées) :

Since I've had my laptop pc running Linux, the battery life has given me around 2-3 hours. Recently, my battery barely gives me 1 hour. How could that happen ? Any suggestions on how to fix it ? I rarely use the battery ; it's usually plugged into the AC adapter. Thanks in advance.

Poids : LONGUES Catégories :

0.6307512 Business/Electronics_and_Electrical/Power_Supplies

0.45559603 Computers/Hardware

En raccourcissant les étiquettes qu'elle produit, la catégorisation de TextWise peut vous aider aussi à concentrer sur le spécifique. Par exemple :

Étiquette de LONGUE catégorie : Recreation/Outdoors/Camping

Étiquette de COURTE catégorie : Outdoors/Camping

La catégorisation sémantique de TextWise est idéale pour toute organisation qui veut conduire des requêtes et des clients vers les meilleures ressources et solutions possibles. [W2]

2) Présentation de DMOZ :

2.1) Définition :

DMOZ, abréviation de Directory Mozilla qui donne son nom au site, dmoz.org, anciennement l'Open Directory Project ou ODP, est un répertoire de sites web, et probablement le répertoire le plus pertinent sur Internet, créé en 1998, sous licence Open Directory.

Il a été fondé dans l'esprit de l'Open Source, et Il est géré par une vaste communauté d'éditeurs bénévoles provenant du monde entier, chacun étant responsable de vérifier l'exactitude et la catégorisation des sites dans une ou plusieurs catégories.

Au 17 février 2011, DMOZ contient dans son ensemble 4 838 135 ressources d'adresses de sites, classées dans plus de 1 005 146 catégories. Plus de 90 093 éditeurs ont participé au projet depuis son lancement, dont près de 8 000 sont toujours actifs. Le répertoire propose des ressources dans 78 langues.

Ce répertoire est très peu connu du grand public, mais la visibilité d'un site qui y est admis vient du fait que d'autres répertoires et moteurs de recherche le consulte. En effet, l'index de DMOZ est utilisé par Google, Netscape, AltaVista, Lycos, ... et une quantité d'autres portails, moteurs de recherche ou sites web. [W1]

2.2) La république du Web :

La croissance du Web se poursuit à une vitesse stupéfiante. Les moteurs de recherche automatisés ont de plus en plus de difficultés à fournir des résultats satisfaisants. Les petites équipes d'édition professionnelles travaillant sur les sites commerciaux des répertoires ne peuvent plus répondre aux requêtes, et la qualité et le contenu de leurs répertoires s'en ressentent. Les liens deviennent obsolètes et ne peuvent plus suivre le rythme de croissance de l'Internet.

Au lieu de combattre la croissance explosive de l'Internet, l'annuaire DMOZ permet à l'Internet de s'organiser. Parallèlement à la croissance de l'Internet, le nombre de citoyens du Web augmente. Ces citoyens sont habilités à organiser une petite partie du Web et à la présenter au reste de la population, en supprimant les éléments inutiles ou inintéressants, et en conservant les meilleurs éléments. [W3]

3) Présentation de la base de données native XML (eXist-db) :

3.1) Définition :

eXist-db est un système de gestion de base de données open source entièrement basé sur la technologie XML. Contrairement à la plupart des systèmes de gestion de base de données relationnelles, eXist-db utilise XQuery, qui est une recommandation du W3C, pour manipuler ses données. [W1]

3.2) Avantages d'eXist-db :

eXist-db permet aux développeurs la manipulation de données XML sans avoir à écrire de lourds programmes intermédiaires. eXist-db respecte et étend beaucoup des standards XML du W3C comme XQuery. eXist-db supporte aussi les interfaces REST pour interagir avec les formulaires web de type AJAX. Les applications telles que XForms sont susceptibles d'enregistrer leurs données par quelques lignes de codes. L'interface de WebDAV vers eXist-db permet aux utilisateurs de glisser/déposer directement des fichiers XML dans la base de données eXist-db. Parce que eXist-db indexe automatiquement les documents par un système de mots clefs, il est aisé de créer un système de recherche performant. [W1]

3.3) Standards et technologies eXist-db :

eXist-db utilise les standards et technologies suivants : [W1]

- XPath - Langage XML Path
- XQuery - Langage XML Query
- WebDAV - Web distributed authoring and versioning
- REST - Representational state transfer (URL encoding)
- SOAP - Simple Object Access Protocol
- XACML - XML Access Control Language
- XInclude - server-side include file processing (limited support)

- XML-RPC - a remote procedure call protocol
- XProc - a XML Pipeline processing language
- XUF - an XML Update Facility Extension to XQuery

4) Fonctionnalités de notre application :

Après la présentation des différents outils que l'on a utilisés pour implémenter notre application, on va éclaircir ses différentes fonctionnalités, en montrant un ensemble de figures, avec les explications nécessaires.

On suppose que notre corpus contienne des documents dont l'extension est TXT (ce sont des articles de journaux), donc, on ne s'intéresse pas, dans ce projet, à l'extraction du texte à partir d'un document qui peut contenir des images, par exemple, mais plutôt à l'extraction des informations que peut contenir un texte donné (comme la sémantique du texte).

Afin de préciser le Pays et la Date d'un document, on suppose que ces deux informations se trouvent dans la dernière ligne de chaque document, On a proposé ça au lieu d'utiliser les métadonnées, juste pour essayer l'approche d'extraction des informations qui existent déjà dans un texte.

4.1) Traitements avant l'analyse :

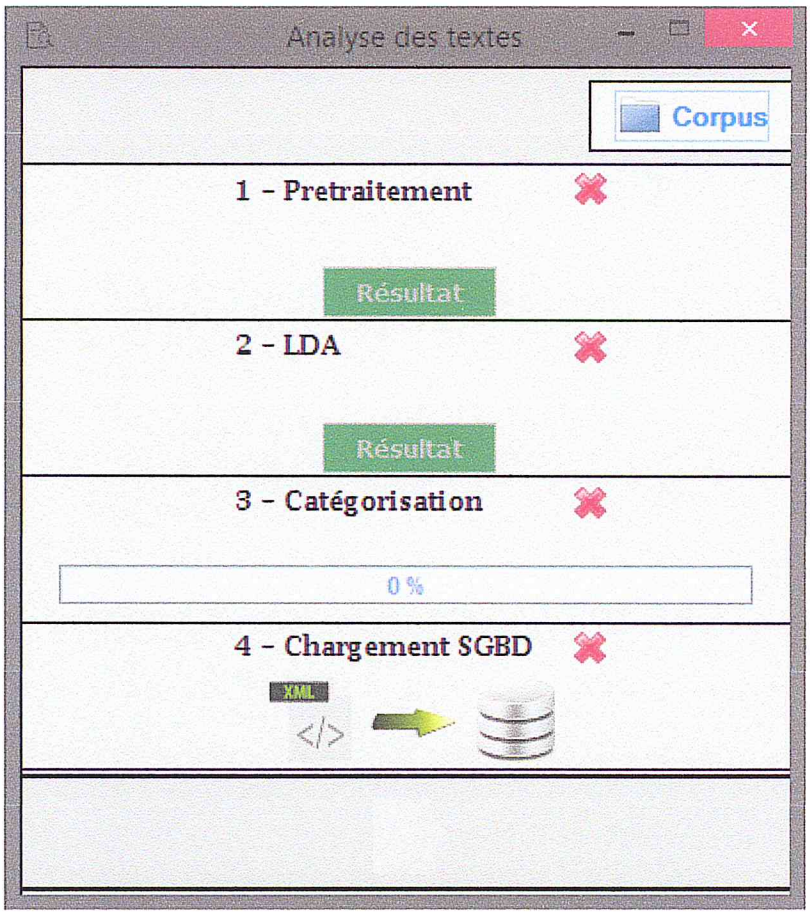

Fig. 4.1 : Traitements nécessaires avant l'analyse (Avant l'exécution du programme)

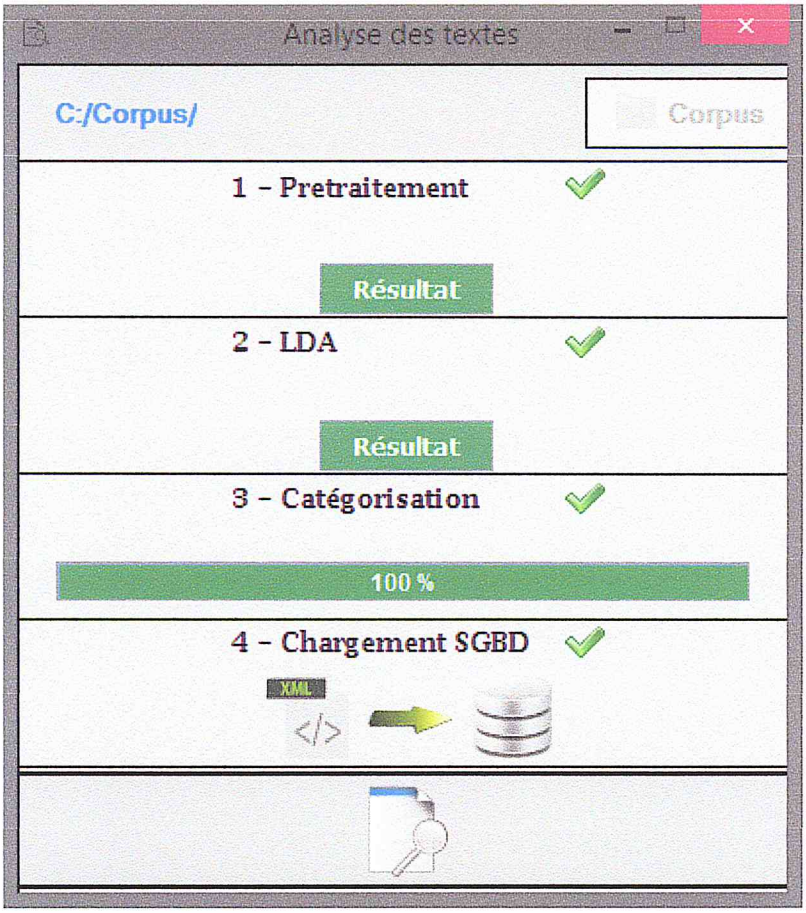

Fig. 4.2 : Traitements nécessaires avant l'analyse (Après l'exécution du programme)

Après le choix d'un corpus de documents à traiter (Fig. 4.2), les traitements suivants sont exécutés automatiquement, un par un :

4.1.1) Prétraitement :

Préparation des documents pour exécuter le programme LDA [chapitre 2].
Il consiste à :

- Eliminer les virgules, points d'interrogation,…
- Eliminer les Stop Words.
- Appliquer un algorithme de Stemmer (On a utilisé l'algorithme : Porter Stemmer, qui est très puissant).

4.1.2) LDA :

Génération des différents sujets (Topics) qui vont être utilisés dans l'étape de génération des catégories pour chaque document. On a utilisé LDA [chapitre 2] comme modèle de sujets, puisque, généralement, c'est le modèle le plus utilisé, à cause de sa puissance.

4.1.3) Catégorisation :

Après l'exécution de cette étape, des fichiers XML sont générés, ils contiennent les hiérarchies sémantiques de chaque document. Une hiérarchie sémantique c'est un ensemble de concepts sémantiques qui ont des relations père-fils entre eux.

L'API de TextWise nous donne une catégorie de chaque document, une catégorie est une hiérarchie de concepts. Les catégories données par l'API de TextWise sont les mêmes données par le site DMOZ.

Le site web DMOZ nous donne un ensemble de catégories (environ 100 catégories), pour chaque mot saisi dans son champ de recherche.

Si on a utilisé juste l'API TextWise, on obtient des catégories très générales. Alors, puisque les catégories de TextWise et de DMOZ sont les mêmes, on a proposé l'approche suivante pour obtenir des catégories détaillées pour les documents de notre corpus :

- On génère une catégorie de texte par l'API TextWise.
- On cherche les catégories de DMOZ de chaque mot donné par LDA [chapitre 2].
- A partir de l'ensemble des catégories de DMOZ, on garde juste les catégories qui se commencent par la catégorie de TextWise.
- On obtient alors plus de catégories pour chaque document.

Et puisque les catégories obtenues ont la même racine, on peut les combiner dans un arbre XML. Ce dernier est très adapté pour les données dont le format est hiérarchique.

Ces catégories sont ce que l'on a appelé dans le modèle MSMTO [chapitre 3] : les objets de contenu sémantique (hiérarchies sémantiques).

4.1.4) Chargement SGBD :

Les fichiers XML générés après l'étape de catégorisation seront chargés dans le SGBD native eXist-db.

Remarque :

On peut voir les résultats de Prétraitement, et de LDA [chapitre 2], en cliquant sur les deux boutons « Résultat ».

```
Topic 0th:
      languag 0.04800540906017579
      immigr 0.04124408384043272 4
Topic 1th:
      world 0.07382550335570469
      mosqu 0.04942037827943868
Topic 2th:
      scienc 0.0965133906013138
      knowledg 0.05103587670540677
Topic 3th:
      team 0.06362311801845556
      player 0.05876639145216124
Topic 4th:
      soccer 0.062355658198614314
      footbal 0.02386451116243264
Topic 5th:
      call 0.02027468933943754
      islam 0.02027468933943754
Topic 6th:
      english 0.03156274056966897
      todai 0.03156274056966897
```

Fig. 4.3 : Exemple de Résultat de LDA

Maintenant, et après tous ces traitements, on peut entamer l'étape d'analyse.

4.2) Analyse :

Vous pouvez voir la flexibilité d'analyse du modèle MSMTO [chapitre 3], en choisissant le concept sémantique comme une dimension ou comme une mesure d'analyse. Pour le premier cas, l'application affiche les documents. Pour le deuxième cas, l'application affiche les hiérarchies sémantiques des documents trouvés.

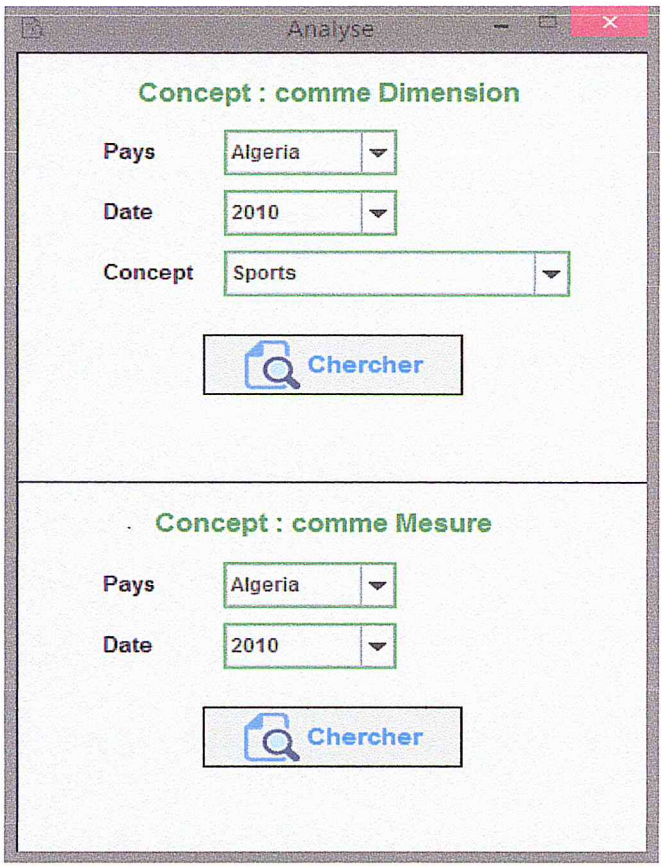

Fig. 4.4 : Fenêtre d'analyse

4.2.1) Concept comme Dimension :

Voici quelques tests :

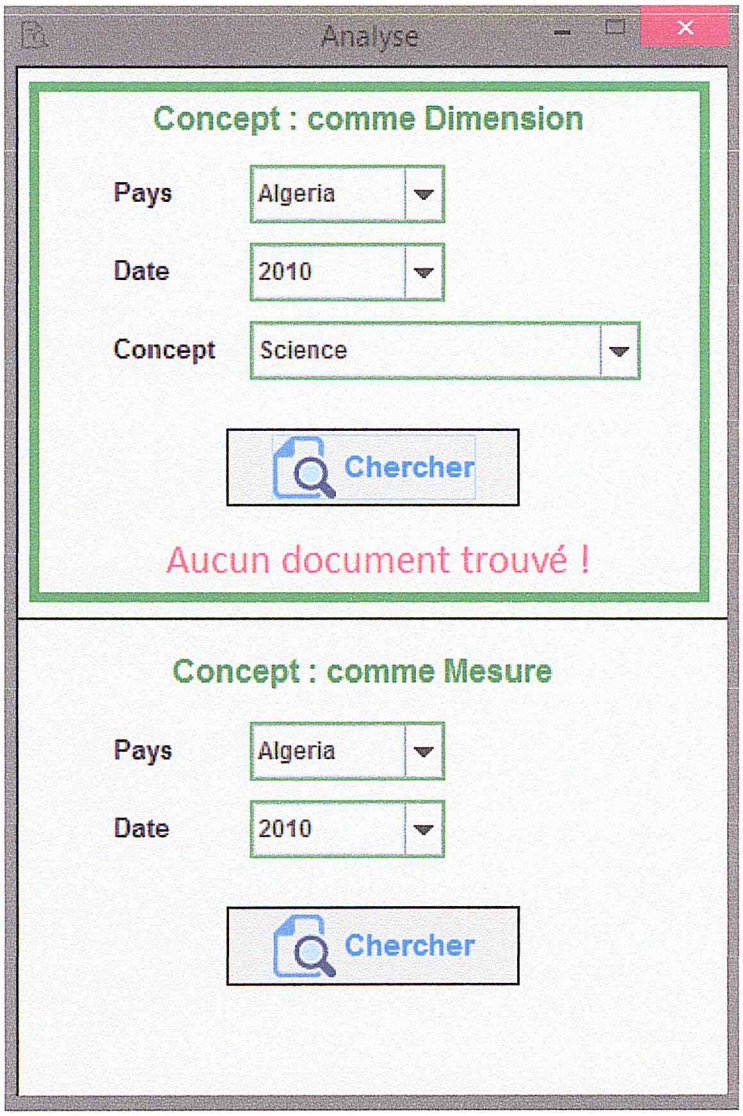

Fig. 4.5 : Aucun document trouvé en choisissant Concept comme Dimension

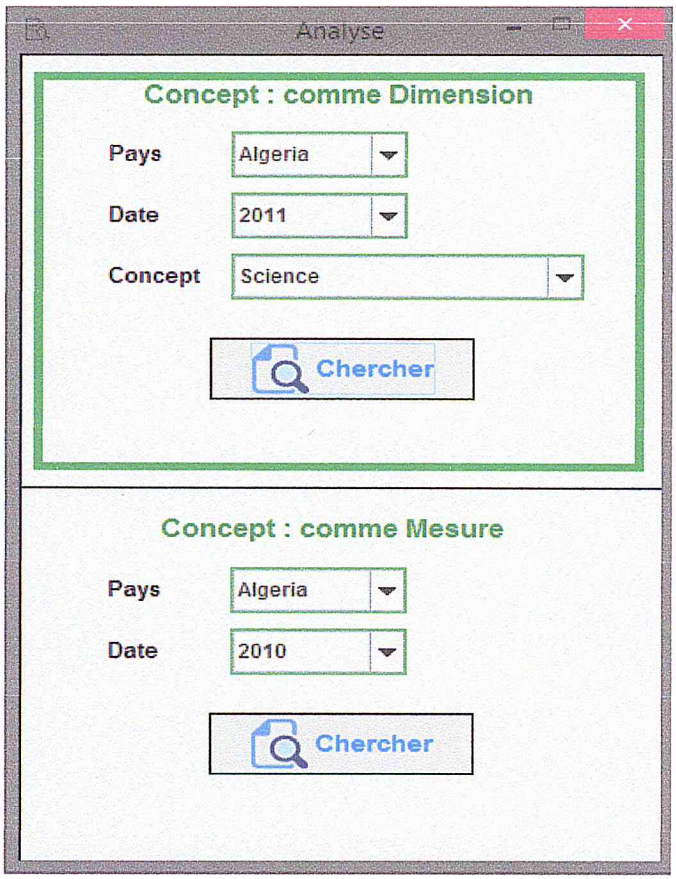

Fig. 4.6 : Choix de recherche pour Concept comme Dimension (Le résultat obtenu est montré dans la figure : Fig. 4.7)

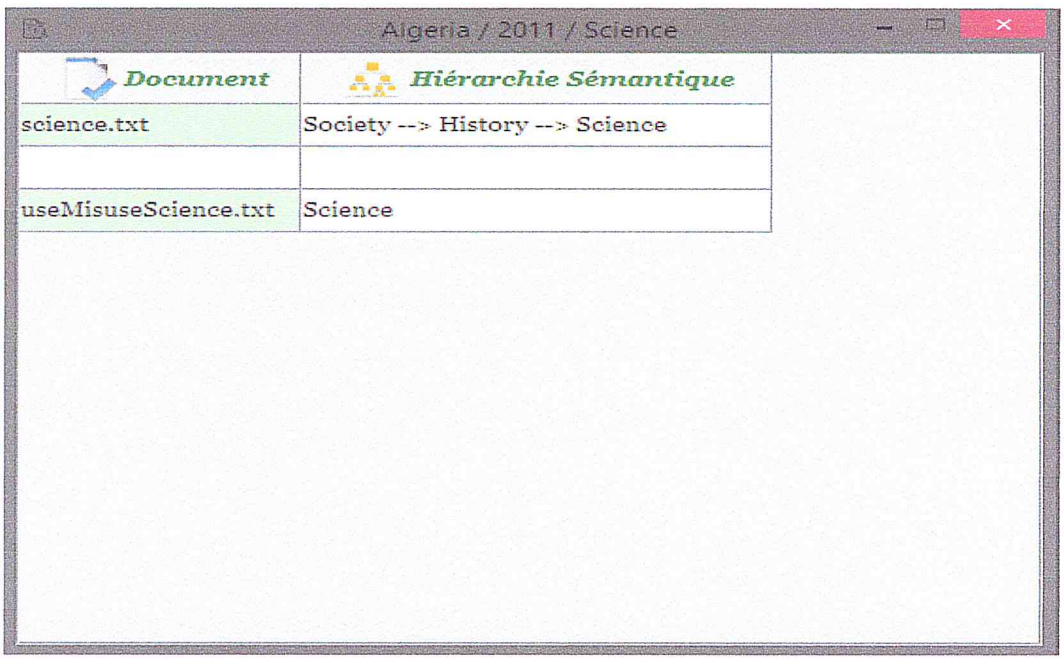

Fig. 4.7 : Résultats de recherche pour la requête entrée dans la figure : Fig. 4.6

Cette fenêtre (Fig. 4.7) affiche les documents trouvés et la hiérarchie sémantique qui correspond au choix Concept introduit.

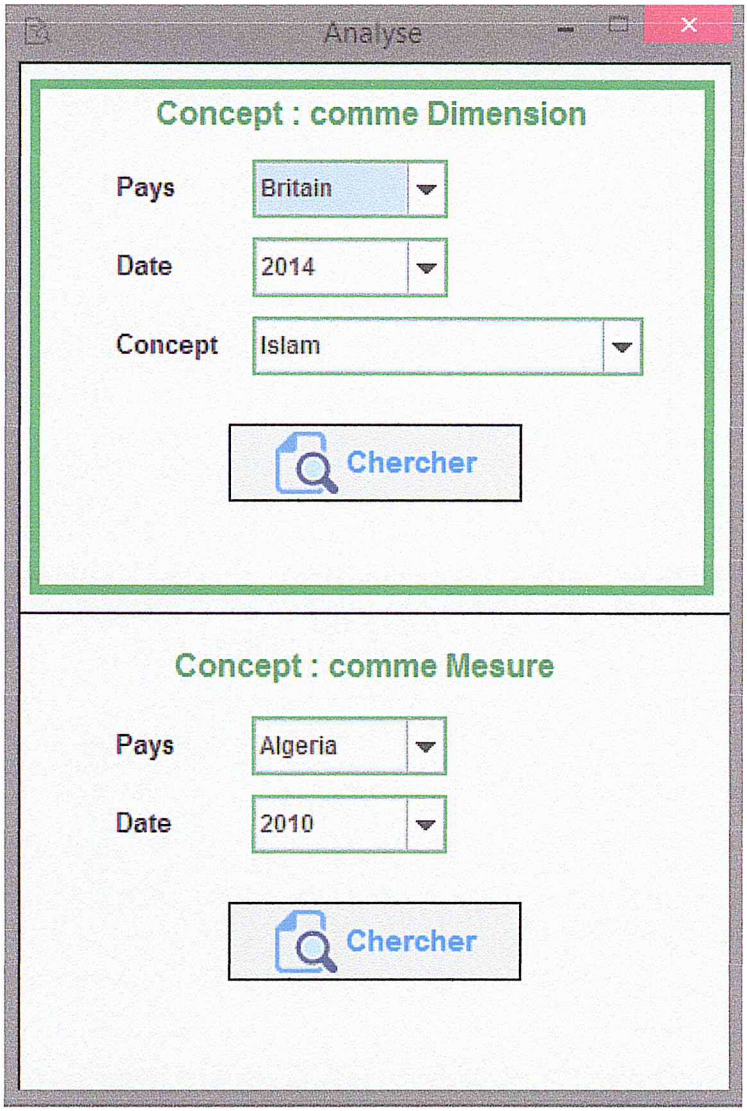

Fig. 4.8 : Choix de recherche pour Concept comme Dimension (Le résultat obtenu est montré dans la figure : Fig. 4.9)

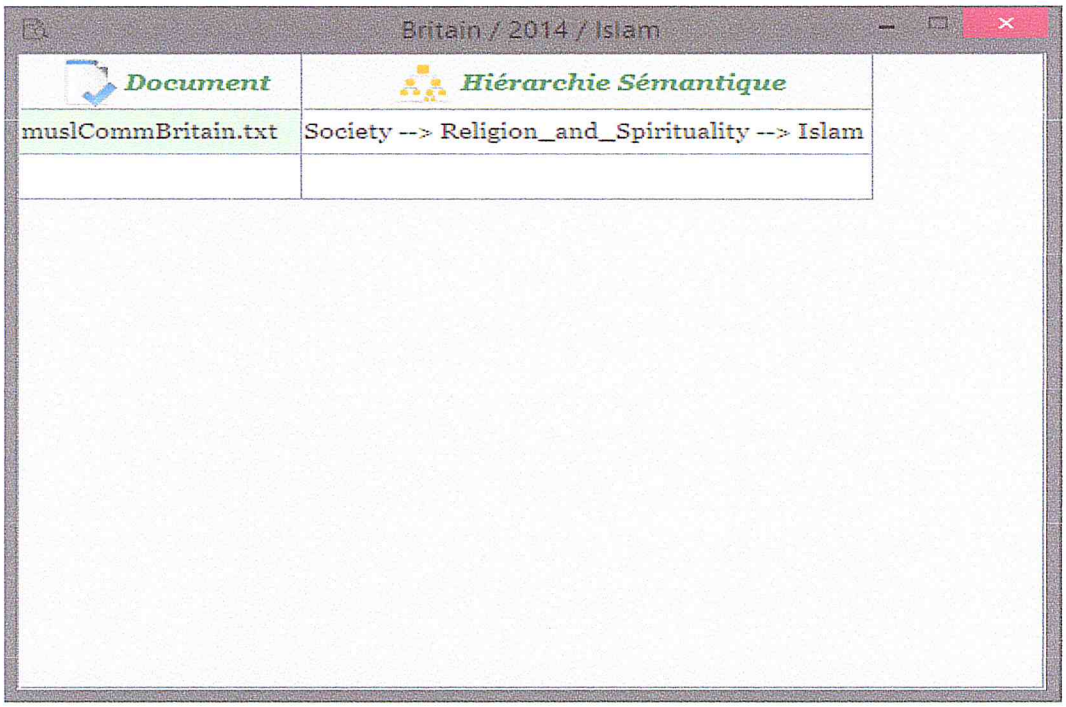

Fig. 4.9 : Résultats de recherche pour la requête entrée dans la figure : Fig. 4.8

En cliquent sur le nom du fichier dans cette fenêtre (Fig. 4.9), le fichier correspondant s'affiche automatiquement. Pour ce cas, voici le fichier « muslCommBritain.txt »

```
                    the muslim community of Britain

the most recent estimates suggest that Britain Muslim population is more than
million the largest number originate from Pakistan and Bangladesh while sizeable
groups have come from India Cyprus the Arab world Malaysia and parts of Africa
a growing community of British born Muslim mainly the children of immigrant
parent includes an increasing number of converts to Islam
  there are some mosques and numerous Muslim prayer centers throughout
Britain mosques are not only places of worship they also offer instruction in the
Muslim way of life and facilities for educational and welfare activities
  the first mosque in Britain was established at Working Surrey in
mosques now range from converted houses in many towns to the Central Mosque in
London and its associated Islamic Cultural Centre one of the most important Muslim
institution in the Western world the central mosque has the largest congregation in
Britain there are also important mosques and cultural centers in Liverpool
Manchester Leicester Bradford Edinburgh and Glasgow
  many of the mosques belong to various Muslim organization and both the Sunni
and shia traditions are presented together with some of the major traditions

                                        Britain An Official Handbook

                                        Britain 2014
```

Fig. 4.10 : muslCommBritain.txt

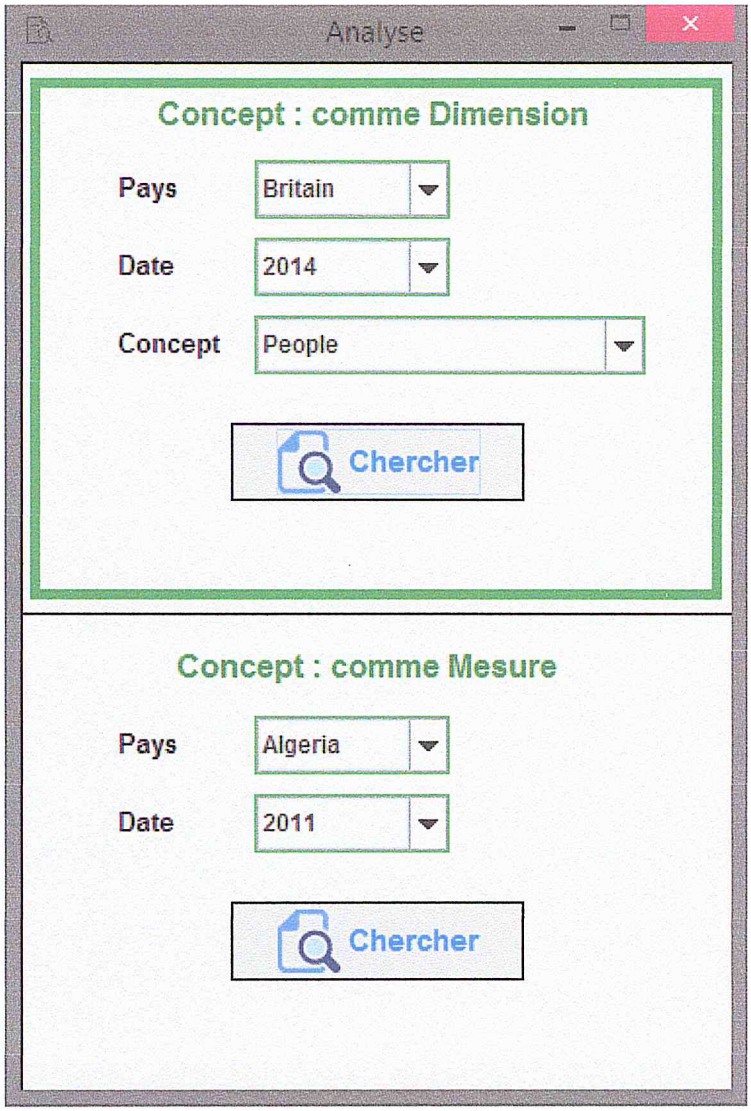

Fig. 4.11 : Choix de recherche pour Concept comme Dimension (Le résultat obtenu est montré dans la figure : Fig. 4.12)

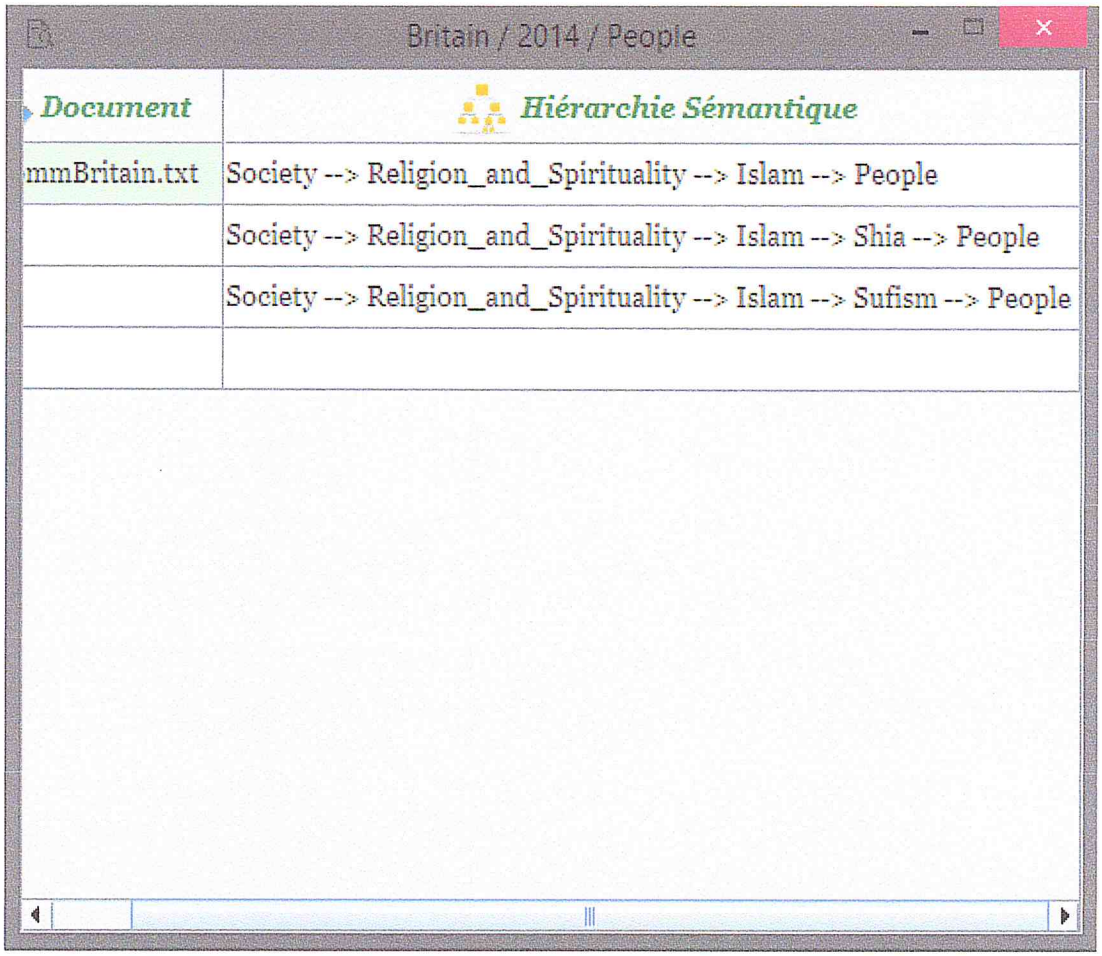

Fig. 4.12 : Résultats de recherche pour la requête entrée dans la figure : Fig. 4.11

On peut clairement voir qu'il peut exister des hiérarchies sémantiques différentes qui correspondent au choix Concept introduit (Fig. 4.12).

4.2.2) Concept comme Mesure :

Voici quelques tests :

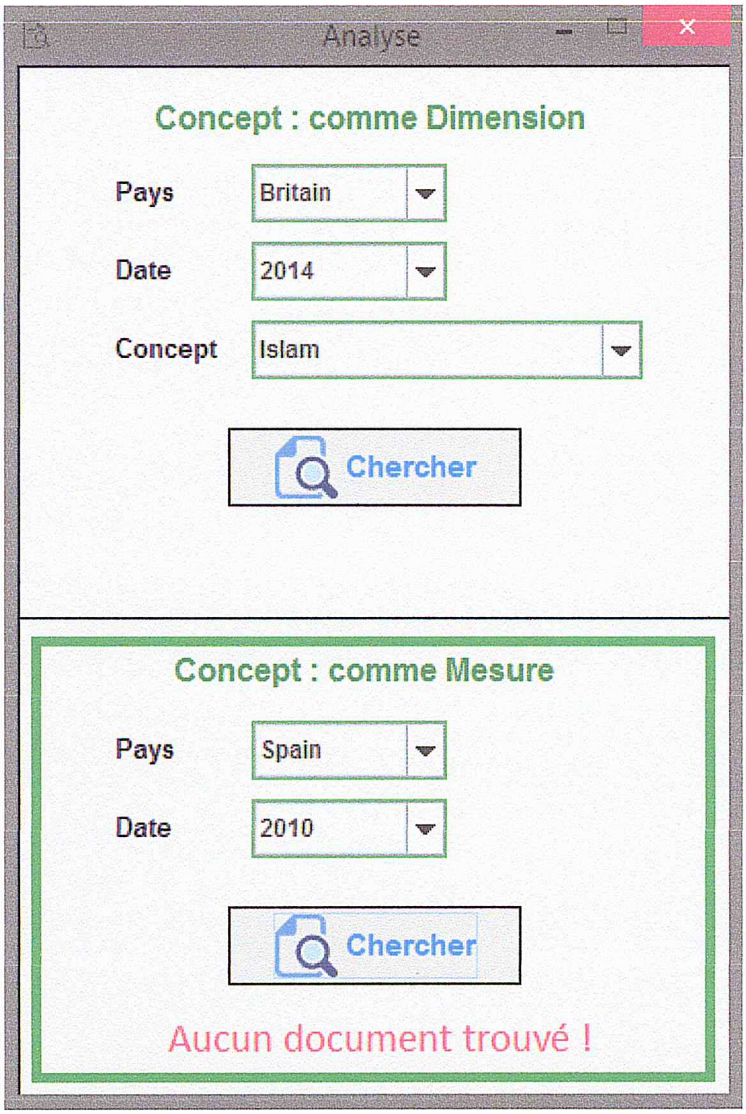

Fig. 4.13 : Aucun document trouvé en choisissant Concept comme Mesure

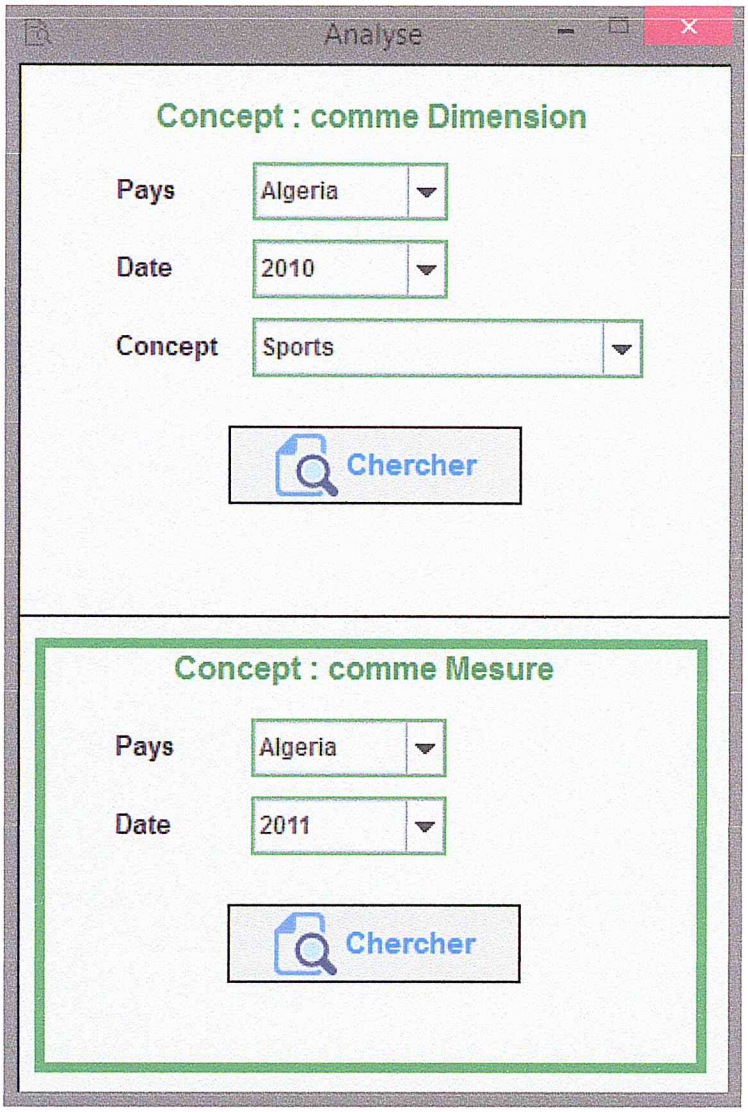

Fig. 4.14 : Choix de recherche pour Concept comme Mesure (Le résultat obtenu est montré dans la figure : Fig. 4.15)

Fig. 4.15 : Résultats de recherche pour la requête entrée dans la figure : Fig. 4.14

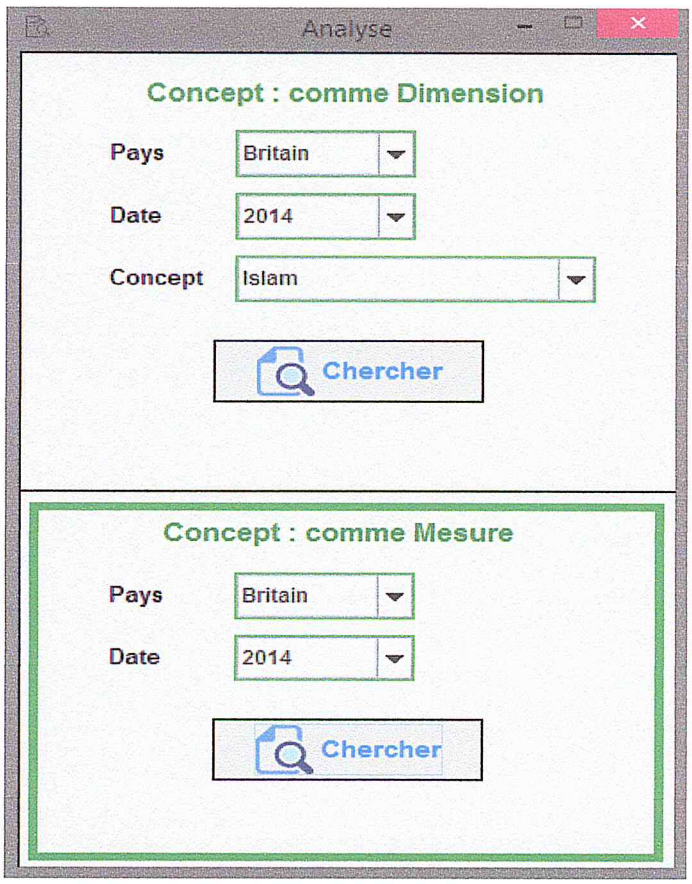

Fig. 4.16 : Choix de recherche pour Concept comme Mesure (Le résultat obtenu est montré dans la figure : Fig. 4.17)

Fig. 4.17 : Résultats de recherche pour la requête entrée dans la figure : Fig. 4.16

En cliquent sur le nom du fichier dans cette fenêtre (Fig. 4.17), la hiérarchie sémantique correspondante s'affiche automatiquement. Pour ce cas, voici la hiérarchie sémantique du fichier « muslCommBritain.txt »

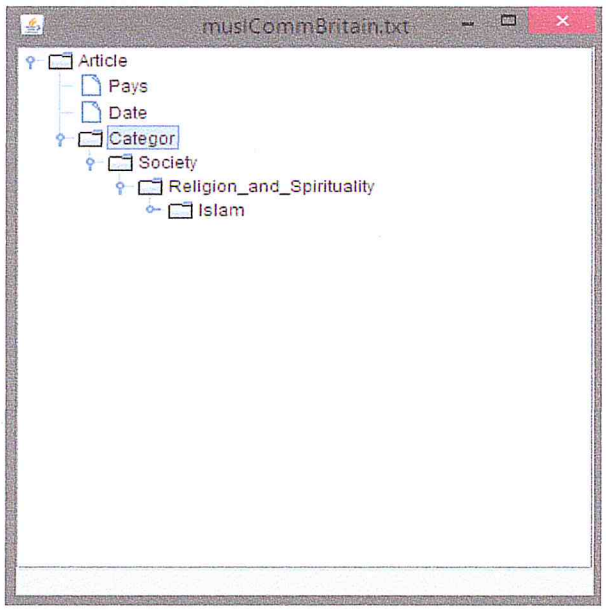

Fig. 4.18 : Hiérarchie sémantique correspondante au document : muslCommBritain.txt

Conclusion :

Enfin, on a pu constater la puissance du modèle MSMTO [chapitre 3], puisque il prend en compte la sémantique des données textuelles, et il est flexible en analyse.

D'une autre part, on a bien confirmé la puissance du modèle de sujets LDA [chapitre 2] pour l'extraction des sujets à partir d'un ensemble de documents, et la puissance de l'API de TextWise pour la catégorisation des documents.

Conclusion et Perspectives

1) Conclusion :

Dans notre projet, on a implémenté un programme Informatique pour l'analyse de données textuelles. La nécessité de créer un tel programme est à cause du volume important des données textuelles, et la quantité d'information qu'elles puissent contenir.

La réalisation de ce projet nous a montré la puissance du modèle MSMTO [chapitre 3] par rapport aux autres modèles proposés [chapitre 3] pour la représentation des données textuelles dont le but est de les analyser par la suite. Sa puissance réside surtout dans le fait qu'il prend en compte la sémantique des données textuelles, et qu'il est flexible en analyse, il considère le contenu sémantique des données comme une mesure, un fait ou même une dimension.

Notre application est adaptée aux fichiers qui contiennent seulement des textes simples (pas de photo, pas de mise en page,...), parce que notre objectif n'est pas l'extraction du texte brut à partir des fichiers, mais, d'appliquer quelques approches aux données textuelles. Parmi ces approches, la fusion entre une méthode statistique d'extraction de sujets à partir d'un document, comme LDA, et une méthode sémantique comme celle proposée par l'API de TextWise, pour obtenir de meilleur résultat concernant la catégorisation des textes. Une autre approche, c'est celle suivie par le modèle MSMTO [chapitre 3] qu'on a appliqué dans notre projet.

Enfin, on peut bien remarquer qu'on a simplifié notre projet, par exemple, notre application traite seulement des fichiers textes (.txt), et en considérant que le Pays et la Date d'un document donné se trouvent sur la dernière ligne,...
On a fait ça, parce qu'en réalité, notre projet c'est un ensemble de projets, comme le projet d'extraction des textes à partir d'un document complexe (qui contient des photos par exemple), et le projet d'extraction des métadonnées des fichiers.

2) Perspectives :

On propose comme perspectives, concernant la réalisation de notre projet, d'utiliser l'approche d'extraction des métadonnées à partir des fichiers, au lieu de supposer que certaines informations se trouvent à un emplacement spécifié, dans un fichier. Le but est de généraliser notre travail, et ne pas le limiter. On propose également l'utilisation d'un autre programme puisant, pour l'extraction des textes à partir des documents qui peuvent contenir des photos par exemple. Aussi, comme vous avez vu dans le chapitre Implémentation, on a simplifié l'analyse en introduisant une seule information pour chaque champ de recherche (par exemple un seul Pays, une seule Date,...). Alors, on propose encore comme perspective de généraliser le choix des informations concernant les champs de recherche pour l'analyse, en introduisant plus qu'une seule information pour chaque champ de recherche. En appliquant ça, on va générer des cubes complets, au lieu de cas particulier d'un cube comme est le cas de notre application.

Bibliographie :

1. D. M. Blei. Introduction to probabilistic topic models. In Communications of the ACM, 2012.

2. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, 2003.

3. L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In CVPR, 2005.

4. S. Attaf and N. Benblidia. Modélisation multidimensionnelle des données textuelles ou en sommes-nous ? In ASD Conference Proceedings, pages 3-25. Conference maghrebine sur les avancées des systèmes décisionnels, 2013.

5. M. Bautista, C. Molina, E. Tejeda3, and A. Vila. Using textual dimensions data warehousing processes. In International Conference, IPMU, Dortmund, Germany, pages 158-167. IPMU, 2010.

6. A. Blei, D.M. and Ng and M. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3(2) :993–1022, 2003.

7. D. Boukraa, O. Boussaid, F. Bentayeb, and D. Zegour. Modle multidimensionnel d'objets complexes : Du modèle d'objets aux cubes d'objets complexes. Ingénierie des Systèmes d'Information, 16, 2011.

8. R. Kimball. The data warehouse toolkit : Practical Techniques for Building Dimensional Data Warehouses. John Wiley and Sons, 1996.

9. C. X. Lin, B. Ding, J. Han, F. Zhu, and B. Zhao. Text cube : Computing ir measures for multidimensional text database analysis. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, pages 905-910. IEEE International Conference on Data Mining, 2008.

10. J. Mothe, B. Chrisment, C.and Dousset, and J. Alaux. Doccube : Multi-dimensional visualisation and exploration of large document sets. Journal of the American Society for Information Science and Technology, 54:650-659, 2003.

11. B.-K. Park, H. Han, and I.-Y. Song. Xml-olap : A multidimensional analysis framework for xml warehouses. In LNCS 3589, Springer, pages 32-42. 7th Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), 2005.

12. R. Tounier. Analyse en ligne (OLAP) de documents. Thèse de doctorat, Université Toulouse III. Paul Sabatier, 2007.

13. D. Zhang, C. Zhai, and J. Han. Topic cube : Topic modeling for olap on multidimensional text databases. In SDM '09 : Proceedings of the 2009 SIAM International Conference on Data Mining, Sparks, NV, USA', pages 1124-1135. SDM 09, 2009.

14. D. Zhang, C. Zhai, and J. Han. Mitexcube:microtextcluster cube for online analysis of text cells. pages 204-218. The NASA Conference on Intelligent Data Understanding (CIDU), 2011.

15. S.Attaf, N.Benblidia et O.Boussaid, The Multidimensional Semantic Model of Text Objects : A Framework for Text Data Analysis. In MEDI 2014 proceding, Lecture note in computer science LNCS springer.

W1. Wikipédia, http://fr.wikipedia.org/

W2. Site officiel de TextWise : http://www.textwise.com/

W3. Site officiel de DMOZ : http://www.dmoz.org/

W4. Introduction au Text Mining, http://www.christian-faure.net/2007/05/30/introduction-au-text-mining/