UNIVERSITÉ BLIDA 1

Faculté des Sciences
Département d'Informatique

T H È S E DE DOCTORAT
TROISIÈME CYCLE

Spécialité : Informatique décisionnelle

LES ENTREPÔTS DE TEXTES (TEXT WAREHOUSING) :
STRUCTURATION, MODÉLISATION ET ALIMENTATION DES TWH

Par

Sarah ATTAF

devant le jury composé de :

Oukid Khouas S.	U. Blida 1	Présidente
Boustia N.	U. Blida 1	Examinatrice
Boukhalfa K.	USTHB	Examinateur
Pr Benblidia N.	U. Blida 1	Promotrice
Boussaid O.	U.Lumière Lyon 2	Co-Promoteur

Blida, Juillet 2018

RESUME

L'analyse des données textuelles connaît un intérêt grandissant depuis plusieurs années. Le développement des moyens de communications a engendré une utilisation croissante du texte comme support de l'information. Les données textuelles numérisées constituent ainsi jusqu'à 80% des flux d'informations stockées quotidiennement dans les entreprises. Dés lors, un besoin d'outils capables d'analyser en profondeur cette masse de données s'impose. Les systèmes décisionnels classiques ont déjà fait leurs preuves dans le domaine de l'analyse des données simples. Or ces systèmes ne sont pas adaptés à l'analyse des données textuelles.

Dans le cadre de notre thèse, nous proposons une solution pour l'entreposage des données textuelles. Notre démarche couvre les principales phases d'un processus classique d'entreposage de données, et utilise de nouvelles méthodes adaptées aux données textuelles.

Pour offrir une solution aux problèmes d'alimentation des entrepôts de textes, nous introduisons une nouvelle technique d'intégration des données textuelles sous la forme d'un processus adapté d'ETL (Extract-Transform-Load) basé sur la technique de fouille de données LDA (Latent Dirichlet Allocation) et la taxonomie ODP (Open Directory Projet).

D'autre part, nous proposons un nouveau modèle baptisé : le Modèle Multi-dimensionnel Sémantique d'Objet Texte (MSMTO) comme une solution de structuration de données textuelles en vue de leur analyse. Afin de pouvoir spécifier des analyses sur les données issues des entrepôts de textes, nous introduisons des opérations permettant la manipulation des concepts du modèle MSMTO. Ainsi nous proposons un opérateur de construction de cube textuel sémantique (ST-Cube). Notre opérateur permet de donner aux décideurs la possibilité de définir le fait lors de l'analyse, ce qui offre plus de flexibilité.

Pour répondre au problème d'agrégation de données textuelles, nous proposons un opérateur d'agrégation : le $Top\_KRankedTopics$ qui permet d'obtenir une vision synthétique des informations. Il agrège un ensemble de n sujets dans un sous-ensemble de k sujets les plus pertinents, puis génère pour chaque sujet une liste de documents classés.

Pour valider nos différentes contributions, nous avons réalisé, en plus des

travaux d'implémentation, une étude expérimentale pour chacune de nos propositions. Les résultats retournés montrent l'efficacité de notre solution pour l'entreposage et l'analyse en ligne des données textuelles.

Mots clés : Entrepôt de textes, ETL, Modélisation multidimensionnelle, MSMTO, ST-Cube, Opérateur OLAP, Top_KRandedTopics, ODP, LDA, fléxibilité d'analyse.

ABSTRACT

The analysis of textual data has known an ever-increasing interest throughout the research community in the last several years. The development and proliferation of communication means has led to a growing usage of the text as a support of information. Textual data thus constitute up to 80% of information stored daily in companies. Therefore, tools for in-depth analysis for such mass of data are much needed. Even though, decision support systems have already given good results in the field of simple data analysis, these systems are not quite suitable for textual data analysis.

As part of our thesis, we propose a solution for textual data multidimensional analysis. Our work covers the main phases of a data warehousing process and uses new methods adapted to textual data. While addressing the problems related to text warehouses feeding, we introduce a new textual data integration technique under the form of an ETL (Extract-Transform Load) process. The proposed approach is based on the Latent Dirichlet Allocation (LDA) technique and the ODP (Open Directory Project) Taxonomy. On the other hand, we propose a new model called the Multidimensional Semantic Model of Text Objects (MSMTO) as a textual data structuring solution for textual data analysis. In order to be able to specify analyzes on the text warehouses data, we introduce operations allowing the manipulation of the MSMTO concepts. Thus, we propose a semantic text cube construction operator (ST-Cube). Our operator gives decision-makers the possibility to define the fact during the analysis phases, which offers more flexibility of the analysis process.

To address the problem of textual data aggregation, we propose an aggregation operator : the Top_KRankedTopics, which aggregates a set of n topics into a subset of k most relevant topics and then generates for each topics a classified list of textual documents.

To validate our different contributions, we realized, in addition to the implementation task, an experimental study for each of our proposals. The results returned show the interest of our solution for the storage and online analysis of textual data.

Keywords : Text warehouse, ETL, multidimensionnal modelling, MSMTO, ST-Cube, OLAP operator, Top_KRandedTopics, ODP, LDA.

REMERCIEMENTS

L'achèvement de ce travail mené sur plusieurs années procure une grande satisfaction. Il est l'occasion de se remémorer les différentes embûches qu'il a fallu surmonter mais surtout les personnes qui m'ont permis d'en arriver là. J'apprécie sincèrement l'inspiration, le soutien et les conseils de tous ceux qui ont collaboré pour faire réussir cette thèse.

J'adresse mes remerciements les plus sincères au professeur Nadjia Benblidia, pour avoir dirigée cette thèse, pour sa rigueur scientifique, pour ses critiques instructives ainsi que pour ses précieux conseils. Je la remercie aussi pour la confiance qu'elle m'a accordée, pour son soutien et ses encouragements. J'ai admiré ses qualités scientifiques, pédagogiques, mais surtout humaines. Merci d'avoir guidé mes premiers pas dans le chemin de la recherche.

Je tiens à exprimer ma plus profonde gratitude à mon co-directeur de thèse, le professeur Omar Boussaid pour son soutien indéfectible, sa disponibilité, ses encouragements tout au long de cette thèse. Il s'y est grandement impliqué par ses directives, ses remarques et suggestions. Je tiens à le remercier aussi pour la liberté qu'il m'a laissé dans la conduite de ma recherche. Je le remercie d'autant plus pour ses qualités d'écoute. Qu'il soit assuré de ma profonde reconnaissance et de mon trés grand respect.

Mes remerciements les plus sincères vont au Dr. Saliha Oukid de m'avoir fait l'honneur d'accepter de présider le jury de ma soutenance, ainsi qu'aux examinateurs, Dr. Narhimene Boustia et Dr. Kamel Boukhalfa, d'avoir accepté d'évaluer mon travail. Je les remercie pour l'honneur qu'ils me font en participant à mon jury.

Je remercie les membres du laboratoire LRDSI, doctorants et enseignants chercheurs en particulier le Dr. Mahfoud Bala pour son soutien et ses bon conseils, mes amis : Yasmine, Lilia et Lamia. Je tiens également à remercier les membres du laboratoire ERIC de l'université Lyon 2 pour leur bon accueil et leur convivialité, en particulier : Dr. Fadila Bentayeb, Rachid Aknouche, Fatima Azzi et Fatima Younsi.

Je remercie mon père et ma mère d'avoir toujours cru en moi. Mes parents ont toujours apprécié l'éducation et ont inculqué ses valeurs en moi. Sans leur bénédiction et leur soutien constant tout au cours de ma vie, je n'aurais pas été la personne que je suis aujourd'hui. Je me sens tellement béni d'avoir des parents si dévoués qui ont toujours exprimé à quel point ils sont fiers de moi. Je leur serai à jamais reconnaissante, sans eux, je n'aurais rien accomplie !

Je remercie ma famille et mes amis qui m'ont toujours soutenu. J'exprime particulièrement ma gratitude envers mes frères : Ouahid, Walid et Mohamed, ainsi qu'á ma très chère sœur Nadia qui m'a toujours soutenue. Je ne manquerai pas de dire un grand merci à mes beaux parents, mes beaux frères : Moussa et Younes ainsi que mes belles sœurs : Amina, Hiba, Marwa et Soumiya. Qu'ils trouvent avec ceci un modeste geste de reconnaissance et de remerciement.

Je serais éternellement reconnaissante envers mon époux bien-aimé et dévoué Ayoub. Ses encouragements, ses sacrifices et surtout sa patience à travers mes études de doctorat m'ont toujours guidés dans ce voyage rude mais agréable. Sans aucun doute, ce travail n'aurait pas été achevé sans sa présence dans ma vie.

Enfin, je voudrais dédier ce travail à mon précieux fils Jad Adam, la lumière et la joie de ma vie. Son regard et ses rires m'ont donné la force et la motivation pour finir ce travail.

TABLE DES MATIERES

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

INTRODUCTION

1. Contexte général :

   Le processus décisionnel ou les systèmes d'information décisionnels (SID) sont nés d'un besoin exprimé par les entreprises, besoin non satisfait par les systèmes de bases de données traditionnels. Les SID sont basés sur l'approche des entrepôts de données pour exploiter d'importants volumes d'information pour des fins d'analyse et d'aide à la décision. Un entrepôt permet de stocker les données nécessaires à la prise de décision ; il est alimenté par des extractions de données portant sur des bases opérationnelles, appelées sources de données.

   Les technologies d'entreposage de données et d'analyse en ligne ont maintenant largement fait leurs preuves dans les domaines de l'analyse des données structurées. Tandis qu'à l'heure actuelle, avec l'accroissement continu du volume de l'information dans les entreprises, les données à analyser ne sont plus seulement numériques ou symboliques, mais sous différents formats (textes, images, son, vidéos, bases de données, etc.), provenant de sources différentes. De cela est né un nouveau besoin, celui de l'analyse de données dites « complexes ».

   Nos travaux de recherche se placent dans le contexte des systèmes d'aide à la décision basés sur l'approche des entrepôts de données. Nos travaux visent plus particulièrement à développer des systèmes décisionnels aptes à supporter efficacement les analyses sur les données textuelles.

2. Problématique :

   L'entreposage des données est constitué de deux processus : la modélisation des données et l'intégration des données. L'organisation et la structuration des données textuelles constituent une problématique déjà connue dans la littérature. La question telle qu'elle se pose dans le cadre de cette thèse est bien différente. Puisse qu'on cherche à modéliser les données textes dans un but d'analyse. Celles-ci ont la particularité de pouvoir être considérées selon des niveaux de granularité différents. La modélisation 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. D'autre part, un document texte peut être perçu comme un ensemble de mots ayant déjà une certaine structure. L'objectif devient alors de concevoir une base de textes orientée analyse.
Comment faut-il alors capter la sémantique des données textes et l'incorporer dans un modèle pour qu'elle puisse être prise en charge lors de la phase d'analyse ? La prise en compte de celle-ci dans la modélisation des bases de textes est un impératif. Dans la littérature, des méthodes de modélisation conceptuelle des entrepôts de données sont proposées et offrent des propriétés permettant de représenter des situations complexes. De telles méthodes sont elles suffisantes pour répondre aux besoins d'une modélisation d'entrepôts de textes orientés analyse ?
L'intégration des données textuelles dans le cadre des système décisionnel représente un vrai challenge. Les outils ETL utilisés pour l'alimentation des entrepôts de données offrent des résultats concluants lorsque les données traitées sont de type structuré, ou semi structuré. À l'opposé, les limites de ces techniques sont évidentes lorsqu'il s'agit d'intégrer les données textuelles dans un système décisionnel.
D'autres part, les fonctions d'agrégation classiques tel que (Sum, "Max,Min...etc) s'adapte bien aux données numériques. Tandis que pour les données textuelle, ces fonctions ne permettent pas de les agréger, ce qui met à l'évidence la nécessité de développer de nouveaux opérateurs spécifiques aux données textuelles.

3. Contributions de nos travaux de recherche :
Pour répondre à la problématique évoquée ci-dessus, nous proposons une nouvelle approche d'entreposage de données textuelles, qui comprend trois composants : (i) une approche d'intégration de données textuelles sous la forme d'un processus ETL, (ii) un modèle multidimensionnel d'entrepôt de textes et (iii) le module texte OLAP. (figure 3).

   (a) Une approche ETL pour l'alimentation des entrepôts de textes :
   Les approches ETL nécessitent des structures rigides pour fonctionner. Cette exigence n'est pas remplie dans de nombreux cas où les documents contiennent des textes non structurés. Ainsi, les infor-

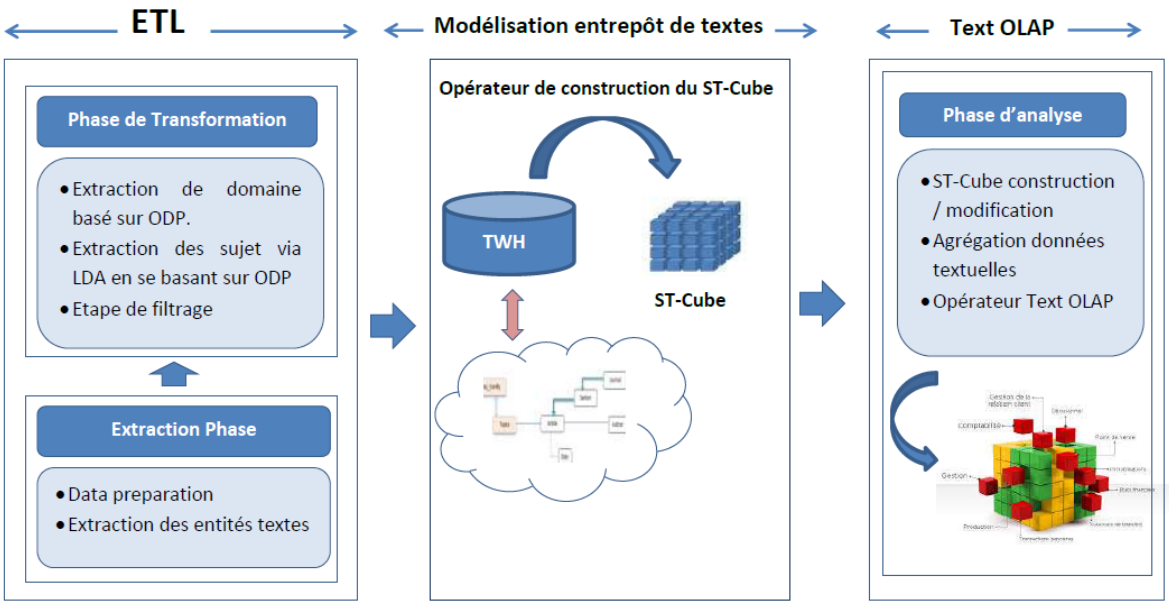

FIGURE 1 – Architecture générale

mations définies par ces documents textuels ne sont guère analysées par les approches conventionnelles en raison des problèmes d'exploration de textes. Afin d'assurer l'intégration des données textuelles dans un processus décisionnel, nous proposons une nouvelle approche ETL dédiée à l'alimentation des cubes de textes sémantique [5]. L'approche proposée consiste à extraire le contenu sémantique des données textuelles, à transformer les données extraites pour qu'elles correspondent au schéma de l'entrepôt de textes et à charger ces données dans l'entrepôt de textes. Pour permettre l'alimentation sémantique de notre ST-Cube, notre approche ETL utilise la méthode LDA (Latent Dirichelet Allocation) avec la taxonomie ODP pour fournir des hiérarchies sémantiques pondérées représentant les documents textuels.

(b) Le Modèle Multidimensionel Sémantique d'objet texte :
La modélisation multidimensionnelle consiste à organiser des données afin que les applications OLAP soient performantes et efficaces. Les modèles d'entrepôt existants offrent un cadre pour une modélisation multidimensionnelle de données simples, mais ils ne

conviennent pas aux données textuelles [6].
Pour résoudre ce problème, nous proposons dans [7] le modèle multidimensionnel sémantique d'objets texte (MSMTO). Le modèle proposé offre une description formelle a ce type de données non structurées pour permettre une utilisation efficace de l'information contenue dans le texte. Notre modèle est basé sur le paradigme objet. Il intègre un nouveau concept, l'objet de contenu sémantique utilisé pour représenter et organiser la sémantique des données textuelles dans un format hiérarchique, afin de permettre une analyse sémantique sur différents niveaux de granularité. Notre approche de modélisation considère la composition interne des documents textuels comme une hiérarchie structurelle, ce qui permet à l'utilisateur d'effectuer des analyses sur différents niveaux hiérarchiques. Il offre également une flexibilité d'analyse en considérant le contenu sémantique des données textuelles comme une mesure, un fait ou même une dimension.

(c) Opérateur d'agrégation Top_KRankedTopics :
L'agrégation des données est l'une des tâches d'analyse les plus importantes. La portée des fonctions agrégation se réduit à un ensemble restreint d'opérations, tel que le comptage, lorsque les mesures d'analyse ne sont plus numériques.
Pour assurer une analyse multidimensionnelle des documents textuels, nous proposons un nouvel opérateur d'agrégation : le $Top\_KRankedTopics$ [5]. Ce dernier représente un outil d'analyse non conventionnel qui prétend une extension de l'analyse en ligne des données complexes. L'opérateur proposé est particulièrement dédié à traiter la nature complexe des données textuelles. Il dépasse les indicateurs numériques pour permettre au puissant cadre OLAP de fonctionner sur ce type de données complexe.
Notre opérateur d'agrégation $Top\_KRankedTopics$ agrège un ensemble de document par : (1) sélectionner les K sujets les plus pertinents (ayant le plus grand poids), (2) pondérer les documents selon les hiérarchies sémantiques qui les représente(obtenue en appliquant l'approche décrit dans le chapitre 3). Notre opérateur

d'agrégation sélectionne les k premiers sujets et retourne pour chaque sujet une liste de documents pondérés.

(d) Expérimentations :

Pour la validation de nos propositions, nous avons développé et mis en place une plate-forme pour le stockage et l'analyse des données textuelles, regroupant nos différentes contributions théoriques. Cette plateforme permet la construction d'entrepôts de textes et leur alimentation ainsi que leur analyse. Elle servira à évaluer et à valider l'efficacité de nos propositions par le biais d'un travail d'expérimentation, tout en assurant une validation fonctionnelle de notre approche.

L'évaluation de nos propositions sera achevé dans différentes phases, ça consiste à évaluer :

- La possibilité de construire un entrepôt de texte en utilisant le modèle du $MSMTO$.
- La possibilité de construire un cube de texte en utilisant le modèle du $ST - Cube$.
- La qualité de l'information qui va alimenter notre entrepôt de textes.
- L'analyse des données textuelles en utilisant notre opérateur $Top\_KRankedTopics$.

L'étude expérimentale montre que l'approche ETL améliore considérablement la qualité de l'information extraite du document textuel par rapport à deux autres travaux, en termes de : $prcision$, $rappel$ et combinaison de ces deux paramètres ($F1 - Score$). De plus, le nouvel opérateur $Top\_KRankedTopic$ permet d'effectuer des analyses en ligne efficaces sur les données textuelles tout en tenant compte des différents facteurs sémantiques. En outre, nous avons procédé à une évaluation sur les articles de presse, ce qui montre que, la prise en compte de la pertinence contextuelle, améliore considérablement la précision des résultats obtenus par notre opérateur $Top\_KRankedTopic$.

4. Organisation de la thèse : ce mémoire est organisé en deux grande partie :

   - La première partie est consacrée à l'état de l'art. Elle comporte les chapitres 2, 3 et 4. Le chapitre 2 expose les concepts de base des systèmes décisionnels sur lesquels reposent nos travaux. Le chapitre 3 décrit les tâches et les techniques les plus fondamentales pour la fouille de textes. Le chapitre 4 présente une étude approfondi sur les travaux traitant l'analyse multidimensionnelle de données textuelles. L'étude de ces travaux nous a permis de les classer, selon le type de concepts utilisés, en deux familles de modèles : (i) travaux avec des modèles extensifs basés sur les concepts de base des modèles d'entrepôts classiques et (ii) travaux avec des modèles à nouveaux concepts proposant une nouvelle manière de percevoir la complexité des données textuelles. Ce survol de modèles multidimensionnels dédiés à l'analyse textuelle nous a permis de constater la diversité des modèles proposés et de décerner les problèmes majors liés à l'analyse des données textuelles.

   - La deuxième partie est consacrée à nos contributions. Elle regroupe les chapitres 5 et 6. Le chapitre 5 présente la première partie de nos contribution, à savoir, une approche ETL pour l'alimentation des entrepôts de textes. le chapitre 6 est consacré à la deuxième famille de contributions, il introduit dans un premier temps le modèle multidimensionnel sémantique d'objets textes ainsi que ces concepts de base. Par la suite, nous présentons notre modèle de cube de textes $ST - Cube$ ainsi que l'opérateur de construction du ST-Cube. Finalement, nous présentons notre opérateur d'agrégation le $Top\_KRankedTopics$. Enfin, le chapitre 7 conclut cette thèse en présentant un bilan sur l'ensemble des contributions ainsi que nos perspectives de recherche.

Partie I

Etat de l'ART

Chapitre 1
Techniques d'exploration des données textuelles

1.1 Introduction

Les grandes masses de données textuelles générées au quotidien augmentent considérablement. Cet énorme volume de texte principalement non structuré est facilement traité et perçu par les humains, mais beaucoup plus difficile à comprendre pour les machines, ce qui les rend encore plus difficile à exploiter pour des tâches d'analyse. En conséquence, il y a un grand besoin de développer des méthodes et des algorithmes afin de traiter efficacement ce type complexe de données.

La fouille de textes regroupe des méthodes, techniques et outils employés pour traiter automatiquement de grandes quantités de données textuelles non structurées (corpus de documents) comme un ensemble d'emails, de tweets, d'articles de presse, de réponses ouvertes à un questionnaire, et bien d'autres encore. L'objectif étant de dégager et de structurer le contenu du corpus, les principales thématiques, dans une perspective d'analyse rapide et de découverte d'informations cachées ou de prise automatique de décision.

Dans la littérature, des méthodes de fouille de textes ont donné de bons résultats pour l'exploration des données textuelles. Dans notre travail de thèse, nous proposons un couplage entre les techniques de fouille de textes et les techniques d'analyse multidimensionnelle dans le cadre de la construction d'un entrepôt de textes.

Dans ce chapitre, nous décrivons les tâches et les techniques les plus fondamentales pour la fouille de textes. La suite de notre chapitre est organisée comme suit : la section 2 illustre les approches de fouilles de textes et ses interactions avec d'autres domaines. La section 3 présente quelques techniques élémentaire de fouille de textes tel que la classification supervisée et le clustering.

1.2 Les approches de fouille de textes

La fouille de textes a été introduit pour la première fois par Fleman [8]. Elle est désignée sous le terme anglo-saxon *text mining*, qui représente l'ensemble de processus permettant, à partir d'un ensemble de ressources textuelles (structurées [9] ou semi-structurées ([10], [11], [12]), de construire des connaissances pouvant être représentées dans un langage de représentation formel et exploitées pour raisonner sur le contenu des textes.

La fouille de textes est un domaine multidisciplinaire qui couvre un large éventail d'algorithmes connexes pour analyser le texte, couvrant diverses communautés telle que : la recherche d'information, le traitement automatique du langage naturel, et l'apprentissage automatique. La figure 1.1 illustre le diagramme Venn sur la fouille de textes et ses interaction avec d'autres domaines.

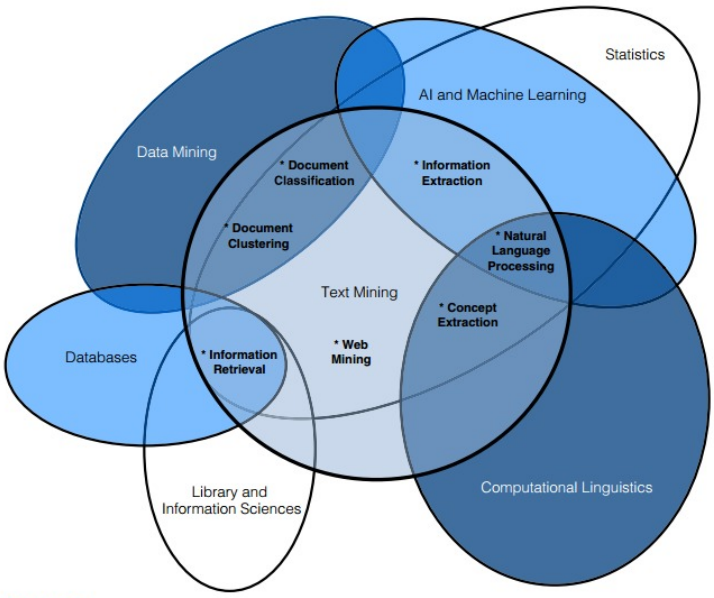

FIGURE 1.1 – Diagramme de Venn de l'interaction de la fouille de textes avec d'autres domaines [1]

1.2.1 La recherche d'information

La recherche d'information est la tâche de trouver des ressources d'information(documents) à partir d'une collection de données non structurés, satisfaisant le besoin d'information d'un utilisateur [13]. Par conséquent, la recherche d'information s'est principalement focalisée sur la facilité d'accès à l'information plutôt que sur l'analyse de l'information, ce qui est le but principal de la fouille de textes. La recherche d'information a moins de priorité sur le traitement ou la transformation de textes alors que la fouille de textes permet d'aller au-delà de l'accès à l'information pour aider les utilisateurs à analyser et à comprendre l'information pour faciliter la prise de décision.

1.2.2 Traitement automatique du langage naturel

Le traitement automatique du langage naturel(TALN) est est une discipline à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle.
On regroupe sous le vocable TALN l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide des machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication [14]. De nombreux algorithmes de fouille de textes font appel aux techniques TALN, tel que : l'étiquetage morpho-syntaxique, l'analyse syntaxique et d'autres types d'analyse linguistique ([15] [16]).

1.2.3 L'extraction d'information

L'extraction d'information (en anglais, information extraction) s'inscrit comme l'une des activités qui consistent à rechercher, découvrir et traiter des connaissances à partir d'un ensemble de textes. L'ensemble de ces activités constitue le domaine dit de fouille de textes.
L'extraction d'information désigne une technologie récente qui vise à extraire et à structurer automatiquement un ensemble d'information précises apparaissant dans un ou plusieurs documents textuels non structurés ou semi-structurés ([17], [18]) écrits en langue naturelle. Elle sert généralement comme point de départ pour différents algorithmes de fouille de textes.

1.2.4 Résumé automatique

De nombreuses applications de fouille de textes ont besoin de résumer les documents textuels afin d'obtenir un aperçu condensé d'un document volumineux ou d'une collection de documents sur un sujet ([19], [20]).
Il existe deux types de résumés : les résumés extractifs et les résumés abstractifs. Le résumé extractif est aujourd'hui l'approche la plus répandue, elle consiste à pondérer les phrases selon leur représentativité. Le résumé abstractif consiste à extraire des informations ou des notions du document à résumer afin de rédiger un tout nouveau texte.

1.2.5 Méthodes d'apprentissage non supervisées

Les méthodes d'apprentissage non supervisées sont des techniques qui visent à trouver les structures cachées des données non étiqueté. Ces techniques n'ont pas besoin de phase d'apprentissage. Le clustering et les modèles de sujets (topic models) sont les deux algorithmes d'apprentissage non supervisés couramment utilisés dans le contexte des données textuelles.

1.2.6 Méthodes d'apprentissage supervisées

Les méthodes d'apprentissage supervisé sont des techniques d'apprentissage automatique qui permettent d'inférer une fonction ou d'apprendre un classificateur à partir des données d'apprentissage afin d'effectuer des prédictions sur les nouvelles données. Il existe un large éventail de méthodes supervisées comme l'algorithme du plus proches voisins, les arbres de décision, les classificateurs basés sur des règles et les classificateurs probabilistes ([21], [22]).

1.3 Classification automatique des documents textuels

La fouille de textes englobe des traitements linguistiques (morphologiques, syntaxiques, sémantiques) et de multiples techniques d'analyse de données. Parmi toutes ces techniques existants ([23], [24], [25]), nous nous intéressons dans cette section aux approches de classification automatique de documents textuels.

La classification (catégorisation) de texte est plus que essentielle à l'ère d'Internet et des Big data. Elle a pour objectif de regrouper les textes similaires, c'est à dire thématiquement proches, au sein d'un même ensemble. L'intérêt d'une telle démarche est d'organiser les connaissances de façon à pouvoir effectuer, par la suite, une recherche ou une extraction d'information efficace.

1.3.1 Le pré-traitement

Avant de pouvoir créer et entrainer un quelconque classificateur à partir d'un corpus de documents donné, les documents textuels doivent être transformés en entrés valides compréhensibles par les différents algorithmes de classification ou de clustering. Ces entrées valides sont en fait des vecteurs ou des matrices de descripteurs (mot ou groupe de mots, n-gram) apparaissant dans chaque document texte. C'est cette transformation entre un document texte et un document vecteur qu'on appelle pré-traitement.
Les approches de classification classiques des documents textuels se composent du pré-traitement, l'extraction des descripteurs, la sélection des descripteurs et les étapes de classification. Bien qu'il soit confirmé que l'extraction des descripteurs [26], la sélection des descripteurs [27] et l'algorithme de classification [28], ont un impact significatif sur le processus de classification, l'étape du pré-traitement peut avoir une influence remarquable sur la réussite de ce processus. L'impact de cette étape dans le domaine de la classification des données textuelles a été étudié dans [29]. L'étape du pré-traitement se compose de trois tâches : Tokenisation, élimination des mots vides et la lemmatisation.

- **Tokenisation :** un token est une unité définie comme une séquence de caractères comprise entre deux séparateurs ; les séparateurs étant les blancs, les signes de ponctuation et certains autres caractères comme les guillemets ou les parenthèses. La tokenisation consiste à segmenter un document texte en tokens de mots, séquences de mots ou carrément de phrases, mais généralement elle s'opère souvent sur des mots [30].

- **Élimination des mots vides :** une fois les documents textes découpés en tokens, nous apercevons que certains de ces tokens sont présents dans tous les textes du corpus, c'est ce que nous appelons les mots vides : les prépositions, les déterminants, les adverbes... comme "la ,le , dans, car,"

dans la langue française, et "the, and, after" dans la langue anglaise. Ils représentent 30% des mots dans un texte. La présence de ces mots n'apporte absolument aucune différence tant sur le plan sémantique que sur le plan lexical. Cela veut dire que leur présence dans tous les textes du corpus les rend non discriminants et du coup leurs utilisation pour une tâche de classification s'avère inutile. Par contre, leur suppression réduit la dimension de notre document vecteur, par conséquent, le temps de traitement et le temps d'apprentissage seront réduit considérablement ([31], [32]).

- **Lemmatisation :** lors de certaines représentations vectorielles, chaque mot présent dans le corpus est considéré comme un descripteur. Cette façon de faire contient certaines irrégularités, notamment pour les verbes à l'infinitif et les verbes conjugués (développer, développe, développé, développèrent...), nous remarquons que ces derniers ont le même sens mais chacun est considéré comme un descripteur à part entière. La racinisation vient parer à ce problème, en considérant uniquement la racine de ces mots plutôt que les mots en entier sans se soucier de l'analyse grammaticale. La substitution des mots par leurs racines permet une meilleure représentation, surtout et réduit considérablement la dimension des descripteurs.

1.3.2 Représentation des documents

Dans la classification automatique de texte, il a été prouvé que le terme est la meilleure unité pour la représentation des documents textuel et pour la classification [33]. Bien qu'un document textuel peut contenir une quantité importante d'information, malheureusement, il lui manque la structure. Il est donc essentiel d'avoir un modèle de représentation de document efficace pour construire un système de classification efficace.

1.3.2.1 Le sac à mots

La façon la plus simple et la plus évidente pour la représentation d'un document texte par un document vecteur, est d'utiliser les mots comme descripteurs. Ainsi, nous construisons un sac à mots (bag of words) de tous les

mots qui apparaissent au moins une fois dans le corpus. Cette méthode, qui conserve le sens naturel des descripteurs, est loin de répondre à toutes les attentes de la classification de texte, à cause, notamment, de certaines anomalies liées à la variation des fréquences par rapport à la longueur du document, au problème des mots composés, et principalement, au fait que l'ordre d'apparition des mots dans les phrases du document n'est pas pris en considération. Les mots sont regroupés en vrac et traités d'une manière indépendante, ce qui nuit considérablement à la sémantique du texte.

1.3.2.2 Les N-grams

Cette méthode de représentation d'un document texte consiste à partager ce dernier en séquences de n caractères. Le choix de n est dans la plus part du temps supérieur à 1.

1.3.2.3 Représentations vectorielles

Les modèles vectoriels sont largement utilisés pour la représentation de textes. Le modèle vectoriel standard et le modèle Latent Semantic Indexing sont les plus utilisés et implémentés. Nous détaillons ces derniers dans la suite de cette section.

- **Le modèle vectoriel standard :** dans le cadre du modèle vectoriel standard, les textes sont considérés comme des « sacs de mots » ([34], [35], [36]). L'idée principale est de transformer les différents documents d'une base documentaire en vecteurs où chacun des éléments d'un vecteur de texte représente des unités textuelles ou tout simplement des mots. Un mot est considéré comme étant une suite de caractères encadrés par des caractères de ponctuation ou appartenant à un dictionnaire spécifique.
  le modèle vectoriel standard a été initialement introduit par Gérard Salton ([37], [34]) dans l'objectif d'implémenter un système de recherche d'information. Dans ce modèle les composants des vecteurs représentent des termes considérés comme les plus discriminants. Dans le cadre du modèle vectoriel standard, ces termes sont sélectionnés en fonction de leurs fréquences d'apparition dans les documents et en fonction du

nombre de documents contenant ces termes.
Le poids donné à un terme dans un document est calculé en fonction de la fréquence d'occurrence du terme TF (Frequency) dans le document et du nombre de documents contenant le terme IDF (*Inverse Document Frequency*). Les calculs des poids et les pondérations accordés à un document ont fait l'objet de nombreuses études ([38], [39], [40], [41]).

- **Fréquence des termes (TF) :** on désigne par TFla fréquence d'un terme (descripteur) dans un texte donné. C'est un calcul de fréquence très simple, mais qui s'avère efficace et pratique. Nous l'utilisons souvent en association avec d'autres fréquences. Nous dénombrons plusieurs manières de calcul de la TF :

  1. **TF absolue :** c'est le nombre de fois qu'un terme apparait dans un texte donné.

     $$TF = NT \tag{1.1}$$

     ou NT est le nombre de fois où le terme est apparu dans le texte.

  2. **TF relative :** c'est le rapport entre le nombre de fois qu'un terme est apparu dans le texte sur le nombre de tous les termes du texte. Cette dernière est utilisée généralement pour limiter l'impacte de la longueur des textes. En effet, un terme qui apparait 6 fois dans un texte de 100 termes n'a pas le même degré de discrimination que celui qui apparait le même nombre de fois (6 fois) dans un texte de 20 termes.

     $$TF_{t,d} = \frac{N_{t,d}}{S_t} \tag{1.2}$$

     où $N_{t,d}$ est le nombre de fois que le terme t est apparu dans le document d, et S_t est le nombre de tous les termes du document.

  3. **TF booléenne :** se contente juste de la présence ou de l'ab-

sence du terme dans le texte.

$$TF = \begin{cases} 0 \\ ou \\ 1 \end{cases} \tag{1.3}$$

Le principal inconvénient de la fréquence des termes est le fait qu'il est possible, et d'ailleurs c'est un cas très probable en pratique, qu'un terme apparait avec une fréquence assez grande dans tous les documents d'un corpus. Dans ce cas, le terme en question perd toute sa notion de discrimination relative au degré de présence.

- **Fréquence documents inverses (IDF) :** elle permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus.

$$Idf_t = \log \frac{N_{Doc}}{Doc_t} \tag{1.4}$$

Où N_{Doc} est le nombre de documents dans le corpus, et Doc_t est le nombre de documents dans lesquels le terme t est apparu.

- **TF-IDF :** cette mesure permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus.

$$W_t = Tf_{t,d} \times Idf_t \tag{1.5}$$

- **Le modèle LSI/PLSI :** le modèle latent semantic indexing (LSI) découle du modèle vectoriel standard. Il tente de prendre en considération la structure sémantique des termes pour la représentation des documents. Dans ce modèle, les documents sont représentés dans un espace réduit de termes d'indexation ([42], [43], [44]). Les techniques LSI utilisent, dans un premier temps, une matrice M(documents X unités

linguistiques), dans laquelle chaque élément W_{ij} est une pondération en fonction du nombre d'occurrences du terme T_j dans le document D_i. Soit n le nombre de documents de la collection et t le nombre des termes d'indexation. La matrice M peut être représenté ainsi :

$$M = \begin{pmatrix} d_1 \\ d_2 \\ \dots \\ d_n \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \dots & W_{1t} \\ W_{21} & W_{22} & \dots & W_{2t} \\ \dots & \dots & W_{ij} & \dots \\ W_{n1} & W_{n2} & \dots & W_{nt} \end{pmatrix}$$

Une décomposition en valeurs singulières (SVD) de la matrice M est ensuite effectuée. Après cette décomposition, seuls les k premiers vecteurs propres sont intégrés. Dans LSI, la valeur représentée dans la matrice sur laquelle est appliquée la décomposition est définie comme étant le produit du poids local du terme, i.e. poids du terme dans le document, et du poids global du terme, i.e. poids du terme dans la collection de documents. Cette valeur peut aussi correspondre à la fréquence d'un terme donné dans un document donné. Ces valeurs ne sont que des pondérations proches de $TF - IDF$ ([42], [44]). Hoffmann a proposé un modèle probabiliste du Latent Semantic Indexing (PLSI). Il considère l'hypothèse que les documents sont associés à un certain nombre de sens (latents) et que les termes correspondent à l'expression de ces sens [45]. De façon probabiliste, notant W l'ensemble des mots, D l'ensemble des documents, tel que : $W = W_1, \dots, W_t$ et $D = D_1, \dots, D_n$, la probabilité de la paire observée $(d \in D, w \in W)$ est donnée par la formule suivante :

$$p(d, w) = p(w|d) * p(d) \tag{1.6}$$

Les dimensions de l'espace réduit du modèle LSI correspondent ici aux sens du modèle PLSI.

1.3.3 Classification supervisée

La classification des données textuelles a été largement étudiée dans différentes communautés telles que : la fouille de données, l'apprentissage automa-

tique et la recherche d'information. La classification supervisée de documents textuels vise à assigner des classes prédéfinies à des documents de texte [21]. Le problème de la classification est défini comme suit : nous possédons un ensemble d'apprentissage de document $D = \{d_1, d_2, ..., d_n\}$, de tel que chaque document d_i de cet ensemble est étiqueté avec un label l_i appartenant à l'ensemble $L = \{l_1, l_2, ..., l_k\}$. La tâche sera de trouver un modèle de classification (un classifieur)f qui peut assigner la bonne classe pour un nouveau document teste :

$$D \longrightarrow L \qquad f(d) = l \tag{1.7}$$

Il existe plusieurs méthodes de classification supervisée de textes. Parmi les méthodes les plus connues nous citons les K-plus proches voisins (k-Nearest Neighbour K-NN), les SVM (Support Vector Machine), la classification naïve bayésienne, et les arbres de décision.

1.3.4 Clustering ou Classification non supervisée

Le clustering est l'un des plus connu algorithmes de fouille de données, et qui a été largement étudié dans le contexte des données textuelles. Il a été appliqué dans plusieurs domaines tels que : la classification ([46],[12]), la visualisation ([47]), et l'organisation des documents [48]. Le clustering des documents textuels consiste à trouver des groupes de documents similaires dans une collection de documents. La similarité entre documents est calculée en utilisant une fonction de similarité spécifique. Le clustering des données textuelles peut être appliqué sur plusieurs niveau de granularité, où les clusters peuvent être des documents, des paragraphes, des phrases ou des termes,...etc. Dans la littérature, il existe de nombreux algorithmes de clustering pouvant être utilisés dans le contexte des données textuelles. Le document textuel peut être représenté par un vecteur binaire, c'est-à-dire en considérant la présence ou l'absence de mot dans le document. Ou nous pouvons utiliser des représentations plus raffinées qui impliquent l'utilisation des méthodes de pondération telles que $TF - IDF$.
Néanmoins, de telles méthodes ne fonctionnent généralement pas bien pour le clustering de textes, puisque les données textuelles ont un certain nombre de caractéristiques distinctes qui exigent la conception d'algorithmes spécifiques au textes.

On distingue deux catégories d'algorithmes de clustering : hiérarchiques et non-hiérarchiques.

- Dans le clustering hiérarchique (CH), les sous-ensembles créés sont emboîtés de manière hiérarchique les uns dans les autres. On distingue deux approches : ascendante et descendante. Dans l'approche descendante, nous commençons avec un cluster qui inclut tous les documents. nous divisons récursivement ce cluster en sous-groupes. Dans l'approche ascendante, chaque document est initialement considéré comme un individuel cluster. Ensuite, successivement, les groupes les plus similaires sont fusionnés jusqu'à ce que tous les documents soient inclus dans un seul cluster ([49], [50], [51]).

- Dans le clustering non-hiérarchique, les documents ne sont pas structurés de manière hiérarchique. Si chaque document ne fait parti que d'un sous-ensemble, on parle de partition. Si chaque document peut appartenir à plusieurs clusters, avec la probabilité p_i d'appartenir au cluster i, alors on parle de recouvrement.

1.3.4.1 Clustering probabiliste et les modèles de sujet

Les modèles de sujets (topic models) sont considérés parmi les algorithmes les plus populaires du clustering probabiliste, qui ont attiré beaucoup d'attention ces dernières années. L'idée principale de ces modèles ([2], [52], [45]) est de créer un modèle génératif probabiliste pour le corpus de documents textuels. Dans les modèles de sujets, les documents sont représentés par un mélange de sujets, où chaque sujet est représenté par une distribution probabiliste de mots. Les deux principaux modèles de sujets sont l'analyse sémantique latente probabiliste (PLSA) [45] et l'allocation de Dirichlet latente (LDA) [2]. Dans cette section, nous présentons la méthode LDA sur laquelle nous avons basé nos travaux.

Latent Dirichlet Allocation (LDA) :
L'algorithme de clustering probabiliste de documents *latent dirichlet allocation* (LDA) est devenu en l'espace d'une dizaine d'années l'un des

plus cités dans la littérature du domaine de la classification. Cet algorithme a la particularité de permettre à un document d'appartenir à plusieurs thématiques dans des proportions variables. Celui-ci se base sur un hyper-paramètre encore peu étudié dans la communauté scientifique, le paramètre α qui contrôle la variabilité des thématiques pour chaque document. Ce paramètre correspond à l'unique paramètre de la distribution de Dirichlet. Il définit la probabilité initiale des documents dans le contexte du LDA. À chaque extrême du spectre des valeurs que l'on peut assigner à ce paramètre, il devient possible de limiter chaque document à une seule thématique, jusqu'à forcer tous les documents de partager toutes les thématique uniformément.
Le paramètre α est un vecteur dont la longueur correspond au nombre de thématiques et qui est généralement fixé à une valeur constante. Cette valeur peut être soit déterminée arbitrairement, soit estimée durant la phase d'apprentissage. Une valeur faible amène la classification vers un petit nombre de thématiques par document, et à l'inverse une valeur élevée amène à assigner plusieurs thématiques par documents.
Le LDA est un modèle bayesien hiérarchique à 3 couches (Figure 1.2) : chaque document est modélisé par un mélange de topics (sujets) 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 (Expectation-Maximisation) [53].

1. **Le modèle LDA :** la Figure 1.2 représente le modèle graphique du LDA. Nous détaillons ci-dessous les différents termes et paramètres du modèle :
   - Un mot w est la donnée discrète, correspondant à l'indice d'un mot dans un vocabulaire fixe de taille V . On peut considérer que w est un vecteur de taille V de composantes toutes nulles sauf pour la composante i où i est l'indice du mot choisi ($w^i = 1$).
   - Un document est un N-uplet de mots, $w = (w_1, ..., w_N)$.
   - Un corpus est une collection de D documents,$D = (w_1, ..., w_D)$.

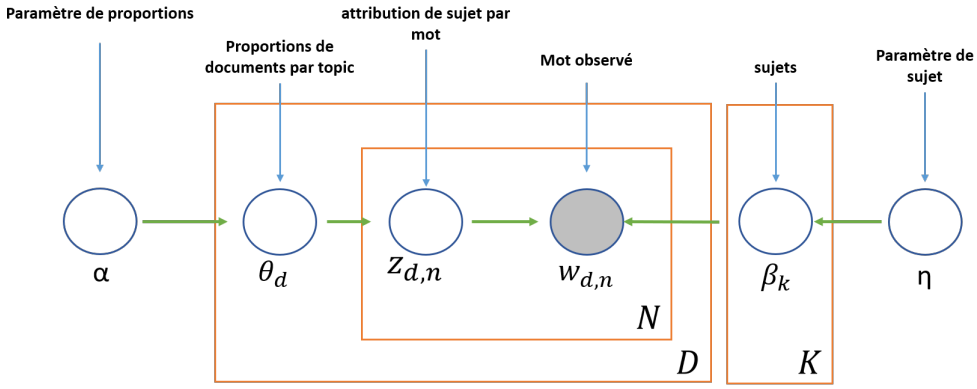

FIGURE 1.2 – Représentation de LDA sous forme de modèle graphique (adapté de [2])

- Les variables $Z_{d,n}$, représentent le sujet choisi pour le mot $w_{d,n}$.
- Les paramètres θ_d représentent la distribution de sujets du document d.
- et η définissent les distributions à priori sur θ et β respectivement, où β_k décrit la distribution du sujet k.

2. **Processus de génération :** le processus génératif suivi par LDA pour un document w est le suivant (voir le modèle graphique de la Figure 1.2)[2]
   (a) Choisir θ $Dirichlet(\alpha)$.
   (b) Pour chaque mot w_n :
   - Choisir un sujet z_n $Multinomial(\theta)$
   - Choisir un mot w_n $Multinomial(\beta_k)$, avec $k = z_n$.
3. **Inférence :** le problème principal de l'inférence pour LDA est celui de déterminer la distribution à posteriori des variables cachées étant donné le document (les paramètres α et β) [2] :

$$P(\theta, z|w, \alpha, \beta) = \frac{P(\theta, z, w|\alpha, \beta)}{P(w|\alpha, \beta)}) \tag{1.8}$$

Cette distribution est très difficile à calculer. 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, le Markov chain Monte Carlo ou le Gibbs sampling.

D'une façon générale nous pouvons décrire le LDA comme étant un modèle génératif probabiliste dont l'idée de base est qu'un document est un mélange probabiliste de thématiques latentes (c'est à dire cachées). Ensuite, chaque thématique est caractérisée par une distribution de probabilité sur les mots qui lui sont associée. On voit donc que l'élément clé est la notion de thématique, c'est à dire que la sémantique est prioritaire sur la syntaxe (la notion de terme ou mot).
Le LDA génère les mots dans un processus en deux étapes : les mots sont générés à partir des sujets et des sujets sont générés par des documents. Plus formellement, la distribution des mots étant donné que le document est calculée comme suit :

$$P(w_i|d) = \sum_{j=1}^{k} p(w_i|z_j) * p(z_j|d) (1.9)$$

1.4 Conclusion

Afin de faire face aux grandes quantités de données textuelles disponibles, il est devenu indispensable d'offrir aux utilisateurs de nouvelles approches qui vont leur permettre d'appréhender, de la manière la plus aisée possible, les éléments significatifs contenus dans ces documents. Les travaux sur l'analyse de documents textuels ont été largement étudiés par différentes communautés.
Dans ce chapitre, nous avons présenté un aperçu général sur les techniques d'exploration des données textuelles. Dans notre travail de thèse, nous nous sommes intéressés à ces techniques afin de développer de nouvelles approches permettant une alimentation sémantique des entrepôts de textes. Dans la suite, nous présentons les principaux travaux traitant d'entreposage des données textuelles.

Chapitre 2

Les entrepôts de textes

2.1 Introduction

Les systèmes décisionnels (SID) ont apporté une première solution aux entreprises désirant analyser les grandes masses de données provenant de leurs systèmes sources, afin d'améliorer leur performances. Les SID classiques ont déjà fait leurs preuves dans le domaine de l'analyse des données simples. Or ces systèmes ne sont pas adaptés à l'analyse des documents textes, ce qui met en évidence la nécessité de créer de nouveaux systèmes pour les données textuelles. L'entreposage de ces dernières demeure encore aujourd'hui l'une des difficultés majeures, et implique de nombreux problèmes, notamment ceux de leur modélisation et leur intégration d'une part et leur analyse d'autre part.

Les entrepôts de textes sont apparus comme une nouvelle solution permettant une analyse multidimensionnelle des données textuelles. La nature complexe de ces données nécessite un traitement bien particulier, qui prend en compte leur sémantique. Dans la littérature, des méthodes de recherche d'information et de fouille de données ont donné de très bons résultats pour l'exploration des données textuelles. L'idée clef derrière ces entrepôts de textes est de faire un couplage entre les techniques de fouille de données et de recherche d'information d'un coté, et les techniques OLAP de l'autre coté.

Dans ce chapitre, nous présentons un état de l'art sur les principaux travaux traitant l'analyse multidimensionnelle des données textuelles. Ce chapitre est organisé en trois partie. La première partie est consacrée à l'étude des différents modèles multidimensionnels dédiés à l'analyse des données textuelles. Nous classifions ces modèles selon deux catégories :

(i) modèles extensifs et (ii) modèles à nouveaux concepts. La deuxième partie présente une étude comparative de ces travaux selon cinq critères de comparaison : la prise en compte de la structure des données textuelles, la prise en compte de la sémantique, la flexibilité des modèles étudiés, la prise en compte des mesures textuelles et la définitions de nouveaux opérateurs OLAP pour les données textuelles. la troisième partie expose les limites des travaux actuels et présente les challenges regardant l'entreposage des données textuelles.

2.2 Modèles Multidimensionnels d'analyse des données textuelles

Dans ces dernières années, le domaine de l'entreposage et l'analyse en ligne OLAP a connu un grand nombre de travaux traitant les données complexes. Ces travaux couvrent les différents aspect de stockage et d'analyse ; nous citons les travaux sur : l'intégration des données web([54], [55]) ; les entrepôt de données multimédia ([56], [57], [58], [59], [60]) ; les données semi-structurées représentés en XML(Extensible Markup Language)([61], [62], [63]) ; et le stockage des données non structurées ([64] ,[65], [66]), ...etc.
Les données textuelles représentent un type particulier des données complexes. Afin de les considérer dans l'analyse multidimensionnelles, plusieurs travaux ont été élaboré. Ces travaux sont groupés selon [6] en deux catégorie :

2.2.1 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) [67] : 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 tout 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 multidimensionnels. 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 tout 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'indexe 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.

- Dans [68], les auteurs ont proposé un modèle basé sur un schéma en étoile nommé *DocCube*, 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.

- Dans [69], les auteurs 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 méta-données 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 méta-données 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 granularité.

- Dans [70], les auteurs ont proposé un cube de textes nommée *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*.

- Dans [3], les auteurs 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, la figure 2.1 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 noeud enfant représente un événement spécialisé de l'événement représenté dans le noeud père, par exemple, *Undershoot* et *Overshoot* sont deux anomalies spécifiques à l'événement *Anomaly Altitude Deviaton*.

  Topic Cube propose deux mesures probabilistes : la distribution d'un mot dans un thème *word distribution of a topic* $p(w_i)$ 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 d_j 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. La figure 2.2 décrit le schéma en étoile d'un *Topic Cube*

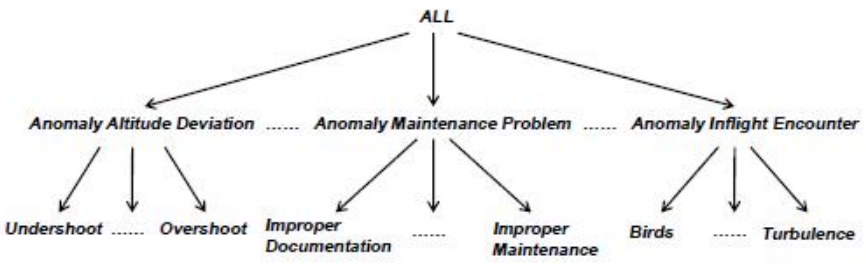

FIGURE 2.1 – Exemple d'un arbre d'hiérarchie de thèmes "cas d'anomalies" [3].

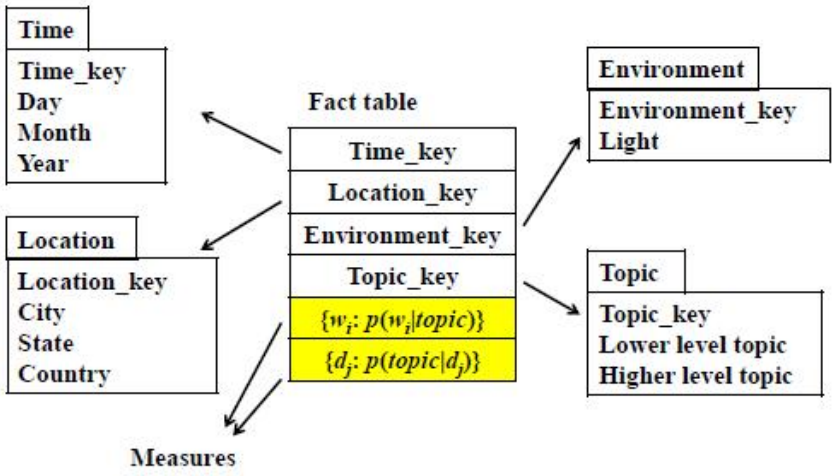

FIGURE 2.2 – schéma en étoile d'un *Topic Cube* [3].

- Dans [71], les auteurs 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 transactionn [72]. 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.

- Dans [73], les auteurs ont proposé *MicroTextClusterCube*, un modèle basé sur un schéma en étoile. Ils ont proposé d'introduire une nouvelle mesure d'analyse $(mean_i, size_i)$ qui représente 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 *micro-cluster*) 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 de granularité structurels.

2.2.2 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 :

- [74] 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, [75].

- [76] 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 noeuds centraux ; chaque noeud modélise les dimensions compatibles pour une même analyse. Son modèle est basé sur la généralisation du concept de constellation de [77]. 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,[78] se sont basés sur le paradigme objet grâce auquel il est possible de repré-

senter 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. La figure suivante 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 :

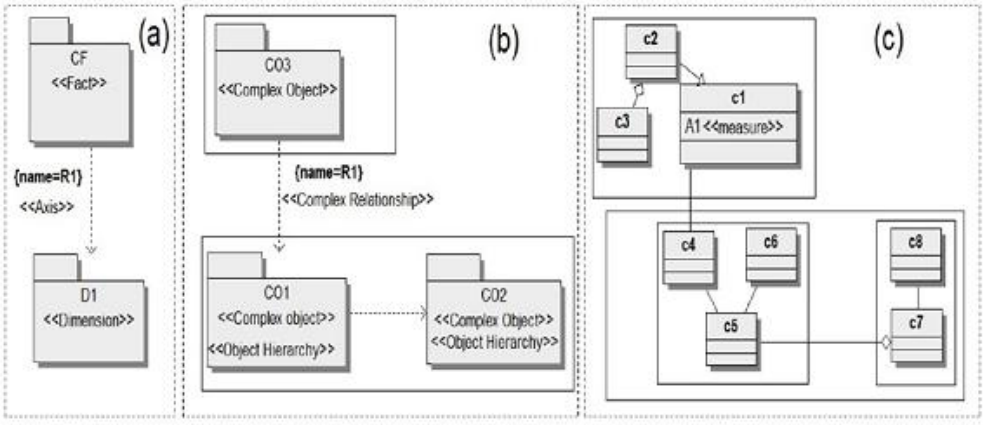

FIGURE 2.3 – Modèle multidimensionnel d'objets complexe à trois niveaux [79].

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.

2.3 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 de 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 :

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 revue de presse. 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, qui permet une analyse sur de différents niveaux de granularité, n'est pas prise en considération.

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 entrepôt de données.

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'analyse peu flexible, car le décideur se voit contraint d'employer ces faits comme sujets [76]. 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.

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.

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 [74]		X	-	-	-	-	X	Flexible
Mire [67]	X		X	-	Drill down et Roll-up	-	-	Non flexible
DocCube [68]	X		-	Score(Dd)	Drill down et Roll-up	X	-	Non flexible
D.cube [69]	X		-	Count	Drill down et Roll-up	-	-	Non flexible
Galaxie [76]		X	X	AVG-KW, Top-KW	Drill down et Roll-up	X	X	Flexible

TextCube [70]	X		X	-	pull-up et push-down	X	-	Non flexible
TopicCube [3]	X		X	-	Drill down et Roll-up	X	-	Non flexible
MMAP-structure [71]	X		-	-	-	X	-	Non flexible
M.Cube [73]	X		-	-	-	X	-	Non flexible
MMOC [78]		X	X	utilisation du Top-KW	Drill down et Roll-up	-	X	Flexible
Diamond Model [80]		X	X	-	-	X	-	Non flexible

TABLE 2.1: Tableau comparatif

2.3.1 Bilan sur les limites des approches de modélisation actuelles

Malgré que les modèles extensifs ont permis d'effectuer des analyses multidimensionnelles sur les données textuelles, nous constatons qu'ils sont toujours limités et ne traitent que quelques aspects de 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.

2.4 Conclusion

Analyser les données textuelles afin de pouvoir tirer profit des informations qu'elles contiennent est devenu essentiel, à cause de leurs volumes importants et de la quantité d'information qu'elles contiennent. Nous avons présenté dans ce chapitre un survol de modèles dédiés à la représentation multidimensionnelle des données textuelles. Nous avons pu catégoriser l'ensemble de ces modèles selon le type de concept utilisé en deux familles : modèles extensifs basés sur les deux concepts de base des modèles classiques **fait** et **dimension**, et modèles à nouveaux concepts, qui ont proposé de nouveaux modèles basés sur de nouveaux concepts.

Nous avons présenté aussi une étude comparative entre ces différents modèles, selon cinq critères de comparaison : la prise en compte de la structure des données textuelles, la prise en compte de leur sémantique, la flexibilité d'analyse, la prise en compte d'une mesure textuelle et la définitions de nouveaux opérateurs OLAP pour les données textuelles.
Ce survol de modèles multidimensionnels dédiés à l'analyse textuelle nous a permis de constater la diversité des modèles proposés et de décerner les problèmes majors liés à l'analyse des données textuelles.
Dans ce qui suit, nous présentons nos contributions relatives à l'entreposage et l'analyse en ligne des données textuelles.

Partie II

Contributions

Chapitre 3

Une Approche ETL pour l'alimentation d'entrepôt de textes

3.1 Introduction

Un entrepôt de données constitue avant tout une alternative pour l'intégration de diverses sources de données. Un système d'intégration a pour objectif d'assurer à un utilisateur un accès à des sources multiples, réparties et hétérogènes, à travers une interface unique. En effet, l'avantage d'un tel système est que l'utilisateur se préoccupe davantage de ce qu'il veut obtenir plutôt que comment l'obtenir, l'objectif étant l'obtention d'informations. Ainsi, cela le dispense de plusieurs tâches telles que chercher et trouver les sources de données adéquates, interroger chacune des sources de données en utilisant sa propre interface et combiner les différents résultats obtenus pour finalement disposer des informations recherchées.

L'alimentation d'un entrepôt de données est une phase essentielle dans le processus d'entreposage. Elle se déroule en plusieurs étapes : extraction, transformation, chargement et rafraîchissement des données, qui sont prises en charge par le processus d'ETL (Extracting, Transforming and Loading). Ce processus constitue la phase de migration des données de production dans le système décisionnel après qu'elles ont subi des opérations de sélection, de nettoyage et de reformatage dans le but de les homogénéiser. Cette phase constitue une étape importante et très chronophage dans la mesure où on l'estime à environ 80% du temps de mise en place de la solution décisionnelle.
À l'heure actuelle, les techniques utilisées dans le processus ETL offrent de très bon résultat lorsque les données sont structurées, et/ou semi structurées. Les limites de ces techniques sont évidentes lorsqu'il s'agit

de manipuler et d'intégrer des données complexes, tel que les données textuelles. Ces dernières ont la particularité de posséder de profondes différences de nature par rapport aux données simples numériques. Une de leurs particularités est qu'il est impossible de trouver une forme standard de représentation d'une donnée textuelle. Aussi, il n'est pas possible de leur associer du sens de façon systématique [81]. Dans la littérature de nombreuses contributions ont traité la modélisation des processus ETL ([82], [83],[84],[85],[86],[87],[88],[89], [90],[91],[92],[93]). Tandis que peu de travaux ont travaillé sur l'adaptation des processus ETL aux données complexes.

Dans la littérature plusieurs travaux ont été proposé dans le cadre de la fouille de textes ([94],[95], [96],[97],[98]). En revanche, dans celui relevant de l'entreposage et de l'analyse en ligne des données textuelles, peu de travaux se sont intéressés à cette problématique ([99]).

Construire un entrepôt de textes nécessite le développement d'un processus ETL qui prend en charge la complexité des données textuelles. Ce processus doit permettre l'extraction des données textuelles, les métadonnées et le contenu sémantique des données textuelles à partir des documents, transformer les données extraites pour correspondre au schéma de l'entrepôt et les charger dans l'entrepôt de textes. La tâche d'extraction de données textuelles et les métadonnées peut être réalisée en adaptant les outils ETL classiques. Cependant, l'extraction sémantique nécessite un traitement spécifique. Le contenu sémantique du texte pourrait être : termes pertinents, des sujets pertinents, des phrases pertinentes,... etc. Pour détecter ce contenu sémantique, de nombreuses applications proposent d'utiliser les modèles de sujets qui sont une suite d'algorithmes qui permettent de découvrir la structure thématique cachée dans les collections de documents. Ces algorithmes nous aident à développer de nouvelles façons de rechercher, de parcourir et de résumer les grandes archives de textes. Une variété de modèles de sujets probabilistes tels que LDA (Latent Dirichelet allocation) [2], PLSA (Probabilistic latent semantic allocation) [45] ou LSA (Latent semantic

analysis) [100], ont été utilisés pour analyser le contenu des documents et le sens des mots. Ces modèles utilisent la même idée fondamentale, qu'un document est un mélange de sujets. Chaque sujet est représenté par une distribution de mots $T_i(word_j/P(word_j|T_i))$ $i, j \in \mathbb{N}$ et la probabilité $P(T_i|Doc_k)$ $i, k \in \mathbb{N}$, tel que $P(word_j|T_i)$ est la probabilité du mot $word_j$ dans le sujet $Topic_i$ et $P(T_j|Doc_k)$ est la probabilité du sujet $Topic_j$ dans le document Doc_k. Toute fois, la distribution de mots donnée pour chaque sujet nous nous permet pas d'extraire les concepts sémantiques des sujets. Pour traiter cet aspect, nous proposons une approche qui associe l'utilisation de la méthode LDA avec la taxonomie ODP afin de reconstruire la sémantique de la distribution de mots, afin d'assurer une alimentation sémantique des entrepôts de textes. Notre approche vise à construire pour chaque document textuel une hiérarchie de concepts qui le représente.

L'approche ETL proposée fournit des hiérarchies de concepts représentant les documents textuels, afin de raffiner cet ensemble à un ensemble plus restreint de concepts pertinents, qui captent avec précision le sujet du document, nous proposons d'introduire la pertinence contextuelle dans le calcule des poids des concepts au sein des hiérarchies obtenues, à travers la technique de rétro-propagation de la pertinence du domaine du document aux hiérarchies de concepts qui le représente.

Dans ce chapitre, nous proposons une approche d'extraction, de transformation et de chargement de données textuelles dans un entrepôts de textes. L'approche proposée associe l'utilisation de la méthode LDA avec la taxonomie ODP comme sources externes de connaissance, pour extraire les sujets pertinents présents dans les documents textuels. La suite de ce chapitre est organisée comme suit : la section 2 évoque la problématique d'intégration de données dans le processus décisionnel. La section 3 présente l'approche ETL proposée et illustre ses différentes étapes. La section 4 introduit la pertinence contextuelle et présente la formule utilisée pour le calcule des poids des concepts. La section 5 conclue le chapitre et introduit le chapitre suivant.

3.2 Problématique d'intégration de données

La constitution d'entrepôts de données est une réponse au problème de l'intégration d'une grande quantité de données variées, relatives à un certain domaine d'application, et stockées physiquement dans différentes sources de données. L'entrepôt de données regroupe certaines informations sous une forme exploitable par des traitements utiles pour l'aide à la décision. Ces informations, qui sont potentiellement pertinentes pour telle ou telle catégorie de décideurs du domaine, doivent être :

- **Extraites :** accéder à la majorité des systèmes de stockage de données (SGBD, documents XML, ...) et récupérer les données pertinentes.
- **Transformées :** transformer les données avant de les charger dans l'entrepôt de données.
- **Chargées :** charger les données dans l'entrepôt.

De manière générale, la problématique peut se résumer comme suit : comment permettre à l'administrateur d'un entrepôt de données de récupérer des données pertinentes qui se situent dans des sources différentes de données distribuées, hétérogènes ; et les transformer afin de les charger dans l'entrepôt.

Des solutions logicielles sont alors nécessaires à leur intégration et à leur homogénéisation, Ces outils ont pour objet de s'assurer de la cohérence des données de l'entrepôt et d'homogénéiser les différents formats trouvés dans les bases de données opérationnelles.

Les données de l'entrepôt sont pour la plupart, issues de différentes sources de données hétérogènes. Cette hétérogénéité est expliquée par de différentes raisons, tel que : la différence dans le matériel utilisé par les systèmes sources, les système logiciels , les systèmes de gestion de base de données, et les données. Ces dernières représentent les ressources les plus précieuses disponibles dans les différents systèmes d'informations. Dans [4] les auteurs ont proposé une classification qui présente les conflits

des données en trois niveaux : conflits syntaxiques, conflits structurels et conflits sémantique (Figure 3.1) :

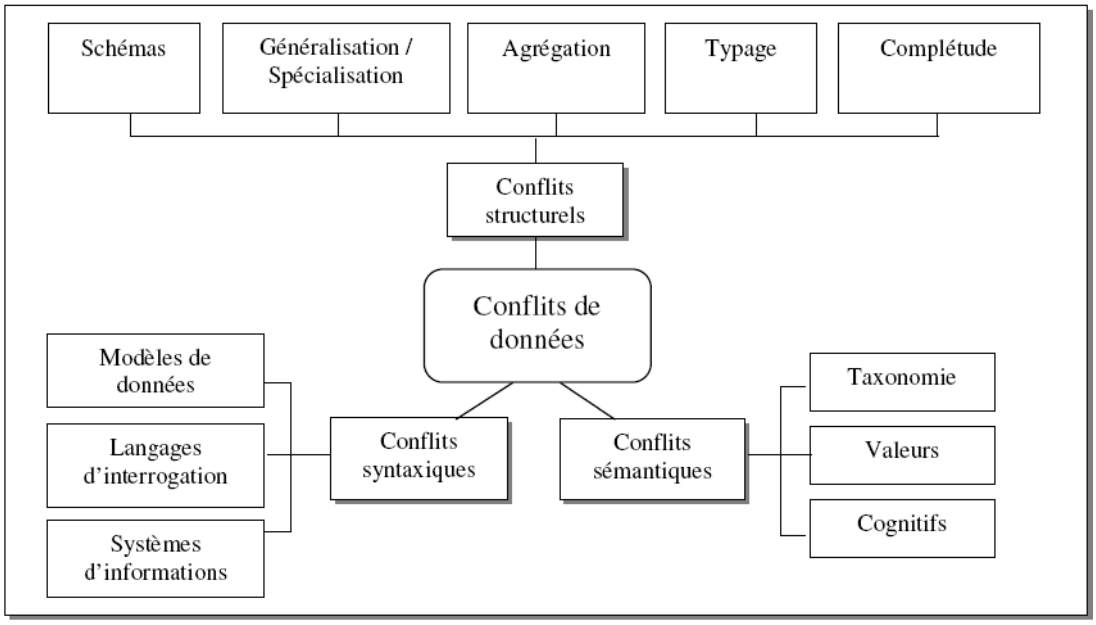

FIGURE 3.1 – Classification des conflits de données [4]

- **Conflits syntaxiques :** résultent de l'utilisation de modèles de données différents d'un système à l'autre. Des concepts différents sont utilisés pour structurer la même information (relation dans le relationnel, classe dans le modèle objet, balise XML, etc.)
- **Conflits structurels :** résultent d'une structuration et classification différente des informations. Ils sont étroitement liés aux choix de conception, nous distinguons :
  - **Les conflits de schémas :** résultent de l'utilisation de différents concepts pour représenter le même objet. Une information peut être représentée par une entité dans un système (S1) et par une relation dans un autre système (S2).
  - **Les conflits de généralisation / spécialisation :** résultent des différences de hiérarchisation des informations, par exemple dans un système (S1) les personnes sont regroupées dans un

seul objet PERSONNE alors que dans (S2) on utilise deux objets HOMME et FEMME pour représenter les personnes.

- **Les conflits d'agrégation :** résultent d'un niveau de granularité différent de deux systèmes. La valeur d'un attribut sur un système correspond à une agrégation des valeurs de plusieurs attributs sur un autre.
- **Les conflits de typage :** résultent des différences de typage pour le même objet dans les différents systèmes sources.
- **Les conflits de complétude :** apparaissent lorsque des objets se correspondent partiellement sur les différents systèmes. C'est-à-dire qu'une partie d'un objet du système (S1) trouve une correspondance sur le système (S2).

- **Conflits sémantiques :** proviennent des différences d'interprétation des informations partagées entre différents domaines d'application. Nous distinguons plusieurs types de conflits sémantiques :
  - **Les conflits de noms :** se retrouvent lorsque les différents schémas utilisent des noms différents pour représenter le même concept ou propriété (synonyme), ou des noms identiques pour des concepts ou des propriétés différents (homonymes).
  - **Les conflits de contexte :** se retrouvent dans le cas où les concepts semblent avoir la même signification dans deux schémas mais sont différents à cause de leur contexte. Par exemple le poids d'une personne dépend de la date où elle se pèse.
  - **Les conflits de mesure de valeur :** sont liés à la manière de coder la valeur d'un concept du monde réel dans un système de mesure. Ils se retrouvent par exemple dans le cas où on utilise des unités différentes pour mesurer la valeur d'une même propriété dans différentes sources (dans une source, on utilise le dinar, et dans une autre on utilise l'euro).

Dans le contexte décisionnel, plusieurs approches d'intégration de données peuvent être utilisées pour résoudre les différents conflits de données. Tandis que dans le cadre du processus d'entreposage de données textuelles, la problématique peut être différente. Les données textuelles

ont la particularité d'être non seulement hétérogènes mais aussi présentées sous un format brut. Donc ces données doivent subir plusieurs traitements avant de pouvoir les intégrés dans un entrepôt de données. Dans ce chapitre, nous offrons une solution aux problèmes d'intégration de données textuelles dans un entrepôt de données à travers un processus ETL adapté aux données textuelles.

3.3 Approche ETL proposée

L'intégration des données textuelles dans un but d'analyse implique de nombreux problèmes. Les données textuelles sont présentées sous un format brut, donc avant de les intégrées dans l'entrepôt, il est nécessaire d'extraire d'abord de l'information pertinente à partir des documents de textes. Ces informations pertinentes peuvent être des métadonnées, des termes, des phrases, des sujets...etc.

Dans cette section nous présentons une approche ETL pour l'alimentation d'entrepôt de textes. L'approche proposée est divisée en trois phases principales présentées dans la figure 3.2.

3.3.1 Phase d'extraction

Dans cette phase, les données textuelles extraites à partir des différentes sources sont préparées et filtrées. Nous distinguons trois étapes :

3.3.1.1 La préparation de données

Dans cette phase, un ensemble de traitements sont enchainés sur le texte du document, afin de filtrer les termes pertinents en éliminant les segments de texte (tokens) non pertinents. Les principales tâches effectuées concernent :

- La Tokenisation ou segmentation du texte ; elle consiste à parcourir le texte en vue de récupérer les termes et supprimer les caractères spéciaux et la ponctuation.

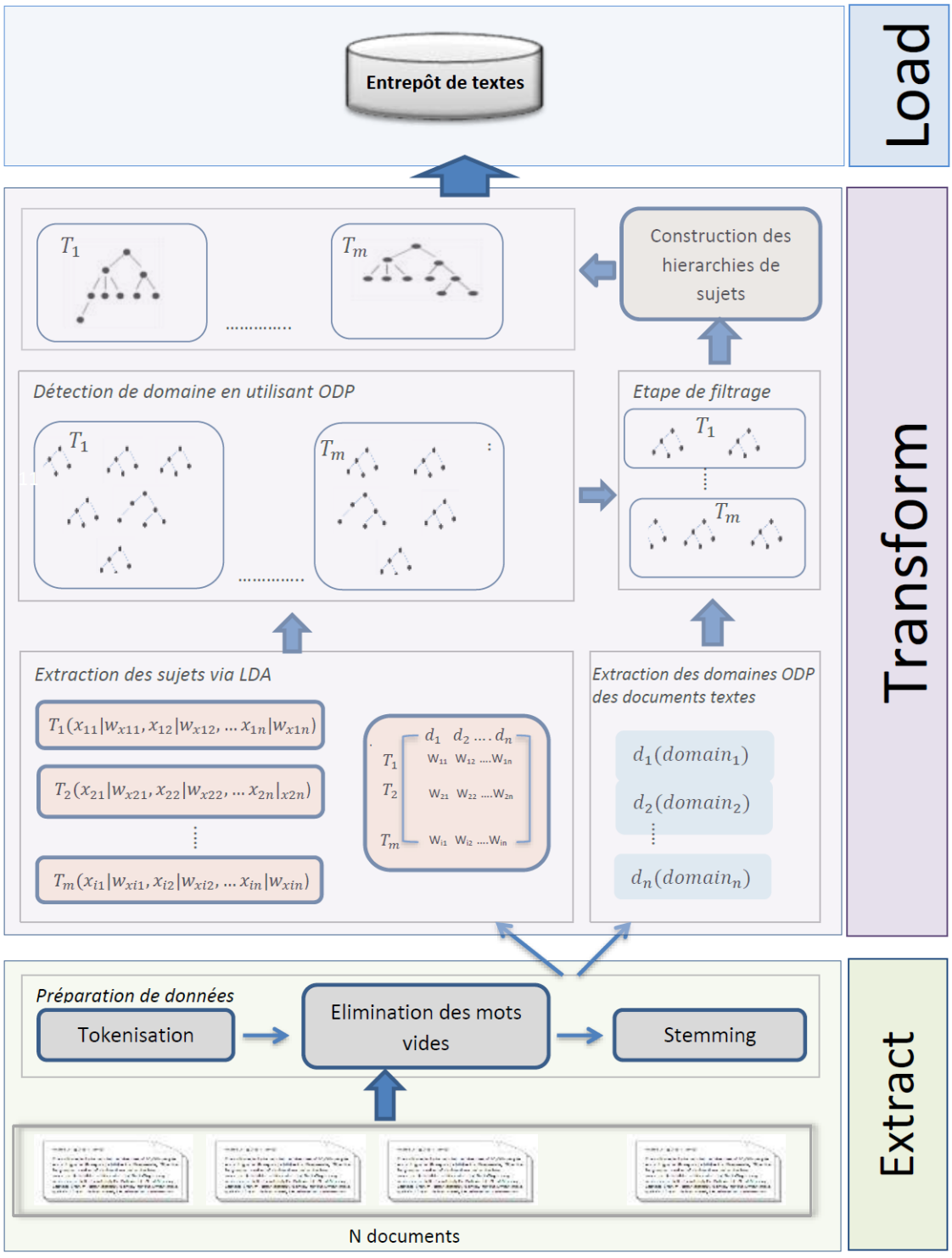

FIGURE 3.2 – Architecture de l'approche ETL

- L'élimination des mots vides ; consiste à éliminer les mots outils de la langue en utilisant une liste de mots vides.

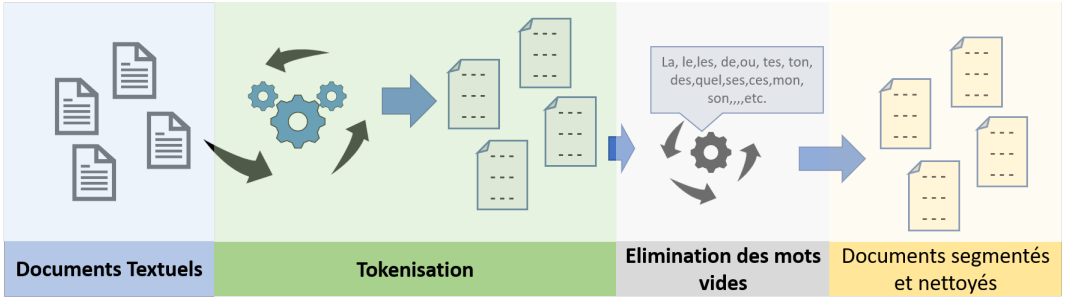

FIGURE 3.3 – Tokenisation et nettoyage des documents textuels

- La Racinisation ou stemming ; consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculin singulier.

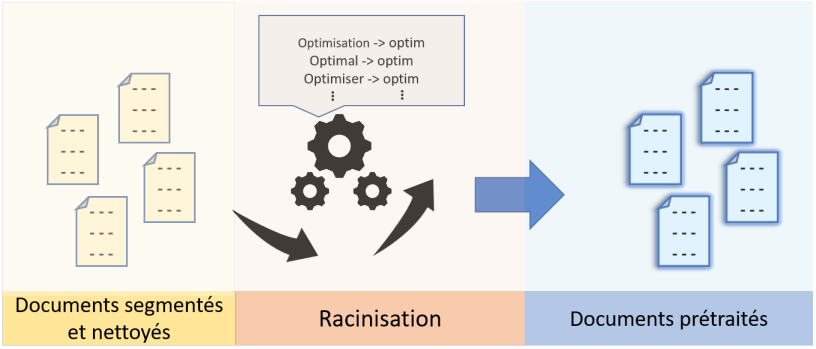

FIGURE 3.4 – Racinisation des données textuelles

Exemple. Considérons le document doc_1 composé du texte suivant : "ETL is a process in data warehousing responsible for pulling data out of the source systems and placing it into a data warehouse." . En appliquant la phase préparation de données, on obtiendra : doc_1= "ETL be process data warehouse responsible pull data source system place data warehouse"

3.3.1.2 L'extraction des entités textuelles

Cette étape inclus deux tâches principales :

- **L'extraction des termes :** consiste à extraire les termes présents dans le document.

- **L'extraction des métadonnées :** pour les données textuelles, les métadonnées sont définies comme une information structurée qui explique, décrit ou localise l'élément texte. Il existe, pour cela, une panoplie d'outils et de méthodes pour extraire de façon automatique ces métadonnées. A titre d'exemple, on peut citer :

  - **Metadata Extraction Tool** [1] **:** c'est un outil libre (Apache Public License version 2) développé par Sytec Resources pour la bibliothèque nationale de Nouvelle-Zélande. Il permet d'extraire automatiquement et en masse des métadonnées à partir de fichiers numériques. Il est capable de prendre en charge une grande variété de formats de fichiers (PDF, fichiers bureautiques, fichiers images, fichiers sons, langages de balisage...) et produit en fin de traitement un fichier XML rassemblant les métadonnées extraites des fichiers. Il est utilisable à la fois via une interface pour Windows et des lignes de commande sous Linux, ce qui permet de lancer des traitements manuels ou automatisés (batch).

  - **TeamBeam [101] :** spécialisé dans les articles scientifiques et qui permet d'extraire un certain nombre de métadonnées structurées, telles que le titre, le nom du journal et le résumé, ainsi que des informations sur les auteurs de l'article (par exemple, noms, adresses e-mail, affiliations).

  - **Pdf2gsdl [102] :** est un programme de ligne de commande qui permet aux utilisateurs d'extraire automatiquement les métadonnées à partir des fichiers PDF.

Example. Considérons le document Doc_1 une publication scientifique, le titre du document, l'auteur, la date de publication,...etc, sont considérés comme des exemple de métadonnées associées au Doc_1.

1. http ://meta-extractor.sourceforge.net

3.3.2 Phase de transformation

Dans cette phase, les données extraites sont transformées pour correspondre au schéma de l'entrepôt de données. Elle est définie par deux étapes principales :

3.3.2.1 Extraction des domaine ODP des documents textes

Dans cette étape, nous identifions pour chaque document texte, son domaine ODP en utilisant l'API [2] textwise.

Exemple. Considérons les documents $Doc_1; Doc_2; ...Doc_n$. En identifiant les domaines des documents nous obtenons $ODP\_Domain(Doc_1 = Sport)$; $ODP\_Domain(Doc_2 = Society)..., ODP\_Domain(Doc_n = ...)$...etc.

3.3.2.2 Extraction des sujets via LDA en se basant sur ODP

Cette phase consiste à extraire les sujets en utilisant le modèle LDA, puis générer des concepts sémantiques pour les sujets extraits en utilisant la taxonomie ODP. Elle est définie par deux étapes :

1. **Extraction des sujets via LDA :** afin d'extraire les sujets cachés dans le corpus de texte, nous appliquant le modèle LDA sur les documents nettoyés. Pour ce fait, nous devons définir le nombre de sujets et le nombre de mots attribués à chaque sujet. Alors, chaque sujet est représenté par une distribution de mots, ex : $T_0\{low(0.5); stadium(0.8); sport(0.7); major(0.2); fight(0.2); football(0.4)\}$ avec la matrice présentée ci-dessous, qui définit le poids de chaque sujet dans chaque document :

$$\begin{array}{c} \\ T_1 \\ T_2 \\ ... \\ T_m \end{array} \begin{array}{c} \begin{array}{cccc} Doc_1 & Doc_2 & ... & Doc_k \end{array} \\ \begin{pmatrix} 0,6 & 0,1 & ... & 0,3 \\ 0,2 & 0,6 & ... & 0,2 \\ 0,35 & 0,05 & ... & 0,6 \\ 0,35 & 0,05 & ... & 0,6 \end{pmatrix} \end{array}$$

TABLE 3.1 – Matrice représentant le poids de chaque sujet par document

2. http ://www.textwise.com/demo

2. **Détection de domaine en utilisant ODP :** les sujets obtenus en appliquant la méthode LDA sont donnés par une distribution de mots, alors on ne peut pas identifier les concepts sémantiques de ses sujets. Pour répondre à cette problématique, nous proposons de présenter chaque sujet comme une distribution de domaines, en identifiant pour chaque mot dans la distribution d'un sujet tous les domaine ODP avec leurs différents niveaux. Afin de présenter chaque sujet par un ensemble de hiérarchies de domaines basé sur ODP, nous générons pour chaque mot toutes les hiérarchies possibles à partir de la taxonomie ODP.

3.3.2.3 Étape de filtrage

Un mot représente une unité très spécifique et peut être connecté à plusieurs domaine ODP. Alors, certaines hiérarchies de concepts obtenues en appliquant l'étape précédente, peuvent être considérées comme non pertinentes pour un document. Comme les domaines de document ainsi que les hiérarchies de concepts représentants les sujets sont basés sur ODP, nous définissons une hiérarchie de concepts comme étant non pertinente pour un document donné, si la racine de cette dernière est différente du domaine du documents. Dans cette étape, nous éliminons toutes les hiérarchies non pertinentes.

Example. Considérons le document doc_1 avec domaine égale à *Sport* et le sujet $Topic_0$ donné par la distribution de mot suivante :$Topic_0 = \{low(0.5)\,;\ stadium(0.8); sport(0.7);\ major(0.2); fight(0.2);\ football(0.4)\}$.
La figure 3.5 présente un exemple de hiérarchies pertinentes et non pertinente basées sur ODP pour le mot *Stadium*.

3.3.2.4 Construction des hiérarchies sémantiques

Dans cette phase, nous construisons pour chaque document textuel une hiérarchie sémantique pondérée, en combinant les hiérarchies de concepts qui lui ont été identifier par l'étape de filtrage. Pour calculer le poids de chaque concepts dans la hiérarchie nous proposons une fonction de pondération dé-

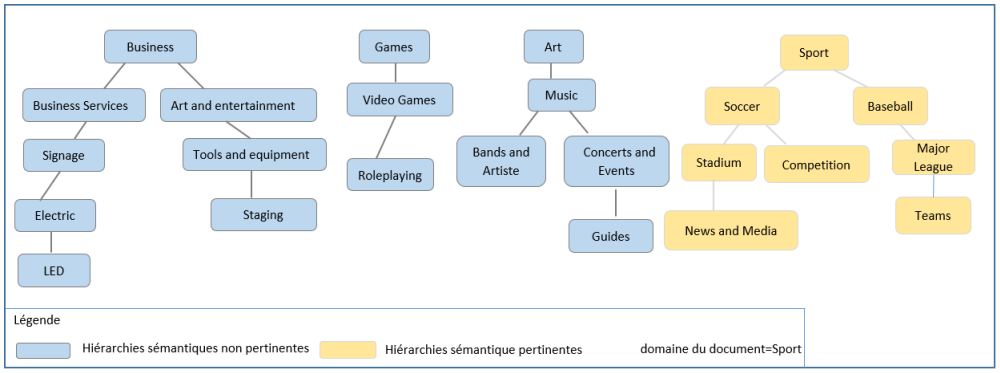

FIGURE 3.5 – Exemple de hiérarchies pertinentes et non pertinentes pour le mot *Stadium*

crite par la formule suivante :

$$P(Concept_j, Document_m) = \sum_{k=1}^{n} \sum_{i=1}^{l} \frac{Nconcept_j}{Nconcept} * P(w_i|topic_k) * P(topic_k|Doc_m) \tag{3.1}$$

tel que :

- $Nconcept_j$ est le nombre d'occurrence du $concept_j$ avec le mot $word_i$ en utilisant la taxonomie ODP.
- $Nconcept$ est le nombre de tous les concepts avec le mot $word_i$ en utilisant ODP.
- $P(w_i|topic_k)$ est la probabilité du mot $word_i$ dans $topic_k$.
- $P(topic_k|Doc_m)$ est la probabilité du sujet $topic_k$ dans le document Doc_m.

3.3.3 Phase de chargement

Dans cette phase, les données transformées sont chargées à l'entrepôt de textes. Cette phase est une étape mécha-nique et la moins complexe.

3.4 La pertinence contextuelle

Le problème de déterminer les entités et les concepts les plus pertinents dans un document donné est d'une importance croissante en raison de l'aug-

mentation constante de l'information disponible. Le problème n'est pas seulement de déterminer l'ensemble de toutes les entités et les entités nommées dans un document, mais de raffiner cet ensemble à un ensemble plus restreint de concepts intéressants et pertinents qui captent avec précision le sujet du document et qui sont intéressants pour une large base d'utilisateurs. L'approche proposée ci-dessus se concentre sur ce problème et présente une nouvelle technique pour extraire et classer les concepts clés dans un document textuel. notre approche utilise le modèle LDA avec la taxonomie ODP pour fournir à chaque document textuel une hiérarchie de concepts pondérés. Ce dernier est représenté par un arbre dont les nœuds sont des concepts pertinents et sa racine représente le domaine du document textuel. Dans [103], le contexte du document a été d'abord utilisé tout au long de la tâche de filtrage, dans laquelle nous utilisons le domaine du document pour éliminer les hiérarchies de concepts non pertinentes. Dans cette section, nous proposons de recalculer les poids des concepts fournis par notre approche ETL tout en tenant compte de la pertinence contextuelle. Nous définissons cette dernière comme l'impact de la pertinence du document textuel sur la pertinence des éléments qu'il contient, en d'autres termes, nous utilisons le contexte du document (dans notre cas, le domaine du document) pour réaffecter les scores dans la hiérarchie des concepts. Pour introduire la pertinence contextuelle, nous utilisons une rétro-propagation de la pertinence du nœud racine (domaine de document) dans notre hiérarchie vers les nœuds internes (concepts dans la hiérarchie) [5].

3.4.1 La rétro-propagation de pertinence

Dans la section précédente, nous avons présenté une nouvelle approche qui fournit des hiérarchies de concepts pondérés représentant les documents textuels. Une hiérarchie de concepts se compose d'un nœud racine qui représente le domaine de document textuel et des nœuds internes qui représentent les concepts pertinents dans ce document. Pour appliquer une rétro-propagation de la pertinence du nœud racine aux nœuds internes, nous proposons de recalculer la pertinence d'un nœud n en utilisant la formule suivante :

$$Weight(n_i, n_{Root}) = Weight(n_i) + Weight(n_{Root})^{(dist(n_i, n_{Root}))} \tag{3.2}$$

- n_i est le nœud feuille pour lequel le nouveau poids est recalculé
- n_{Root} est le nœud racine à partir du quel la rétro-propagation de la pertinence est effectuée.
- $dist(n_i, n_{Root})$ est la distance sémantique entre les nœuds n_i et n_{Root}, représentée par le nombre d'arêtes entre eux.

La rétro-propagation de pertinence dépend de deux paramètres :

- Le nœud racine n_{Root} à partir du quel la rétro-propagation de la pertinence est effectuée ; lorsque le poids du nœud racine est important, la propagation de la pertinence augmente.
- La distance entre le nœud racine n_{Root} et la feuille $noden_i$; Lorsque la distance entre les deux nœuds augmente, la rétro-propagation de la pertinence diminue.

3.5 Évaluation expérimental

Dans cette section, nous discutons l'évaluation expérimentale de nos propositions. Cette section est organisée comme suit : premièrement nous commençons par présenter le corpus de données utilisé dans nos expérimentations. Deuxièmement , nous présentons les deux paramètres utilisés pour l'évaluation de notre l'approche ETL : le rappel et la précision, qui ont été très utilisés dans la littérature pour l'évaluation des systèmes. Enfin, nous présentons comment le rappel et la précision ont été mesurés pour l'approche ETL et nous discutons les résultats obtenus.

3.5.1 Corpus de données

Pour faire nos expérimentations, nous avons constitué un corpus de taille moyenne d'articles de presse. Nous avons utilisé pour cela le site Europe Topics sur le premier semestre de l'année 2014 [1]. Nous avons utilisé apache [1] pour extraire le texte des fichiers html. Nous avons séparé les données sur 6 mois, comme indiqué sur le tableau 3.2.

1. http ://www.eurotopics.net/fr/home/presseschau/aktuell.html
1. Ti-ka23http ://www.tika.apache.org

TABLE 3.2 – corpus de données

Mois	Totale des documents	Totale des mots
1	254	76300
2	595	105450
3	510	89502
4	300	82404
5	405	81318
6	577	102005

3.5.2 Les métriques d'évaluation de l'approche ETL

Le processus ETL représente une phase critique dans le processus de l'entreposage de données, puisqu'il est responsable de l'alimentation des entrepôts de données. La qualité des données stockées affecte considérablement les tâches d'analyse. C'est pourquoi il est important de mesurer la qualité de notre approche ETL.
L'approche ETL proposée prend un ensemble de documents textuels comme entrée et retourne comme résultat un ensemble de hiérarchies sémantiques représentants ces documents. Les hiérarchies sémantiques obtenues sont chargées dans des entrepôts de textes pour être utilisées dans l'analyse OLAP. Afin d'évaluer l'approche ETL, nous devons évaluer la qualité des hiérarchies sémantiques fournies par notre approche. Dans le domaine de recherche d'informations, il existe de nombreuses mesures d'évaluation. Ces métriques dépendent souvent sur le jugement de pertinence des résultats retournés, parmi les plus connus d'entre eux nous citons la *Prcision* et le *Rappel*, ces mesures tiennent compte du taux de documents pertinents récupérés (Précision) et de la quantité de documents pertinents récupérés (Rappel). Ces deux paramètres peuvent ensuite être combinés pour obtenir une nouvelle métrique appelée F-score qui donne une valeur qualitative de l'approche évaluée. Ces paramètres sont bien adaptés pour évaluer la pertinence des hiérarchies sémantiques fournies par notre approche ETL.

3.5.2.1 L'évaluation du Rappel

Étant donné qu'il n'y a pas d'indice de référence standard pour évaluer la qualité des hiérarchies sémantiques générées, nous avons mené une étude humaine, en essayant d'évaluer le rappel et la précision des hiérarchies sémantiques générées. Pour évaluer le rappel des hiérarchies sémantiques globales extraites pour tous les documents, nous utilisons un rappel moyen. Un rapprochement moyen supérieur signifie que les hiérarchies sémantiques sont plus représentatives des documents textuels. Le rappel moyen (AvR) pour un ensemble de documents est calculé comme suit :

$$AvR = \frac{\sum_{i=1}^{n} Recall_i}{n} \tag{3.3}$$

Tel que n est le nombre totale de documents ,avec pour chaque document $document_i$, le rappel(recall) est calculé comme suit :

$$Recall_i = \frac{relevant\_concepts \cap retrieved\_concepts}{relevant\_concepts} \tag{3.4}$$

Pour évaluer l'approche ETL, nous comparons les résultats du rappel donné de notre approche avec les résultats du rappel donné par deux autres travaux : [104] et [105]. Ces travaux utilisent la même idée que notre approche, qui utilise la méthode LDA avec la taxonomie ODP pour les extractions de concepts. Cependant, pour l'extraction des concepts, [104] utilise seulement les cinq premiers niveaux de taxonomie ODP et [105] utilisent uniquement les trois premiers niveaux de la taxonomie ODP, alors que dans notre approche, nous utilisons tous les niveaux ODP avec une tâche de filtrage. Pour l'évaluation du rappel, nous comparons la qualité des concepts extraits pour chaque approche. Les résultats obtenus sont indiqués dans la figure 3.6, plus de détails sont fournis dans la section discussion.

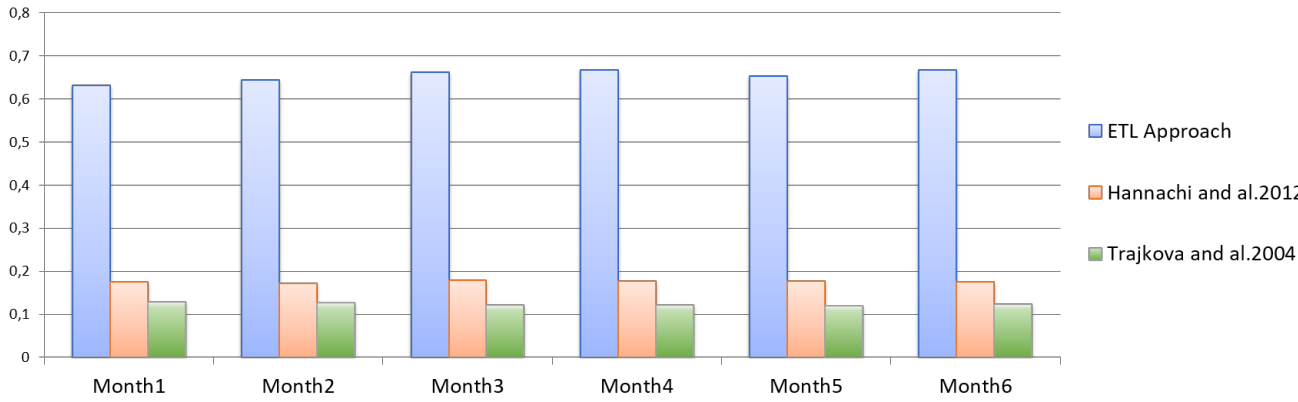

FIGURE 3.6 – Recall evaluation

3.5.2.2 L'évaluation de la précision

Nous pourrions également évaluer la précision des hiérarchies sémantiques extraites en utilisant la même méthodologie que nous avons utilisé pour estimer le rappel. Cependant, notre approche ETL extrait un nombre significatif de concepts que les annotateurs humains n'ont pas identifiés lors du marquage des concepts importants dans les documents. Pourtant, lorsque les annotateurs ont examiné les hiérarchies sémantiques générées, ils pourraient facilement dire si une hiérarchie sémantique particulière représente avec précision le contenu du document textuel. Ainsi, pour estimer la précision, nous avons demandé aux annotateurs d'examiner si les concepts dans les hiérarchies sont utiles. Pour évaluer la précision des hiérarchies sémantiques globales extraites pour tous les documents, nous utilisons une précision moyenne. Une précision moyenne supérieure signifie que les hiérarchies sémantiques sont plus représentatives des documents textuels. La précision moyenne (AvP) pour un ensemble de documents est calculée comme suit :

$$AvP = \frac{\sum_{i=1}^{n} Precision_i}{n} \tag{3.5}$$

Tel que n est le nombre totale de documents, avec pour chaque document texte i, la précision est calculée comme suit :

$$Precision_i = \frac{relevant\_concepts \cap retrieved\_concepts}{retrieved\_concepts} \tag{3.6}$$

Pour évaluer la précision de notre approche, nous comparons nos résultats avec les résultats de précision donnés par les deux autres travaux : [104] et [105]. Pour l'évaluation de la précision, nous comparons la qualité des concepts extraits pour chaque approche. Les résultats obtenus sont indiqués dans la figure 3.7, plus de détails sont fournis dans la section discussion.

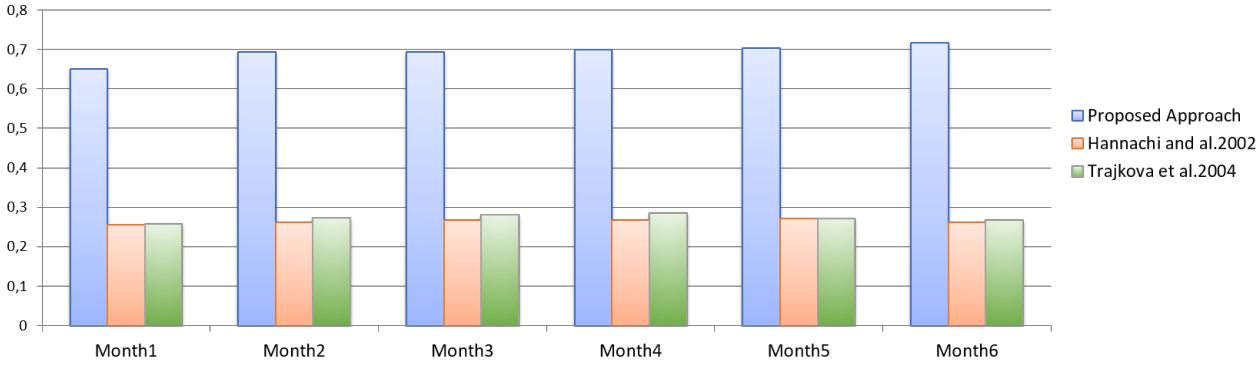

FIGURE 3.7 – L'évaluation de la précision

3.5.2.3 Discussion

Pour évaluer notre approche, nous comparons les résultats donnés par l'approche ETL avec les résultats donnés par les deux travaux : [104] et [105]. La comparaison concerne l'évaluation de la qualité des concepts générés en utilisant les trois approches. Les trois approches utilisent la méthode LDA avec la taxonomie ODP pour générer des concepts ODP, mais la première approche proposée par [105] utilise les trois premiers niveaux de taxonomie ODP. La deuxième approche proposée par [104] utilise les cinq premiers niveaux de la taxonomie ODP, tandis que notre approche ETL utilise tous les niveaux ODP disponibles avec une tâche de filtrage qui supprime les concepts non pertinents de l'ensemble des concepts générés.
La figure 3.6 et la figure 3.7 montrent les résultats de comparaison. L'expérience est menée sur le corpus de données présenté ci-dessus. Les résultats montrent que notre approche ETL améliore considérablement le rappel et la précision. Cette amélioration s'explique par le fait que

notre approche ETL considère plus de niveaux dans la taxonomie ODP, ce qui réduit la perte d'information, et aussi elle comprend également une tâche de filtrage qui élimine la plupart des concepts non pertinents. Afin de combiner nos résultats sur la précision et le rappel, nous considérons maintenant la répartition du $F1 - score$, donné par :

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.7}$$

Les résultats obtenus sont donnés par la figure 3.8. Ils montrent que notre approche ETL donne des scores beaucoup meilleurs que[104] et [105]. En conséquence, nous pouvons conclure que notre approche ETL donne des hiérarchies sémantiques plus pertinentes, et ne provoque pas de perte d'information.

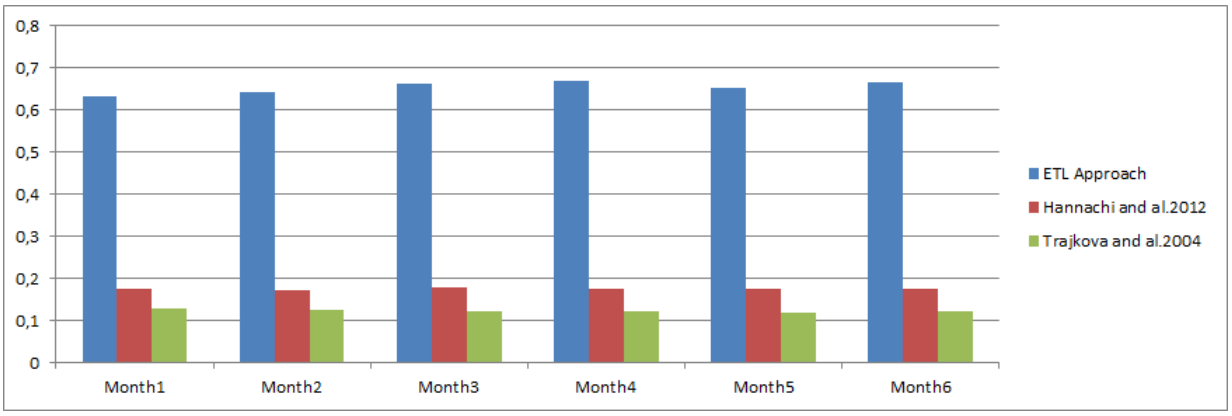

FIGURE 3.8 – Evaluation des score

3.6 Conclusion

L'ETL représente une étape cruciale dans le processus d'entreposage de données. Afin de pallier aux limites des processus ETL classiques inadaptés à l'intégration des données textuelles dans un système décisionnel, nous avons proposé dans ce chapitre une nouvelle approche ETL adaptée aux données textuelles. L'approche proposée utilise les méthodes de fouille de textes pour préparer et intégrer les données textuelles dans un entrepôt de textes. Elle tient compte de la sélection de l'information porteuse de sens avant toute opération de chargement

dans l'entrepôt. En outre, elle s'appuie sur une source externe de connaissances (ODP) pour représenter les connaissances décrivant les documents textuels.

L'approche ETL proposée fournit des hiérarchies de concepts représentant les documents textuels, afin de raffiner cet ensemble à un ensemble plus restreint de concepts pertinents, qui captent avec précision le sujet du document, nous avons proposé d'introduire la pertinence contextuelle dans le calcule des poids des concepts aux sein des hiérarchies obtenues, à travers la technique de rétro-propagation de la pertinence du domaine du document aux hiérarchies de concepts qui le représente.

Pour valider notre approche, nous avons développé un prototype logiciel mettant en œuvre les étapes de notre processus ETL. L'étude expérimentale a montré que l'approche ETL améliore considérablement la qualité de l'information extraites à partir des documents textuel par rapport à d'autres travaux([104] et [105]), en termes de : *prcision*, *rappel* et la combinaison de ces deux paramètres ($F1-1Score$).

Après avoir proposé dans ce chapitre une approche d'alimentation d'entrepôts de textes, nous présentons dans le chapitre suivant un nouveau modèle multidimensionnel, permettant de représenter les données textuelles sous forme multidimensionnelle dans un but d'analyse.

Chapitre 4

Modèle Multidimensionnel Sémantique d'Objet Texte

4.1 Introduction

La modélisation multidimensionnelle consiste à organiser les données de façon que les applications OLAP soient performantes et efficaces. Les modèles d'entrepôts classiques offrent un cadre de travail pour effectuer une modélisation multidimensionnelle des données simples. Un modèle multidimensionnel classique permet d'observer des faits à travers des indicateurs (mesures) et des dimensions [106]. De nombreuse variations de modèles multidimensionnel classiques tel que le modèle en étoile et le modèle en flocon de neige [107] ont donné de bons résultats dans le domaine de l'entreposage des données simple (les mesures représentent des données numériques). Or ces modèles ne sont pas adaptés à l'analyse des données textuelles et présentent un certains nombre de déficit [76] tels que :

- **Non analyse du contenu de documents**
  La modélisation reposant sur les concepts de fait et de dimension associés à des indicateurs numériques permet des analyses simples de documents XML. Ces analyses reposent principalement sur le comptage de documents.
- **Analyses prédéfinies et structures non flexibles**
  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.

L'entreposage des données textuelles demeure encore aujourd'hui une des difficultés majeures et implique de nombreux problèmes, notamment ceux de leur modélisation et leur intégration d'une part et

leur analyse d'autre part. Les entrepôts de textes sont apparus comme une nouvelle solution permettant une analyse multidimensionnelle des données textuelles. La nature complexe de ces données nécessite un traitement bien particulier qui prend en compte leur sémantique. Dans la littérature, des méthodes de recherche d'information et de fouille de données ont donné de très bons résultats pour l'exploration des données textuelles. L'idée clef derrière ces entrepôts de textes est de faire un couplage entre les techniques de fouille de données et de recherche d'information d'un coté, et les techniques OLAP de l'autre coté.

Les modèles d'entrepôt de textes existants ont permis de traiter quelques aspects de complexité liés à l'analyse des données textuelles selon différents niveaux. Toutefois, au regard de nos besoins, ces modèles présentent un certain nombre d'insuffisances tel que la sémantique des données textuelles qui a été prise en compte grâce à l'intégration d'une dimension sémantique dans certains modèles, toutefois elle reste toujours peu exploitée. L'ensemble de ces modèles ne permettent pas une analyse sur de différents niveaux sémantiques. Aussi, les autres aspects comme la prise en compte de la structure des données textuelles ainsi que la flexibilité d'analyse restent toujours limités, bien que le sujet d'analyse *Fait* n'est pas prédéfini au préalable dans quelques modèles tel que dans ([108] et [76]). Nous constatons que le problème de flexibilité est assez complexe. Le contenu sémantique des données textuelles peut être observer comme étant une mesure d'analyse (*K-top keyword,Topic... etc*), comme il peut être considéré comme étant un axe d'analyse (hiérarchie de thèmes), les travaux actuels ne traitent pas ce double rôle.

Notre objectif est de proposer un modèle multidimensionnel pour l'entreposage des données textuelles, qui permet de représenter la sémantique des données textuelles et de l'organiser sous forme hiérarchique pour assurer une analyse sémantique sur différents niveaux de granularité. Dans notre approche de modélisation nous nous intéressons en particulier aux trois aspects suivant : la prise en compte de la structure des données textuelles et de leur sémantique d'une part et la flexibilité d'analyse d'autre part.

La suite de ce chapitre est organisée comme suit : la section 2 présente un exemple de motivation que nous allons utilisé tout au long de notre chapitre pour expliquer nos propositions. Dans la section 3, nous présentons le modèle MSMTO dédié à l'analyse des données textuelles. La section 4 présente le cube textuel sémantique (ST-Cube) et son opérateur de construction. La section 5 définit un nouvel opérateur d'agrégation (Top_KRankedTopics), et la section 6 conclut le chapitre.

4.2 Exemple de motivation

Les revues de presse représentent une importante source de données grâce aux grandes masses d'informations qu'elles véhiculent. L'exploitation de ces informations est devenue essentiel pour répondre aux différents besoins analytiques tel que : l'analyse des principaux sujets d'actualités, l'évaluation de la qualité des articles publiés selon différents critère (ex. domaine des articles, les sujets traités, les avis des lecteurs), l'évaluation de la production des auteurs (selon la fréquence de leurs publication et la qualité de leurs articles),...etc. L'entreposage de ces données qualifiés complexes implique de nombreux difficultés concernant leur modélisation et leur intégration d'une part et leur analyse d'autre part. La représentation de ces données par les modèles d'entrepôts classiques tel que le modèle en étoile permet de faire des analyses simples des articles publiés tandis que l'analyse du contenu n'est pas assurée. Aussi, les modèles actuels ne donnent pas la possibilité d'analyser une section spécifique d'une revue de presse (ex. analyser juste la section sport), car ils ne prennent pas en considération la structure interne des documents textuels. La figure 4.1 nous illustre le diagramme de classe représentant une revue de presse que nous allons utilisé comme exemple pour présenter notre modèle.

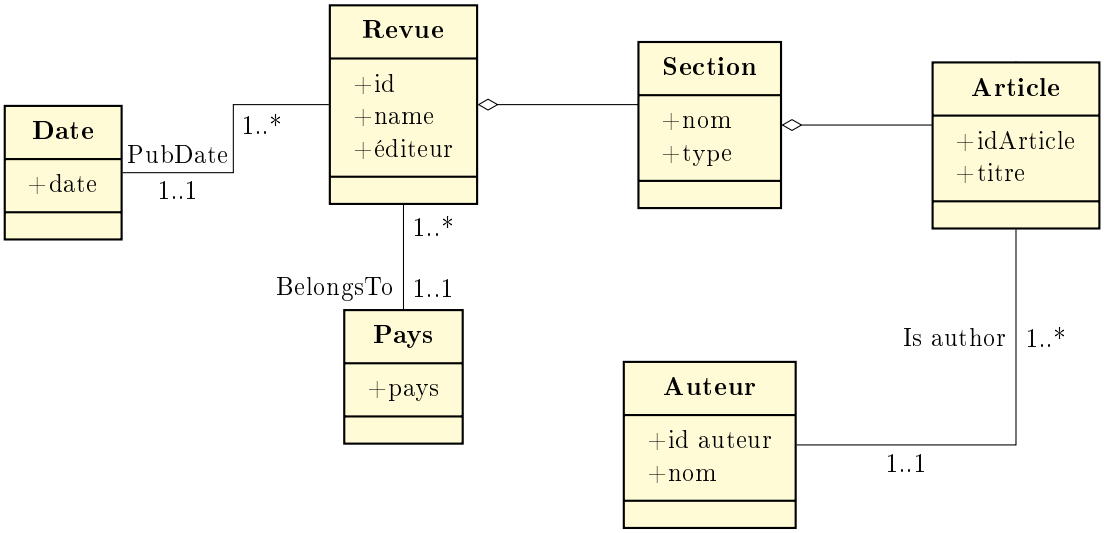

FIGURE 4.1 – Diagramme de classe représentant une revue de presse

4.3 Modèle Multidimensionnel Sémantique d'objet texte(MSMTO)

Dans notre approche de modélisation, nous nous sommes focalisés sur les trois aspects de complexités principales d'analyse de données textuelles : (1) la structure interne des documents textes, (2) la sémantique véhiculée dans les données textuelles et (3) la flexibilité d'analyse. Le MSMTO modèle est basé sur le paradigme objet, un choix justifié par la capacité représentative des modèles orienté objet. Il permet une analyse profonde sur de différents niveaux structurels et sémantiques grâce aux deux types de hiérarchies (hiérarchie structurelle et hiérarchie sémantique), ainsi qu'une bonne flexibilité d'analyse grâce à notre opérateur de construction de cube sémantique textuel. Nous présentons ci-dessous les différents concepts de base de notre modèle.

4.3.1 Définitions des concepts

Nous modélisons notre modèle selon 6 concepts qui sont : l'objet texte, la relation entre objet texte, la hiérarchie inter-objets texte que nous appelons hiérarchie structurelle, l'objet contenu sémantique, la hiérarchie inter-objets contenu sémantique que nous appelons hiérarchie sémantique, ainsi que la relation entre objet texte et objet contenu sémantique.

1. **Objet texte :** un objet texte est une entité représentant un élément texte qu'il est possible d'analyser en tant que sujet ou axe d'observation. Un objet texte est défini par un ensemble d'attributs simples qui sont représentés par des attributs de classe UML. Ces attributs représentent les données textuelles extraits des document textes. Un objet texte est noté TObjt et il est définit comme suit :

$$TObjt = (IDTObjt, SATObjt) \tag{4.1}$$

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

tel que : IDTObjt représente l'identifiant de l'objet texte et SATObjt représente l'ensemble de ses attributs.

2. **Contenu Sémantique (SCObjt)** : un objet contenu sémantique est une entité représentant le contenu sémantique d'un objet texte. Il est obtenu par l'application d'une méthode permettant d'extraire la sémantique véhiculée dans l'élément texte auquel il a été associé. Il est possible d'analyser l'objet contenu sémantique en tant que sujet ou axe d'observation. Il est défini par un ensemble d'attributs simples et une méthode d'agrégation sémantique qui opère sur ce dernier afin de produire d'autres objets contenu sémantique de niveau de granularité plus haut.
Il faut noter que l'objet contenu sémantique du plus bas niveau de granularité est associé au niveau « 0 ». Un objet contenu sémantique est noté SCObjt et il est définit comme suit :

$$SCObjt = (IDSCObjt, SASCObjt, MSCObjt) \tag{4.2}$$

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

tel que $IDSCObjt$ représente l'identifiant de l'objet contenu sémantique, $SASCObjt$ représente l'ensemble de ses attributs et $MSCObjt$ représente une fonction d'agrégation.

La figure 4.2 illustre le méta modèle sémantique d'objet texte décrit par le diagramme de classe UML.

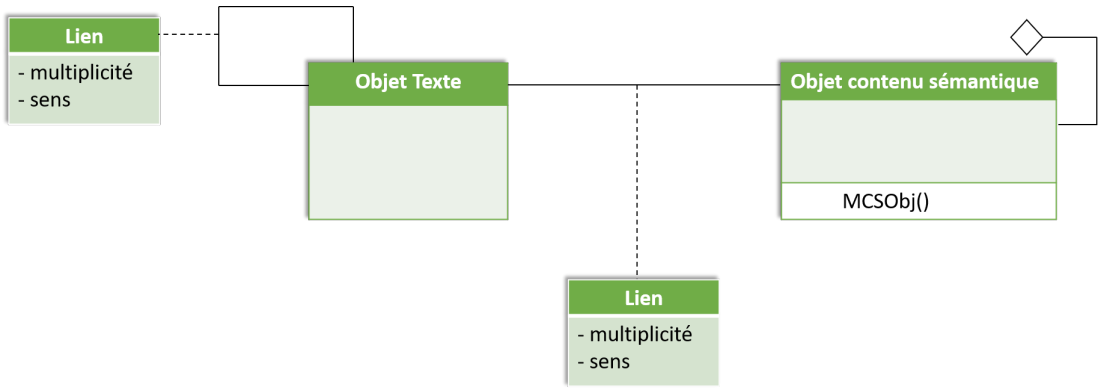

FIGURE 4.2 – Méta Modèle sémantique d'objet texte

Ce modèle est décrit comme suit :

- La classe objet texte représente l'objet texte.
- La classe objet contenu sémantique représente l'objet contenu sémantique.
- Le lien d'association entre objets textes ($Lien(TObjt, TObjt)$) représente les liens entre les objets textes, ex. association, héritage, agrégation, composition.. etc.
- Le lien d'association entre objets contenu sémantique ($Lien(SCObjt, SCObjt)$) représente les liens d'agrégation entre les objets contenu sémantique.
- L'association entre objet texte et objet contenu sémantique représente une relation qui associe à chaque objet texte, un objet contenu sémantique.

Exemple.
Une revue de presse est un exemple d'objet texte. Elle peut être décrite par un ensemble d'attributs tels que : le nom de la revue, le type, l'éditeur.. etc. Aussi une revue peut être décrite par un ensemble d'objets textes tels que : section, sous section, article...etc. Les sujets associés à chaque objet texte (revue, section, sous section, articles) sont des exemples d'objets contenu sémantique. Un exemple de la relation entre objets contenu sémantique est la relation d'agrégation entre l'objet contenu sémantique *sujets* et l'objet contenu sémantique *famille de sujets*. La figure 4.3 nous illustre la

représentation de l'exemple de motivation présenté ci-dessus par notre modèle multidimensionnel sémantique d'objet texte.
Nous notons que dans l'exemple, les objets textes : journal, section ont été liés au même objet contenu sémantique "Sujets", toute fois il est possible d'attribuer à chaque objet texte un objet contenu sémantique spécifique et cela dépend des besoins d'analyse, par exemple l'objet texte "Article" est associé à un objet contenu sémantique différent de celui de sa classe composite qui est «Termes».

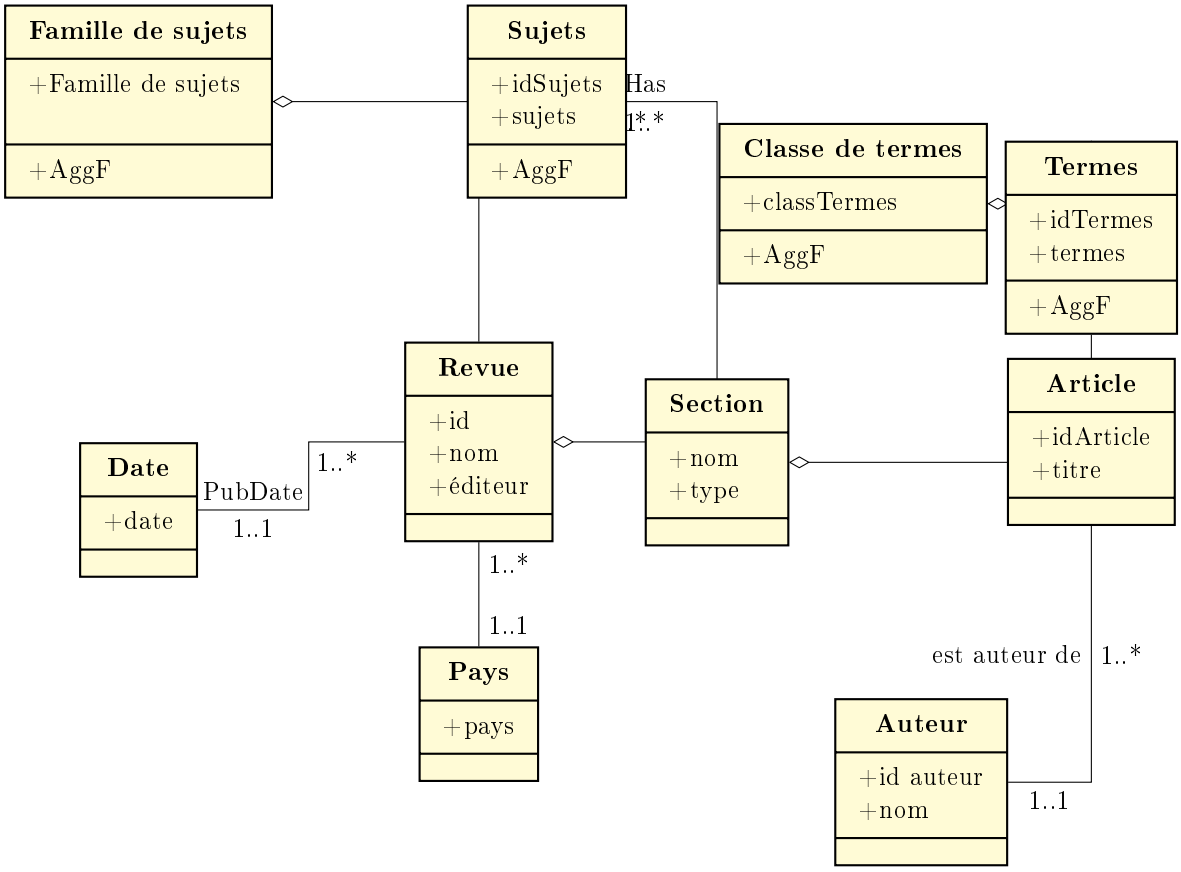

FIGURE 4.3 – Représentation d'une revue de presse par le modèle MSMTO

3. **Relation complexe :** cette relation modélise les liens entre les objets texte hors ceux d'agrégation, de généralisation et de composition d'une part, et les liens entre les objets texte et les objets contenus sémantique d'une autre part. Sa complexité réside dans le fait qu'elle définisse les axes et les mesures d'analyse de certains objets par rapports à d'autres. La figure 4.4 nous illustre un exemple

d'une relation complexe.

Une relation complexe est notée R et elle est définit comme suit :

$$R = (TObjt^R, Object^R) \tag{4.3}$$

$$Tel\ que : \quad Object \in \{TObjt, SCObjt\}$$

$$et \quad R \notin \{Aggregation, composition, generalisation\}$$

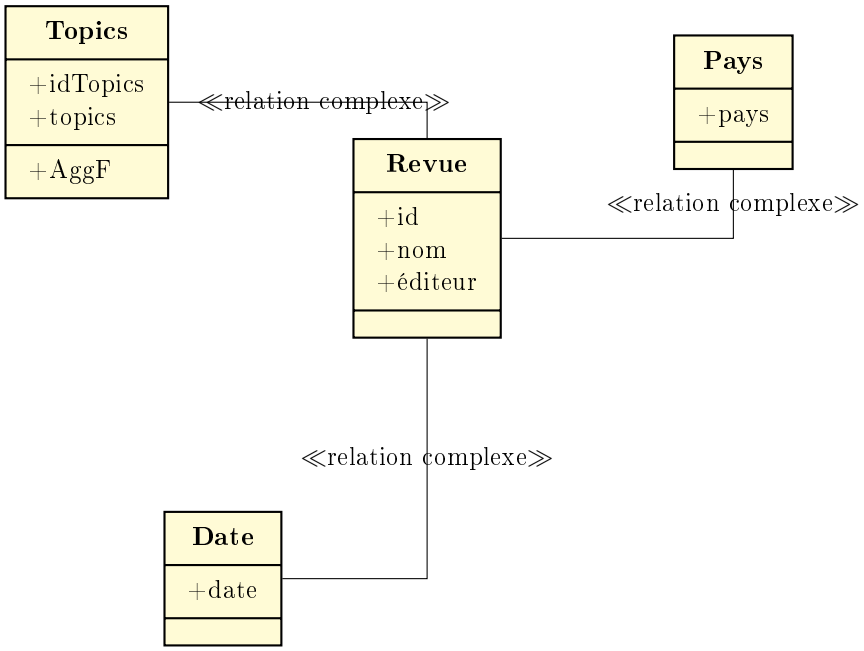

FIGURE 4.4 – Un exemple de relations complexes pour l'objet texte "Revue"

4. **complexe étendue** : elle modélise les liens non directs entre les objets textes, par exemple elle relie un objet texte composants d'un objet texte composite aux objets textes auquel l'objet composite a été relié. Cette relation est très utile pour l'analyse des données textuelles car elle nous permet d'analyser des objets textes qui ne sont pas liés par des liens directes. Si nous prenons l'exemple présenté dans la section précédente, nous remarquons que l'analyse des sujets récents dans la section sport d'une revue par date et pays n'est pas assuré car il n'y a aucun lien directe entre l'objet

section et les objets date et pays. Cette relation définit aussi les axes d'analyse de certains objets par rapports à d'autres.

Une relation complexe étendue est notée ER et elle est définit comme suit :

$$ER = (Tobjt^{ER}_{Source}, TObjt^{ER}_{Cible}) \tag{4.4}$$

$$tel\ que\ :\ \ Relation(Tobjt^{Relation}_{Source}, TObjt^{Relation}_{x})$$

$$et\ \ R(objt^{R}_{x}, TObjt^{R}_{Cible})$$

où : R est une relation complexe (voir (5.3))
$ER \notin \{Aggregation, composition, generalisation\}$
et $Relation \in \{Aggregation, composition, generalisation\}$
Exemple : un exemple d'une relation complexe étendue est la relation entre l'objet texte Article et les objets Date et Pays. La figure 4.5 nous illustre un exemple de relations complexes étendues.

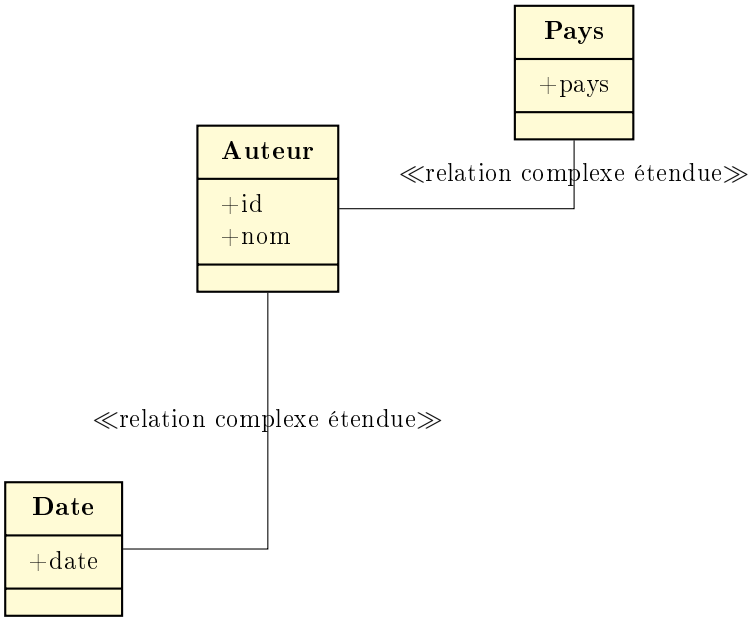

FIGURE 4.5 – Exemple de relations complexes étendues

5. **Hiérarchie structurelle** : une hiérarchie structurelle est définie entre plusieurs objets texte. Elle permet d'effectuer des opérations d'agrégation entre les objets texte selon leur structure. Une hié-

rarchie structurelle définit un ordre partiel entre certains objets texte selon leur degré de granularité. Une hiérarchie structurelle est notée StH et elle est définie comme suit :

$$StH = \{TObjt_1, TObjt_2, ..., TObjt_n / n \in \mathbb{N}\} \cup \{AllObjt\} \quad (4.5)$$

Tel que : AllObjt représente un objet artificiel possédant la plus faible granularité.

Exemple : un exemple d'une hiérarchie structurelle est la relation d'agrégation entre l'objet texte revue et l'objet texte section et la relation d'agrégation entre l'objet texte section et l'objet texte article (figure 4.6)

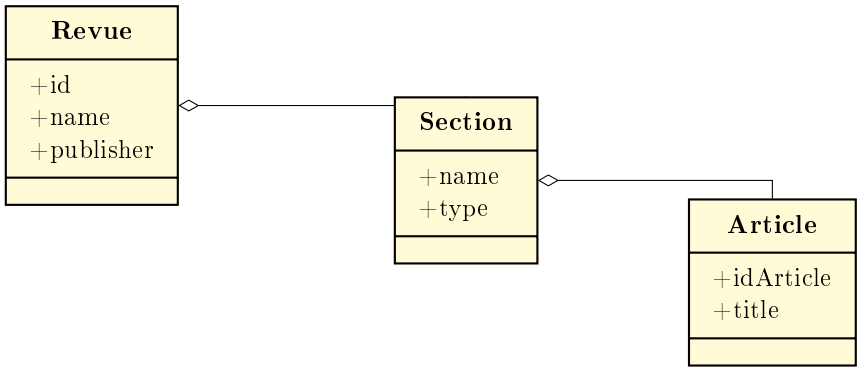

FIGURE 4.6 – Exemple d'une hiérarchie structurelle

6. **Hiérarchie sémantique** : une hiérarchie sémantique est définit entre plusieurs objets contenu sémantique, c'est un type particulier de hiérarchie qui permet d'établir des agrégations sémantiques entre les objets contenu sémantique. Nous définissons une fonction $SCObjtLevel(SCObjt, SH)$ qui retourne le niveau de chaque objet contenu sémantique au sein de la hiérarchie sémantique. Nous supposons que l'objet le plus détaillé de la hiérarchie possède le niveau 0. L'objet le plus détaillé de la hiérarchie sémantique est associé à un objet texte.

   Une hiérarchie sémantique est notée SH et est définie comme suit :

$$SH = \{SCObjt_1, SCObjt_2, ..., SCObjt_n / n \in \mathbb{N}\} \quad (4.6)$$

$Tel\ que :\ R(TObjt^R, SCObject^R) \Longrightarrow SCObjtLevel(SCObjt, SH) = 0$

et AllObjt est un objet artificiel possédant la plus faible granularité.

Exemple : la relation d'agrégation entre l'objet contenu sémantique « Sujets » et l'objet contenu sémantique « Famille de sujets», représente un exemple de hiérarchie sémantique. L'objet Sujets représente l'objet le plus détaillé de la hiérarchie, il est associé à l'objet texte « Revue » (Figure 4.7).

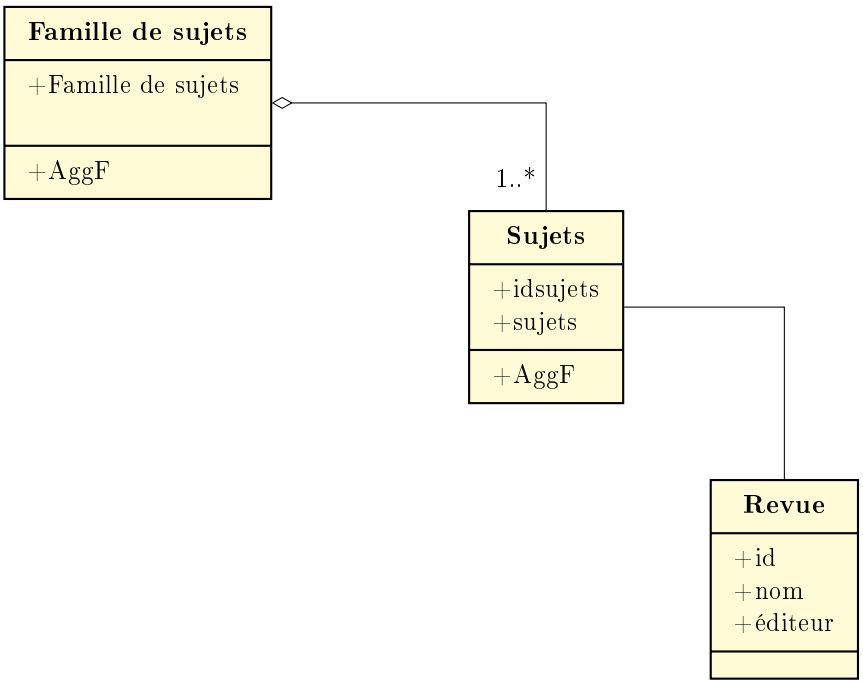

FIGURE 4.7 – Exemple de hiérarchie sémantique

4.4 Le Cube Textuel Sémantique(ST-Cube)

Afin d'effectuer une analyse multidimensionnelle dans notre modèle, nous définissons un cube textuel sémantique comme suit :

Un cube textuel sémantique est construit à base du modèle multidimensionnel sémantique d'objets textes, il peut modéliser un sujet d'analyse nommé "FactObject" défini par plusieurs objets

dimensions textes $TODim_i, i \in [1..*]$ et objets dimension sémantiques $SODim_j, j \in [1..*]$. Un objet dimension texte inclut des attributs textes $TA =< ta_1, ta_2, .., ta_* >$ et il est associé à une hiérarchie structurelle composée d'objets textes $StH =< to_1, to_2, .., to_* >$. Un objet dimension sémantique inclut des attributs sémantique $SA =< sa_1, sa_2, .., sa_* >$ et il est associé à une hiérarchie sémantique composée d'objets contenu sémantique $SO =< so_1, so_2, .., so_* >$.
Un cube textuel sémantique est défini par un ensemble d'objet dimensions textes, objets dimension sémantiques , objet fait et un ensemble de mesure. Le Drill down et le roll up le long des hiérarchies structurelles permet de faire des analyses sur de différents niveaux hiérarchiques. Le Drill down et le roll up le long des hiérarchies sémantiques permet de faire une analyse sémantique sur de différents niveaux de granularité. Notre cube textuel sémantique peut stocker de différents types de mesure : mesure numérique, mesure textuel, mesure sémantique...etc.

4.5 Text OLAP

L'avantage d'un entrepôt de données est de fournir aux décideurs une vue analytique des données. Les systèmes OLAP permettent de naviguer à travers des cubes multidimensionnels d'une vue à l'autre de manière interactive. Les opérateurs OLAP peuvent être regroupés en fonction de leur utilisation. Nous considérons d'abord les opérateurs de construction de cubes qui créent des cubes à la volée à partir des sources de données ou de l'entrepôt de données. Deuxièmement, nous considérons les opérateurs de présentation, qui présentent les cubes de données aux analystes. Cette famille s'intéresse aux opération qui modifient les valeurs des mesures (agrège) ([78]). Le *slice*, *Dice* et la rotation sont les opérateurs OLAP les plus fréquemment référencés, ils permettent d'exprimer des requêtes complexes et de voir des résultats agrégés pertinents pour le décideur. En pratique, les opérateurs OLAP s'appliquent à des cubes de données bien structurés où les paramètres de dimen-

sion sont catégoriels et les mesures sont exclusivement numériques. Dans ce cas, les opérations d'agrégation utilisent les fonctions de regroupement SQL habituelles, telles que SUM,AVG,... etc.
Cependant, avec l'expansion de la technologie des entrepôts de données sur de nouveaux domaines, caractérisés par la complexité des données qu'ils génèrent et manipulent, il est devenu nécessaire de créer de nouveaux concepts et techniques de modélisation multidimensionnelle.
Par conséquent, le traitement analytique en ligne doit également être adapté pour faire face aux différents types de données.
Dans cette section, nous présentons deux opérateur OLAP : l'opérateur de construction du cube sémantique et l'opérateur d'agrégation $Top\_KRankedTopics$.

4.5.1 Opérateur de construction du cube textuel sémantique

Le modèle proposé dans la section précédente offre une bonne flexibilité d'analyse, non seulement par donner aux décideurs la possibilité de définir le fait lors de l'analyse, mais aussi par donner la possibilité de considérer le contenu sémantique des données textuelle comme : un axe d'analyse, un fait ou même une mesure pour un objet fait. Le processus de construction du cube textuel sémantique consiste à : (i) sélectionner un objet à partir du modèle multidimensionnel sémantique d'objet texte et de lui attribuer le rôle de sujet d'analyse, cet objet sera appelé objet-fait. (ii) Sélectionner un ensemble d'objets du modèle multidimensionnel sémantique d'objet texte et de leurs attribuer le rôle d'axes d'analyse. Ces objets seront appelés objets-dimension.
L'objet fait peut être : un objet texte ou un objet contenu sémantique ce qui nous donne deux cas :

(a) **Premier Cas** : l'objet fait est un objet texte : dans ce cas nous percevons deux possibilités :
   - La mesure est un attribut simple de l'objet fait. Dans ce cas l'objet contenu sémantique lié à l'objet texte en question par une relation complexe sera pris comme étant un axe

d'analyse.

- La mesure est un objet contenu sémantique : dans ce cas la relation complexe entre l'objet texte et l'objet contenu sémantique en question ne sera pas prise en charge lors de la sélection des objets dimensions.

(b) **Deuxième Cas** : l'objet fait est un objet contenu sémantique. Dans ce cas la mesure serait un attribut simple de l'objet contenu sémantique.

Nous avons choisis d'utiliser les diagrammes de paquetage UML pour la représentation de notre cube textuel sémantique, où :

- Un paquetage représentant l'objet fait contient l'objet jouant le rôle d'objet fait ainsi que l'ensemble de ses mesures.
- Un paquetage représentant l'objet dimension contient l'objet jouant le rôle d'objet dimension.
- Les relations entre les paquetages sont nommées « relation complexe » et « relation complexe étendue »

Exemple : la bonne flexibilité offerte par notre modèle nous donne la possibilité de répondre à plusieurs type de requêtes d'analyses, nous prenons par exemple la requête suivante : analyser les principaux sujets des sections d'une revue de presse pour : $Pays = \{"Algerie"\}$ et $Annee = \{"2016","2015"\}$.
La figure 4.8 présente le schéma multidimensionnel qui répond à cette requête. Ce schéma est obtenu par l'application de l'opérateur de construction du cube textuel sémantique en sélectionnant l'objet texte *section* comme objet fait, l'objet contenu sémantique *sujet* comme mesure d'analyse, l'objet texte *date* comme objet dimension et l'objet texte *pays* comme objet dimension. La figure 4.9 présente l'objet fait *Section* en détail et la figure la table 4.1 présente un exemple d'une cellule du ST-Cube qui répond à la requête d'analyse précédente.

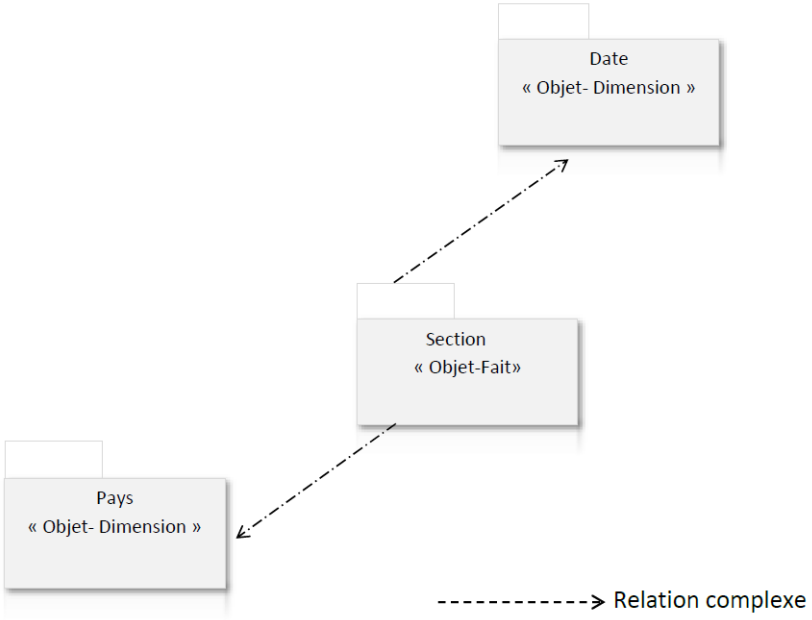

FIGURE 4.8 – Un exemple d'un schéma multidimensionnel pour l'objet fait "Section"

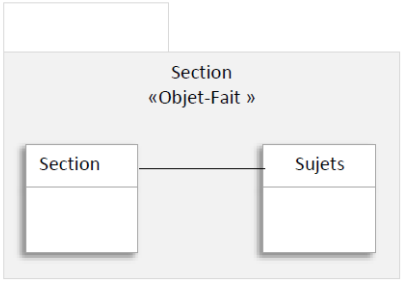

FIGURE 4.9 – Un package détaillé représentant l'objet fait "section"

Section (topics)		Année	
		2015	2016
Pays	Algérie	Football , sécurité, Economie , corruption	jeux olympiques , crise économique, cyclisme, immigration

TABLE 4.1 – Un exemple d'une cellule dans le ST-Cube

Le contenu sémantique peut être considéré selon le modèle MSMTO comme une mesure d'analyse aussi bien qu'un axe d'analyse. En

effet, notre système peut répondre d'autres requêtes, dans lesquelles les sujets sont considérés comme axe d'analyse, tel que l'analyse des articles pour : $Pays = \{"Algerie"\}$ et $Annee = \{"2016","2015"\}$ et $sujets = \{"Sport","Politics"\}$.

4.5.1.1 L'algorithme de l'opérateur de construction du ST-Cube

L'algorithme prend en entrée une liste d'objets textes et d'objets contenu sémantiques qu'on nommera $Object_{List} = \{Obj_1, Obj_2, \ldots, Obj_n\}$. Il produit en sortie un objet fait $FactObject$, un ensemble d'objets dimension $DimensionSet$ et un ensemble $HierarchyDimensionSet$ noté $HDimensionSet$. L'algorithme emploie les fonctions et procédures suivantes :

- **GetHDimension (DimensionObject)** : est une fonction qui prend en entrée un objet dimension et retourne l'ensemble des objets qui ont un lien direct avec lui tel que : $Relation(Object^{Relation}; DimensionObject^{Relation})$ où Relation exprime une relation de composition(respectivement agrégation) de l'objet composant(resp agrégat) $Objet$ vers l'objet composite (resp agrégé) $DimensionObject$ (voir 4.4).
- **GetDimensionSet(FactObject)** : est une fonction qui prend en entrée un objet $FactObject$ et retourne l'ensemble des objets reliés à cet objet la par le biais d'une relation complexe définit dans 4.3 et d'une relation complexe étendue définit dans 4.4.
- **SetAgregationFunction(Function)** : est une fonction qui attribut à chaque mesure une fonction d'agrégation $function$.
- **Select (mesure, AttributeSet)** : est une fonction qui attribut à la mesure une valeur de l'ensemble $AttributeSet$.

Nous présentons ci-dessous l'algorithme de construction du cube textuel sémantique :

Algorithme 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
```

4.5.2 Opérateur d'agrégation

L'agrégation des données est l'une des tâches d'analyse les plus importantes. La portée des fonctions agrégation se réduit à un ensemble restreint d'opérations, tel que le comptage, lorsque les mesures d'analyse ne sont plus numériques. La figure 4.10 montre un exemple d'analyse, où le décideur analyse le nombre d'articles publiés chaque mois par les auteurs. Afin d'obtenir une vue plus globale sur les données, il change d'affichage par années (rolls-up). Par conséquent, les valeurs mensuelles sont agrégées en une valeur pour chaque année. Pour traiter l'analyse OLAP des données textuelles, plusieurs travaux proposent des opérateurs d'agrégations tel que [109] où les auteurs proposent un opérateur nommé TOP-KW. Un autre opérateur d'agrégation appelé adaptatif $Tf - Idf$ a été développé dans [110]. Cependant, la plupart de ces travaux dépendent du $Tf - Idf$ comme mesure. Il s'agit d'une information structurelle et ne prend pas vraiment compte du texte sémantique dans l'analyse OLAP de texte. Par exemple, dans le cas des articles de presse, il est plus intéressant d'extraire des informations sur les sujets pertinents dans le document plutôt que d'extraire les termes les plus fréquents. Le modèle $ST - Cube$ présenté précédemment permet de modéliser des données textuelles dans un environnement

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

```
Input : Object_List={Obj_1,Obj_2,...,Obj_n}
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
```

OLAP. Par conséquent, pour permettre l'analyse OLAP sur notre modèle, nous proposons l'opérateur Top_KRankedTopics.

4.5.2.1 Top_KRankedTopics

Pour assurer une analyse multidimensionnelle des documents textuels, nous proposons un nouvel opérateur d'agrégation : le Top_KRankedTopics. Ce dernier représente un outil d'analyse non conventionnel qui prétend une extension de l'analyse en ligne des

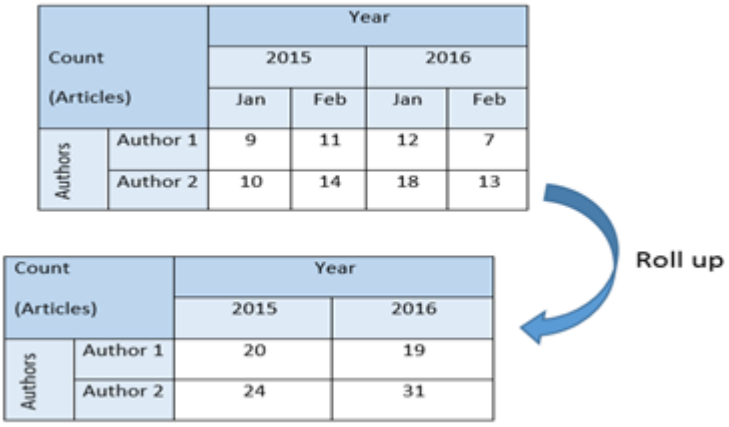

FIGURE 4.10 – Analyse multidimensionnelle d'articles affiché par : auteurs, mois et déployés jusqu'aux années

données complexes. L'opérateur proposé est particulièrement dédié à traiter la nature complexe des données textuelles. Il dépasse les indicateurs numériques pour permettre au puissant cadre OLAP de fonctionner sur ce type de données complexe.
Notre opérateur d'agrégation Top_KRankedTopics agrège un ensemble de document par : (1) sélectionner les K sujets les plus pertinent (ayant le plus grand poids), (2) pondérer ces documents selon la hiérarchie sémantique qui les représente(obtenue en appliquant l'approche décrit dans le chapitre 3). Notre opérateur d'agrégation sélectionne les k premiers sujets et retourne pour chaque sujet une liste de documents pondérés.

- **Spécification formelle :**
  L'opérateur Top_KRankedTopics agrège un ensemble de n sujets dans un sous-ensemble de k sujets les plus pertinents qui correspondent aux sujets avec le poids le plus élevé, puis génère pour chaque sujet une liste de documents classés qui couvre ces sujets.

Exemple. Considérons une requête q_1, dans laquelle le décideur veut analyser les deux sujets les plus pertinents dans les revues de presse pour date={"2014" ;"2013"} ;Location="{Syria" ;"France"}. les résultats présentés dans la figure 4.11

Definition : Le *Top_KRankedTopics* operator est défni comme suit:

$$Top\_KRankedTopics: \quad Topics^n \quad \rightarrow \quad Topics^k$$

$$\left(topic_1(d_1,d_2,...,d_{m1}) < topic_2(d_1,d_2,...,d_{m2}) < .... < topic_n(d_1,d_2,...,d_{m....})\right) \rightarrow \left(topic_1(d_1 < d_2 < ... < d_{m1}) < topic_2(d_1 < d_2 < ... < d_{m2}) < .... < topic_n(d_1 < d_2 < ... < d_{m....})\right)$$

sont obtenus par l'application de notre opérateur d'agrégation.

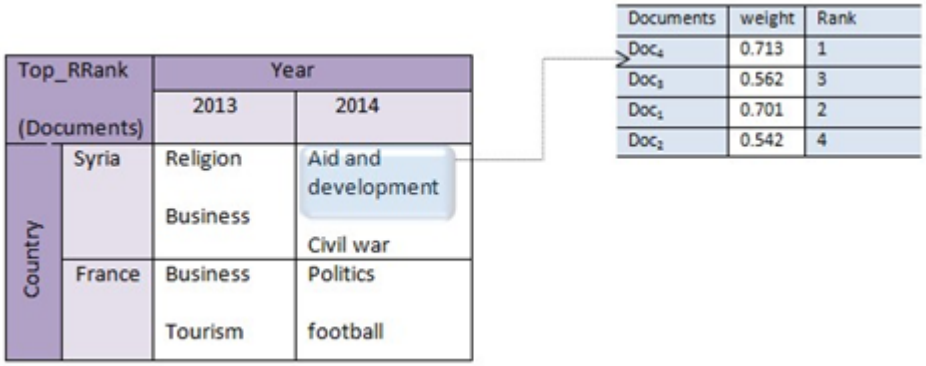

FIGURE 4.11 – Exemple d'analyse utilisant Top_KRankedTopics

4.6 Implémentation et expérimentations

Dans cette section, nous montrons la faisabilité de nos propositions, à savoir le modèle multidimensionnelle sémantique d'objets textes, le cube textuel sémantique et les opérateurs OLAP.

4.6.1 Implémentation

Dans cette section, nous présentons les technologies de base que nous avons utilisé afin de fournir un environnement d'analyse en ligne de données issue de documents textuels. Cet environnement représente une plateforme logicielle d'entreposage de données textuelles regroupant nos différentes contributions théoriques. Cette plateforme permet la construction d'entrepôts de textes et leur alimentation ainsi que leur analyse. Elle servira à évaluer et à valider

l'efficacité de nos propositions par le biais d'un travail d'expérimentation, tout en assurant une validation fonctionnelle de notre approche.
L'évaluation de nos propositions sera achevé dans différentes phases, ça consiste à évaluer :

- La possibilité de construire un entrepôt de texte en utilisant le modèle du $MSMTO$.
- La possibilité de construire un cube de texte en utilisant le modèle du $ST-Cube$.
- L'analyse des données textuelles en utilisant notre opérateur Top_KRankedTopics.

L'architecture de la plateforme développé est de type Client/Serveur. Elle est composée de trois modules complémentaires :

- **Le module ETL :** ce module assure : (1) l'alimentation mécanique de l'entrepôt (l'extraction des méta-données et l'alimentation des objets de types texte). (2) L'alimentation sémantique de l'entrepôt (l'extraction des hiérarchies sémantiques représentant les documents textuels).
- **Le module de spécification du ST-Cube :** il implémente l'opérateur de construction du cube textuel sémantique.
- **Le module de spécification OLAP :** ce module implémente l'opérateur Top_KRankedTopics.

La figure 4.12 illustre l'architecture générale de notre plateforme.

Pour le stockage de l'entrepôt de textes et du cube textuel sémantique, nous avons utilisé Postgresql qui est un système de gestion de base de données orienté objet. Les trois modules composant notre plateforme ont été développé en java sous l'environnement Eclipse.

Implémentation de l'opérateur de construction du ST-Cube

Afin d'implémenter notre opérateur de construction de cube textuel

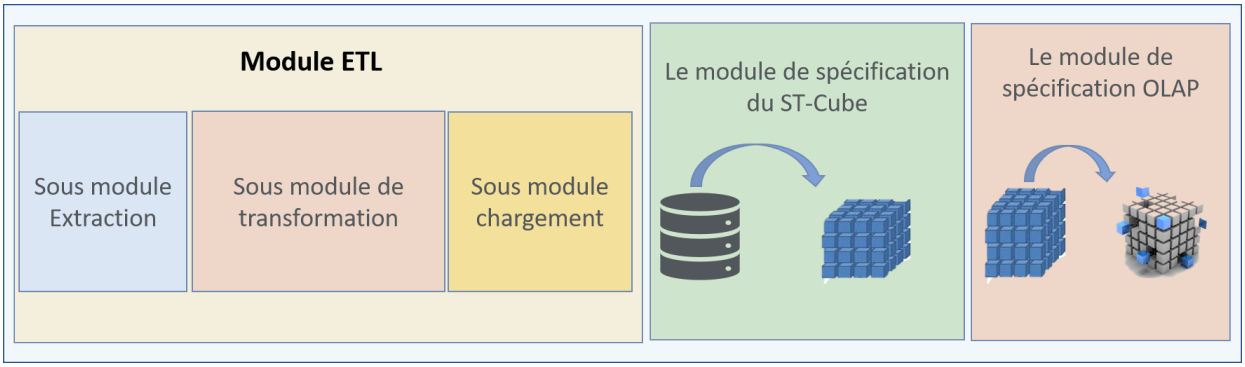

FIGURE 4.12 – Architecture générale

sémantique, nous avons utilisé la technologie java sous l'environnement Eclipse. Donc, nous avons développé une application java qui consulte l'entrepôt de textes au sein de postgresql et produit un cube textuel sémantique. La figure 4.13 illustre le processus de construction de cube textuel sémantique.

4.6.2 Évaluation de l'opérateur Top_KRankedTopics

L'opérateur proposé révèle une nouvelle façon d'effectuer des analyses en ligne sur des cubes de textes, qui prend en charge la sémantique. Dans ce qui suit, nous présentons le mode de fonctionnement de notre opérateur dans deux cas : (1) sans pertinence contextuelle, (2) avec une pertinence contextuelle. Pour illustrer le mode de fonctionnement de l'opérateur Top_KRankedTopics, nous avons effectué un test sur un cube de données basé sur le modèle ST-Cube. Ainsi, nous avons demandé à un décideur de soumettre des requêtes d'analyse. Nous considérons une requête où le décideur veut analyser les deux sujets pertinents des articles de presse pour : $Time = \{"2014"\}$, $Pays = \{"Algrie"\}$. Pour répondre à cette requête, notre opérateur procède comme suit :

- **Mode d'opération du Top_KRankedTopics sans considérer la pertinence contextuelle :**
  (a) L'opérateur Top_KRankedTopics procède au parcours du ST-Cube.

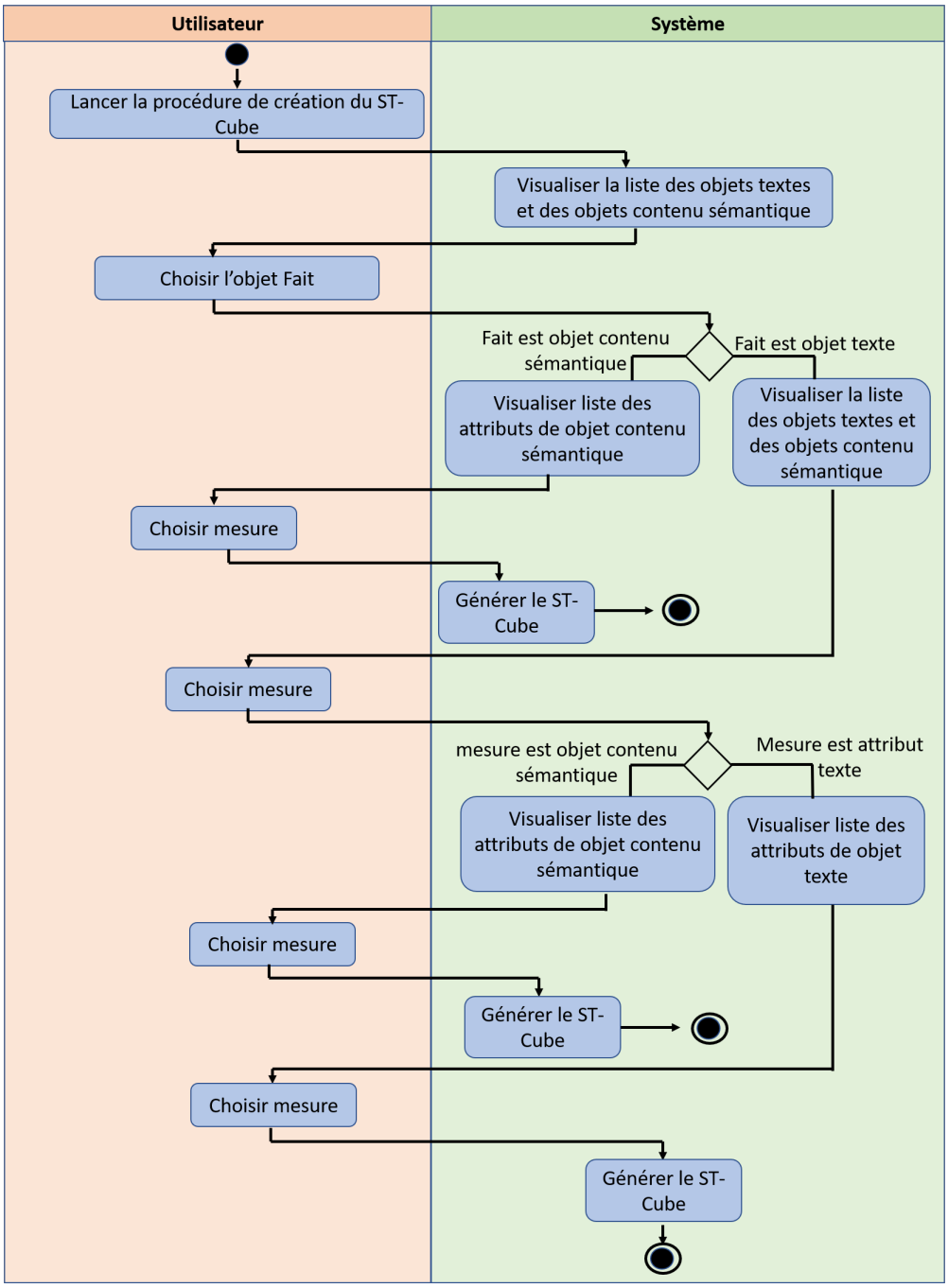

FIGURE 4.13 – Construction d'un ST-Cube

(b) Il sélectionne les deux premiers sujets ayant les deux poids les plus élevés pour country= «Algeria » et Time = « 2014 », puis il retourne la liste de documents qui correspond a chaque sujet.

(c) Ensuite, en se basant sur la formule présenté dans (5.1), il procède au classement des documents, après avoir affecter un poids à chaque concept.

(d) Finalement, il revoie comme réponse : les deux sujets avec les poids les plus élevés et pour chaque sujet une liste classée de documents qui correspond à ce dernier.

- **Mode d'opération du $Top_K RankedTopics$ en considérant la pertinence contextuelle :**

(a) L'opérateur Top_KRankedTopics procède au parcours du ST-Cube.

(b) Il sélectionne les deux premiers sujets ayant les deux poids les plus élevés pour country= «Algeria » et Time = « 2014 », puis il retourne la liste de documents qui correspond a chaque sujet.

(c) Ensuite, en se basant sur la formule présenté dans (5.1), il procède au classement des documents, après avoir affecter un poids à chaque concept.

(d) Utilisez la formule présentée dans (5.2) pour recalculer les poids des concepts, afin de procéder au classement des documents retournés.

(e) Finalement, il revoie comme réponse : les deux sujets avec les poids les plus élevés et pour chaque sujet une liste classée de documents qui correspond à ce dernier.

Les résultats obtenus suite à l'exécution de la requête précédente sont présentés dans la figure 4.14. Cette dernière montre que l'introduction de la pertinence contextuelle pour recalculer les poids de chaque concept a changé le rang de certains documents dans la liste. En outre, un document a été carrément retiré (ex : doc 32), alors que d'autres documents ont été introduits (ex : doc 55).
Les expérimentations conduites offre une validation fonctionnelle de notre opérateur. Toutefois, pour évaluer la performance de l'opérateur Top_KRankedTopics, nous devons évaluer de plus près les résultats qu'il a pu fournir. Dans le chapitre précédente, nous avons considéré trois métriques d'évaluation utilisés dans le domaine de la

recherche d'information(précision, rappel et F-score). Néanmoins, ces mesures sont principalement basées sur des ensembles (set based). elles sont calculées en utilisant des ensembles de documents non classés. Dans un contexte de recherche classé, des ensemble appropriés de documents retourné sont naturellement donnés par les k meilleurs documents retournés. Par conséquent, nous proposons d'utiliser la métrique $Precision@k$ [111]. Cette dernière a été utilisée pour évaluer de nombreux systèmes, et a également été utilisée pour évaluer le rendement de certains opérateurs OLAP tels que l'opérateur ORank [112]. La métrique $Precision@k$ représente le rapport entre le nombre de documents pertinents dans les premiers k documents récupérés et k. En outre, nous proposons une comparaison entre : la performance des Top_KRankedTopics sans pertinence contextuelle et le Top_KRankedTopics avec une pertinence contextuelle. Ainsi, nous devons évaluer les résultats retournés pour savoir qui sont les plus pertinents pour les décideurs. Pour calculer la métrique d'évaluation $Precision@k$, nous avons mené une étude humaine en incorporant des jugements subjectifs de 20 utilisateurs. Nous avons examiné pour chaque utilisateur trois requêtes expérimentales. Par conséquent, un total de 60 requêtes sont sélectionnées en tant que requêtes expérimentales. Nous calculons la $Precision@k$ en utilisant la formule suivante :

$$Precision@k(q) = \frac{Relevent\_Returned\_document^q}{k} \tag{4.7}$$

Tel que : $Relevent\_Returned\_document^q$ est l'ensemble des document pertinents retournés par le $Top_KRankedTopic$s pour une requête q. Nous avons calculé la $prcision@k$ de l'opérateur proposé pour toutes les 60 requêtes expérimentales. Nous donnons les valeurs 5, 10 et 20 à k pour calculer les valeurs de$Precision@5$, $Precision@10$ et $Precision@20$. La figure 4.15 montre une comparaison entre $Precision@5$,$Precision@10$ et $Precision@20$ dans les deux cas donnés ci-dessus.

Dans les deux cas, nous notons que la moyenne de la précision

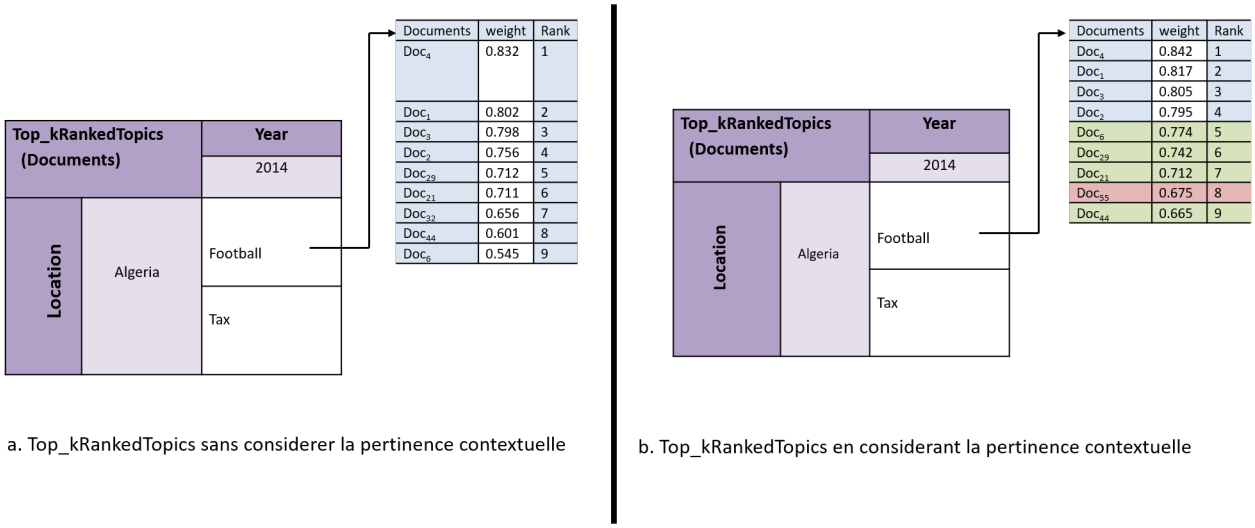

FIGURE 4.14 – Evaluation du Top_KRankedTopics

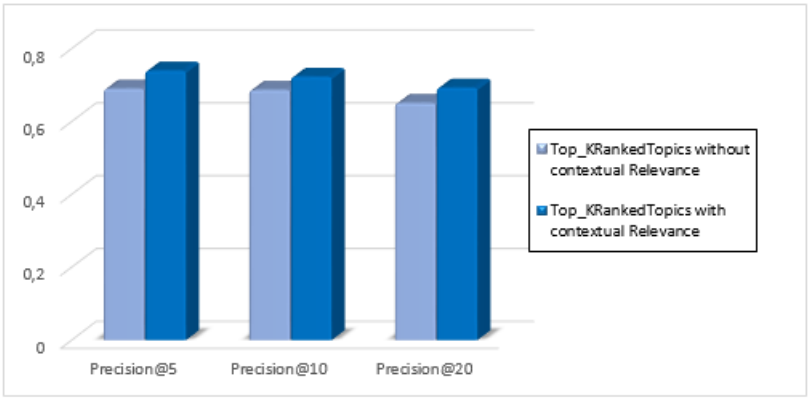

FIGURE 4.15 – Comparaison de la *Precision@k*

de notre opérateur d'agrégations est assez bonne. Cela prouve que notre approche prend en compte la sémantique des données. En outre, la moyenne de précision des Top_KRankedTopics en considérant la pertinence contextuelle est meilleure que la moyenne de précision des *Top_KRankedTopics* sans considérer la pertinence contextuelle. Cela prouve que l'inclusion de la pertinence contextuelle dans le texte OLAP a amélioré la précision et donc les résultats.

4.7 Conclusion

La modélisation des données textuelles dans un but d'analyse implique de nombreux problèmes, notamment en ce qui concerne leur structure et la sémantique qu'ils véhiculent d'une part et la flexibilité d'analyse d'autre part. Les modèles d'entrepôts actuels se trouvent incapables de répondre aux différents aspects de complexité liés à l'analyse de ce type de données. Pour répondre à cette problématique nous avons présenté dans ce chapitre trois contributions. La première aborde la modélisation multidimensionnelle à travers la proposition d'un nouveau modèle multidimensionnelle adaptés aux données textuelles baptisé MSMTO. Notre modèle basé sur le paradigme objet répond à des exigences de modélisation que les modèles existants ne satisfont pas, comme la prise en compte de la structure et de la sémantique des données textuelles à travers les deux types de hiérarchies (d'objets texte et d'objets contenu sémantique), ainsi que la flexibilité d'analyse grâce aux deux relations (relation complexe et relation complexe étendue). Dans notre deuxième contribution, nous avons proposé un opérateur de construction de cube de texte sémantique, qui permet de projeter notre modèle orienté objet sous forme de modèle multidimensionnel pour procéder à l'analyse en ligne (OLAP). Notre troisième contribution a porté sur l'analyse en ligne (OLAP) à travers la proposition d'un opérateur d'agrégation *Top_KRankedTopics*, qui dépasse les indicateurs numériques pour permettre au puissant cadre OLAP de fonctionner sur les données textuelles.
Pour la validation de nos propositions, nous avons développé et mis en place une plate-forme pour le stockage et l'analyse des données textuelles. L'étude expérimentale montre que l'opérateur *Top_KRankedTopic* permet d'effectuer des analyses en ligne efficaces sur les données textuelles tout en tenant compte des différents facteurs sémantiques. En outre, nous avons procédé à une évaluation sur les articles de presse, ce qui montre que, la prise en compte de la pertinence contextuelle, améliore considérablement la précision des résultats obtenus par notre opérateur..

CONCLUSION

Le document électronique représente aujourd'hui un vecteur et un support d'information que les organisations ne doivent pas négliger. En effet, il est entendu aujourd'hui que plus de 80% des données nécessaires au fonctionnement d'une organisation sont encapsulées dans des documents, et non uniquement dans les bases de données opérationnelles. Ces données textuelles restent hors portés des systèmes décisionnels classiques, ce qui induit à une importante perte d'information. Pour répondre à cette problématique et afin de pouvoir prendre profit des informations contenues dans ces documents, il est devenu plus que nécessaire d'intégrer ces données complexes dans des systèmes décisionnels permettant leurs analyses.

Les travaux de recherche présentés dans cette thèse s'inscrivent dans le cadre des systèmes d'aide à la prise de décision et plus précisément dans la construction des entrepôts de textes. Nous avons apporté, à travers ces travaux, des solutions à la double problématique de l'intégration et de la modélisation multidimensionnelle des données textuelles.

Notre première contribution a porté sur la problématique d'alimentation d'entrepôts de textes. Nous avons proposé une approche ETL dédiée à l'alimentation des entrepôt de textes [5]. L'approche proposée utilise les méthodes de fouille de textes pour préparer et intégrer les données textuelles dans un entrepôt de textes. Elle tient compte de la sélection de l'information porteuse de sens avant toute opération de chargement dans l'entrepôt. En outre, elle s'appuie sur une source externe de connaissances (ODP) pour représenter les connaissances décrivant les documents textuels. Ainsi, nous assurons que l'intégration des données textuelles dans l'entrepôt de textes porte également sur la sémantique de ces dernières.
L'approche ETL proposée fournit des hiérarchies de concepts

représentant les documents textuels, afin de raffiner cet ensemble à un ensemble plus restreint de concepts pertinents, qui captent avec précision le sujet du document, nous avons proposé d'introduire la pertinence contextuelle dans le calcule des poids des concepts aux sein des hiérarchies obtenues, à travers la technique de rétro-propagation de la pertinence du domaine du document aux hiérarchies de concepts qui le représente.

Notre deuxième famille de contributions concerne la structuration et la modélisation des entrepôts de textes à travers la proposition du Modèle multidimensionnel sémantique d'objet texte(MSMTO), un modèle dédier à l'analyse multidimensionnelle des données textuelles. Le MSMTO est un modèle basé sur le paradigme objet qui répond à des exigences de modélisation que les modèles existants ne satisfont pas ou alors partiellement, comme la prise en compte de la structure et de la sémantique des données textuelles à travers les deux types de hiérarchies (d'objets texte et d'objets contenu-sémantique), ainsi que la flexibilité d'analyse grâce aux deux relations (relation complexe et relation complexe étendue) auxquelles nous avons basé notre opérateur qui permet de projeter notre modèle orienté objet sous forme de modèle multidimensionnel pour procéder à l'analyse en ligne (OLAP).

Afin d'assurer des analyses multidimensionnelles sur le modèle MSMTO, nous avons défini un modèle de cube et un opérateur de construction de cube textuel sémantique (ST-Cube). celui ci est défini par un ensemble d'objet dimensions textes, objets dimension sémantiques , objet fait et un ensemble de mesures. Le Drill down et le roll up le long des hiérarchies structurelles permet de faire des analyses sur de différents niveaux hiérarchiques. Le Drill down et le roll up le long des hiérarchies sémantiques permet de faire des analyses sémantiques sur de différents niveaux de granularité. Notre cube textuel sémantique peut stocker de différents types de mesure : mesure numérique, mesure textuel,

mesure sémantique...etc.

Les analyses multidimensionnelles reposent sur une capacité à résumer les informations en les agrégeant avec des fonctions d'agrégation. Toutefois, il n'existe pas de moyen dans un environnement OLAP pour agréger des données textuelles . Ainsi nous proposons un opérateur d'agrégation baptisé le $Top\_KRankedTopics$ qui permet d'obtenir une vision synthétique des informations. Il agrège un ensemble de n sujets dans un sous-ensemble de k sujets les plus pertinents, puis génère pour chaque sujet une liste de classés de documents.

Pour la validation de nos propositions, nous avons développé et mis en place une plate-forme pour le stockage et l'analyse des données textuelles, regroupant nos différentes contributions théoriques. Cette plateforme permet la construction d'entrepôts de textes et leur alimentation ainsi que leur analyse. Elle a servi à évaluer et à valider l'efficacité de nos propositions par le biais d'un travail d'expérimentation, tout en assurant une validation fonctionnelle de notre approche. L'étude expérimentale a montré que l'approche ETL améliore considérablement la qualité de l'information extraite du document textuel. De plus, le nouvel opérateur $Top\_KRankedTopic$ permet d'effectuer des analyses en ligne efficaces sur les données textuelles tout en tenant compte des différents facteurs sémantiques. En outre, nous avons procédé à une évaluation sur les articles de presse, ce qui montre que, la prise en compte de la pertinence contextuelle, améliore considérablement la précision des résultats obtenus par notre opérateur $Top\_KRankedTopic$.

Perspectives

Les travaux de recherche menés dans le cadre de cette thèse ouvrent diverses perspectives de recherche. Nous proposons dans ce qui suit quelques perspectives à ces travaux pour améliorer davantage notre approche :

Analyse en ligne OLAP : nous envisageons de développer de nouveaux opérateur OLAP, tels que :

- Des opérateurs de modification de cube textuel sémantique, qui vont permettre à l'utilisateur de modifier un cube déjà existant.
- Des opérateurs d'agrégation de données textuelles.
- Des opérateurs de manipulation de cube textuel sémantique.

Visualisation de résultats : les résultats d'analyse donnés par notre approche sont visualiser au moyen d'une table multi-dimensionnelle. Hors cette visualisation ne devient pas possible lorsqu'il s'agit de grand volume de données ou lorsqu'il s'agit de plusieurs axes d'analyse. Nous envisageons de voir de plus près les travaux sur la visualisation des données textuelles afin de pouvoir proposer un moyen de visualisation qui s'adapte à l'analyse multidimensionnelles des données textuelles.

Interface graphique : nous prévoyons de fournir des interfaces graphique pouvant interagir avec les modules déjà développés pour aider les analystes et l'administrateur de l'entrepôt à concevoir l'entrepôt de textes et à effectuer les différents traitements possibles.

Performance : les traitements effectués par notre module ETL sont très couteux en terme de temps, surtout lorsqu'il s'agit de très grands volumes de données textuelles. Pour améliorer les performances de notre module, nous envisageons de développer ce dernier sur un environnement MapReduce.

Annexe A

Liste des publications

A.1 Revues internationales

- S. Attaf, N. Benblidia, and O. Boussaid, Warehousing and analysing textual data, IJMNO, vol. 8, no. 3, pp. 216-244, 2018.

A.2 Conférences internationales

- S. Attaf and N. Benblidia, Modelisation multidimensionnelle des données textuelles ou en sommes-nous ?, in ASD Conference Proceedings, pp. 3-25, Conference maghrebine sur les avancees des systemes decisionnels, 2013.
- S. Attaf, N. Benblidia, and O. Boussaid, The multidimensional semantic model of text objects(msmto) : A framework for text data analysis, in Model and Data Engineering - 4th International Conference, MEDI 2014, Larnaca, Cyprus, September 24-26, 2014. Proceedings, pp. 113-124, 2014.
- S. Attaf, N. Benblidia, and O. Boussaid, Analyse des données textuelles : Une approche d'extraction de contenu sémantique et un opérateur d'agrégation top_krankedtopics, in Proceeding de la 12 journées francophones sur les Entrepôts de Données et l'Analyse en Ligne, RNTI-B-12, pp. 51-64, 2016.

REFERENCES

1. Sholom M. Weiss, Nitin Indurkhya, and T. Zhang. *Text Mining. Predictive Methods for Analyzing Unstructured Information.* Springer, Berlin, 2004.
2. A.Y. Blei, D.M.and Ng and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(2) :993–1022, 2003.
3. D. Zhang, C. Zhai, and J Han. Topic cube : Topic modeling for olap on multidimensional text databases. *SDM '09 : Proceedings of the 2009 SIAM International Conference on Data Mining, Sparks, NV, USA*, pages 1124–1135, 2009.
4. Fabrice Jouanot. *Dilemma : vers une coopération de systèmes d'informations basée sur la médiation sémantique et la fusion d'objets.* PhD thesis, 2001. Thèse de doctorat dirigée par Yétongnon, Kokou et Cullot, Nadine Informatique Dijon 2001.
5. Attaf, Nadjia Benblidia, and Omar Boussaid. Analyse des données textuelles : Une approche d'extraction de contenu sémantique et un opérateur d'agrégation top_krankedtopics. In *Proceeding de la 12 journées francophones sur les Entrepôts de Données et l'Analyse en Ligne, RNTI-B-12*, pages 51–64, 2016.
6. Sarah Attaf and Nadjia Benblidia. Modelisation multidimensionnelle des donnees textuelles ou en sommes-nous ? In *ASD Conference Proceedings*, pages 3–25. Conference maghrebine sur les avancees des systemes decisionnels, 2013.
7. Sarah Attaf, Nadjia Benblidia, and Omar Boussaid. The multidimensional semantic model of text objects(msmto) : A framework for text data analysis. In *Model and Data Engineering - 4th International Conference, MEDI 2014, Larnaca, Cyprus, September 24-26, 2014. Proceedings*, pages 113–124, 2014.
8. Ronen Feldman and Ido Dagan. Knowledge discovery in textual databases (kdt). In *Proceedings of the First Internatio-*

nal Conference on Knowledge Discovery and Data Mining, KDD'95, pages 112–117. AAAI Press, 1995.

9. Ming-Syan Chen, Jiawei Han, and P. S. Yu. Data mining : an overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8(6) :866–883, 1996.
10. M. Doroodchi, A. Iranmehr, and S. A. Pouriyeh. An investigation on integrating xml-based security into web services. In *2009 5th IEEE GCC Conference Exhibition*, pages 1–5, March 2009.
11. Seyedamin Pouriyeh and Mahmood Doroodchi. Secure sms banking based on web services. In *In SWWS*, pages 79–83, 01 2009.
12. Seyedamin Pouriyeh, Mahmood Doroodchi, and M R. Rezaeinejad. Secure mobile approaches using web services. In *In SWWS*, pages 75–78, 01 2010.
13. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
14. Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.
15. Anne Kao and Steve R. Poteet. *Natural Language Processing and Text Mining*. Springer Publishing Company, Incorporated, 2006.
16. M. Rajman and R. Besançon. *Text Mining : Natural Language techniques and Text Mining applications*, pages 50–64. Springer US, Boston, MA, 1998.
17. Jim Cowie and Wendy Lehnert. Information extraction. *Commun. ACM*, 39(1) :80–91, January 1996.
18. Sunita Sarawagi.
19. Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. A brief survey of text mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20(1) :19–62, May 2005.

20. Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. Introduction to the special issue on summarization. *Comput. Linguist.*, 28(4) :399–408, December 2002.
21. Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.
22. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1) :1–47, March 2002.
23. Ronen Feldman and James Sanger. *Text Mining Handbook : Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, New York, NY, USA, 2006.
24. Jiawei Han and Micheline Kamber. *Data mining : concepts and techniques*. Kaufmann, San Francisco [u.a.], 2005.
25. I.H. Witten and E. Frank. *Data Mining : Practical Machine Learning Tools and Techniques, Second Edition*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2005.
26. Serkan Günal, Semih Ergin, M. Bilginer Gülmezoğlu, and Ö. Nezih Gerek. *On Feature Extraction for Spam E-Mail Detection*, pages 635–642. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
27. Guozhong Feng, Jianhua Guo, Bing-Yi Jing, and Lizhu Hao. A bayesian feature selection paradigm for text classification. *Inf. Process. Manage.*, 48(2) :283–302, March 2012.
28. Songbo Tan, Yuefen Wang, and Gaowei Wu. Adapting centroid classifier for document categorization. *Expert Systems with Applications*, 38(8) :10264 – 10273, 2011.
29. Alper Kursat Uysal and Serkan Gunal. The impact of preprocessing on text classification. *Information Processing Management*, 50(1) :104 – 112, 2014.
30. Jonathan J. Webster and Chunyu Kit. Tokenization as the initial phase in nlp. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 4*, COLING '92, pages 1106–1110, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

31. Hassan Saif, Miriam Fernández, Yulan He, and Harith Alani. On stopwords, filtering and data sparsity for sentiment analysis of twitter. In *LREC 2014, Ninth International Conference on Language Resources and Evaluation. Proceedings.*, pages 810–817, 2014.
32. C. Silva and B. Ribeiro. The importance of stop word removal on recall values in text categorization. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 1661–1666 vol.3, July 2003.
33. Fengxi Song, Shuhai Liu, and Jingyu Yang. A comparative study on text representation schemes in text categorization. *Pattern Analysis and Applications*, 8(1) :199–209, Sep 2005.
34. G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.
35. G. Salton. The performance of interactive information retrieval. *Information Processing Letters*, 1(2) :35 – 41, 1971.
36. Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.
37. G. Salton and M. E. Lesk. The smart automatic document retrieval systems—an illustration. *Commun. ACM*, 8(6) :391–398, June 1965.
38. *Term Weighting Revisited*. PhD thesis.
39. Joon Ho Lee. Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 180–188, New York, NY, USA, 1995. ACM.
40. Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using smart. In Donna K. Harman, editor, *TREC*, volume Special Publication 500-207, pages 59–72. National Institute of Standards and Technology (NIST), 1992.

41. Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5) :513–523, August 1988.

42. Scott Deerwester, Susan T. Dumais, George Furnas, Thomas Landauer, and Richard Harshman. Indexing by latent semantic analysis. 41 :391–407, 09 1990.

43. Hinrich Schütze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 229–237, New York, NY, USA, 1995. ACM.

44. Thomas K Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3) :259–284, 1998.

45. D. Zhang, C. Zhai, and J Han. Probabilistic latent semantic indexing. pages 50–57. SIGIR, 1999.

46. L. Douglas Baker and Andrew Kachites McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 96–103, New York, NY, USA, 1998. ACM.

47. Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, 7(4) :399–424, Oct 2003.

48. Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather : A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 318–329, New York, NY, USA, 1992. ACM.

49. Fionn Murtagh. A survey of recent advances in hierarchical clustering algorithms. 26 :354–359, 11 1983.

50. Fionn Murtagh. Complexities of hierarchic clustering algorithms : State of the art. 1, 01 1984.

51. Peter Willett. Recent trends in hierarchic document clustering : A critical review. *Information Processing Management*, 24(5) :577 – 597, 1988.

52. Tom Griffiths and Mark Steyvers. A probabilistic approach to semantic representation. In *Proceedings of the 24th annual conference of the cognitive science society*, pages 381–386. Citeseer, 2002.

53. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1) :1–38, 1977.

54. Sourav S. Bhowmick, Sanjay Kumar Madria, and Wee Keong Ng. Representation of web data in A web warehouse. *Comput. J.*, 46(3) :229–262, 2003.

55. Lucie Xyleme. A dynamic warehouse for XML data of the web. *IEEE Data Eng. Bull.*, 24(2) :40–47, 2001.

56. Edwige Pissaloux, Jane You, James Liu, and Tharam S. Dillon. On hierarchical multimedia information retrieval. In *ICIP (2)*, pages 729–732, 2001.

57. Anne-Muriel Arigon, Maryvonne Miquel, and Anne Tchounikine. Multimedia data warehouses : a multiversion model and a medical application. *Multimedia Tools Appl.*, 35(1) :91–108, 2007.

58. Andrei Vanea and Rodica Potolea. Semantically enhancing multimedia data warehouses - using ontologies as part of the metadata. In *ICEIS 2011 - Proceedings of the 13th International Conference on Enterprise Information Systems, Volume 1, Beijing, China, 8-11 June, 2011*, pages 163–168, 2011.

59. and Ganesh K. Bleyberg, M. Dynamic multi-dimensional models for text warehouses in : Systems, man, and cybernetics. In *Lecture Notes in Computer Science (LNCS)*, pages 2045–2050. IEEE International Conference, 2000.

60. M. Catherine McCabe, Jinho Lee, Abdur Chowdhury, David A. Grossman, and Ophir Frieder. On the design and evaluation of a multi-dimensional approach to information retrieval. In *SIGIR*, pages 363–365, 2000.

61. Matteo Golfarelli, Stefano Rizzi, and Boris Vrdoljak. Data warehouse design from XML sources. In *Proceedings of the 4th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2001), Atlanta, Georgia, USA, November 9, 2001*, pages 40–47, 2001.

62. Boris Vrdoljak, Marko Banek, and Stefano Rizzi. Designing web warehouses from XML schemas. In *Data Warehousing and Knowledge Discovery, 5th International Conference, DaWaK 2003, Prague, Czech Republic, September 3-5,2003, Proceedings*, pages 89–98, 2003.

63. Byung-Kwon Park, Hyoil Han, and Il-Yeol Song. XML-OLAP : A multidimensional analysis framework for XML warehouses. In *Data Warehousing and Knowledge Discovery, 7th International Conference, DaWaK 2005, Copenhagen, Denmark, August 22-26, 2005, Proceedings*, pages 32–42, 2005.

64. Akihiro Inokuchi and Koichi Takeda. A method for online analytical processing of text data. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 455–464, 2007.

65. Steven Keith, Owen Kaser, and Daniel Lemire. Analyzing large collections of electronic text using OLAP. *CoRR*, abs/cs/0605127, 2006.

66. Omar Khrouf, Kaïs Khrouf, and Jamel Feki. OLAP de documents. modélisation et mise en œuvre. *Ingénierie des Systèmes d'Information*, 21(1) :11–37, 2016.

67. J. Lee, O. Grossman, D .and Frieder, and McCabe. Mire : A multidimensional information retrieval engine for structured data and text. *Proceedings of the International Conference on*

Information Technology : Coding and Computing, Las Vegas, NV, USA, 14 :224–229, 2002.

68. 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.
69. A. Tseng, S.C. Frank, Y.H. Annie, and Chou. The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence. *Decision Support*, 42 :727–744, 2006.
70. C. X. Lin, B. Ding, J. Han, F. Zhu, and B. Zhao. Text cube : Computing ir measures for multidimensional text database analysis. *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, 54 :905–910, 2008.
71. MJM. Bautista, C. Molina, E. Tejeda3, and A.M Vila. Using textual dimensions in data warehousing processes. *International Conference, IPMU, Dortmund, Germany*, pages 158–167, 2010.
72. M.J.M. Bautista, M. Prados, M.A. Vila, and S Martinez-Folgoso. A knowledge representation for short texts based on frequent itemsets. *Proceedings of IPMU, Paris, France*, 2006.
73. D. Zhang, C. Zhai, and J Han. Mitexcube :microtextcluster cube for online analysis of text cells. *The NASA Conference on Intelligent Data Understanding (CIDU)*, pages 204–218, 2011.
74. K. Khrouf and C. Dupuy. Conception d'entrepôts de documents décisionnels. *INFORSID*, pages 387–401, 2001.
75. K. Khrouf and C. Dupuy. Docware : Vers l'entreposage et l'analyse multidimensionnelle de documents. *CORIA*, pages 405–420, 2005.
76. R. Tounier. *Analyse en ligne (OLAP) de documents*. Thse de doctorat, Universit Toulouse III . Paul Sabatier, 2007.

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

78. D. Boukraa, O. Boussaid, F. Bentayeb, and D Zegour. Modle multidimensionnel d'objets complexes. du modle d'objets aux cubes d'objets complexes. *Ingénierie des Systèmes d'Information*, 16, 2011.

79. D. Boukraa. *Complex Object Data Warehouses : Multidimensional Modeling and Vertical Fragmentation.* Thse de doctorat, Ecole Nationale Suprieure d ?Informatique, Algerie, 2013.

80. M. Azabou, K. Khrouf, J. Feki, C. Soulé-Dupuy, and N. Vallès. Diamond multidimensional model and aggregation operators for document olap. In *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, pages 363–373, May 2015.

81. Rachid Aknouche. *Entrepôt de textes : de l'intégration à la modélisation multidimensionnelle de données textuelles.* PhD thesis, 2014. Thèse de doctorat dirigée par Boussaid, Omar et Bentayeb, Fadila Informatique Lyon 2 2014.

82. Panos Vassiliadis, Alkis Simitsis, and Spiros Skiadopoulos. Conceptual modeling for etl processes. In *Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP*, DOLAP '02, pages 14–21, New York, NY, USA, 2002. ACM.

83. Juan Trujillo and Sergio Luján-Mora. *A UML Based Approach for Modeling ETL Processes in Data Warehouses*, pages 307–320. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

84. Alkis Simitsis, Panos Vassiliadis, Manolis Terrovitis, and Spiros Skiadopoulos. Graph-based modeling of etl activities with multi-level transformations and updates. In *Proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery*, DaWaK'05, pages 43–52, Berlin, Heidelberg, 2005. Springer-Verlag.

85. Alkis Simitsis and Panos Vassiliadis. A methodology for the conceptual modeling of etl processes. In *In Proc. of DSE'03*, 2002.

86. Alkis Simitsis and Panos Vassiliadis. A method for the mapping of conceptual designs to logical blueprints for etl processes. *Decis. Support Syst.*, 45(1) :22–40, April 2008.

87. Oscar Romero, Alkis Simitsis, and Alberto Abelló. Gem : Requirement-driven generation of etl and multidimensional conceptual designs. In *Proceedings of the 13th International Conference on Data Warehousing and Knowledge Discovery*, DaWaK'11, pages 80–95, Berlin, Heidelberg, 2011. Springer-Verlag.

88. Faten Atigui, Franck Ravat, Ronan Tournier, and Gilles Zurfluh. A unified model driven methodology for data warehouses and etl design. In Runtong Zhang, José Cordeiro, Xuewei Li, Zhenji Zhang, and Juliang Zhang, editors, *ICEIS (1)*, pages 247–252. SciTePress.

89. Lilia Muñoz, Jose-Norberto Mazón, and Juan Trujillo. Automatic generation of etl processes from conceptual models. In *Proceedings of the ACM Twelfth International Workshop on Data Warehousing and OLAP*, DOLAP '09, pages 33–40, New York, NY, USA, 2009. ACM.

90. Faten Atigui, Franck Ravat, Olivier Teste, and Gilles Zurfluh. Using OCL for Automatically Producing Multidimensional Models and ETL Processes (regular paper). In Alfredo Cuzzocrea and Umeshwar Dayal, editors, *International Conference on Data Warehousing and Knowledge Discovery (DaWaK), Vienna (Austria), 03/09/2012-06/09/2012*, LNCS, pages 42–53, http ://www.springerlink.com, septembre 2012. Springer.

91. Faten Atigui, Franck Ravat, Olivier Teste, and Gilles Zurfluh. Modélisation conjointe des données et des processus pour l'implantation de schémas d'entrepôts. *Journal of Decision Systems*, 21(1) :27–49, 2012.

92. Zineb El Akkaoui, Esteban Zimányi, Jose-Norberto Mazón, and Juan Trujillo. A bpmn-based design and maintenance framework for ETL processes. *IJDWM*, 9(3) :46–72, 2013.
93. Zineb El Akkaoui, Esteban Zimànyi, Jose-Norberto Mazón, and Juan Trujillo. A model-driven framework for etl process development. In *Proceedings of the ACM 14th International Workshop on Data Warehousing and OLAP*, DOLAP '11, pages 45–52, New York, NY, USA, 2011. ACM.
94. Crist-Jan Doedens. *Text databases*. editions Rodopi, 1994.
95. Ellen M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, pages 171–180, New York, NY, USA, 1993. ACM.
96. Mohamed Mbarki and Chantal Soulé-Dupuy. A conceptual modeling of multimedia documents. In *Proceedings of the IADIS International Conference WWW/Internet 2004, Madrid, Spain, 2 Volumes*, pages 1051–1056, 2004.
97. Mohamed Mbarki, Chantal Soulé-Dupuy, and Nathalie Vallès-Parlangeau. Vers une exploitation flexible de documents multimédia. In *Actes du XXIIIème Congrès INFORSID, Grenoble, France, 24-27 mai, 2005*, pages 95–112, 2005.
98. Mohand Boughanem, Mohamed Tmar, and Hamid Tebri. Filtrage d'information. In M. Ihadjadene, editor, *Méthodes avancées pour la recherche d'informations*, pages 137–162. Hermes-Lavoisier, LAVOISIER 2004, Paris, 2004. hermes science.
99. Rachid Aknouche, Ounas Asfari, Fadila Bentayeb, and Omar Boussaid. Decisional architecture for text warehousing : Etl-text process and multidimensional model twm. In *Proceedings of the 19th International Conference on Management of Data*, COMAD '13, pages 101–104, Mumbai, India, India, 2013. Computer Society of India.
100. Thomas Landauer and Susan Dumais. A solution to Plato's problem : The Latent Semantic Analysis theory of the acqui-

sition, induction, and representation of knowledge. *Psychological Review*, 104 :211 – 240, 1997.

101. Roman Kern, Kris Jack, Maya Hristakeva, and Michael Granitzer. Teambeam - meta-data extraction from scientific literature. *D-Lib Magazine*, 18, 2012.
102. S. Marinai. Metadata extraction from pdf papers for digital library ingest. In *2009 10th International Conference on Document Analysis and Recognition*, pages 251–255, July 2009.
103. Sarah Attaf, Nadjia Benblidia, and Omar Boussaid. Warehousing and analysing textual data. *IJMNO*, 8(3) :216–244, 2018.
104. Lilia Hanachi, Ounas Asfari, Nadjia Benblidia, Fadila Bentayeb, Nadia Kabachi, and Omar Boussaid. Community extraction based on topic-driven-model for clustering users tweets. In *Lecture Notes in Artificial Intelligence (LNAI)*, pages 39–51. ADMA, 2012.
105. Joana Trajkova and Susan Gauch. Improving ontology-based user profiles. In *Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications) - RIAO 2004, 7th International Conference, University of Avignon, France, April 26-28, 2004. Proceedings*, pages 380–390, 2004.
106. Surajit Chaudhuri and Umeshwar Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26(1) :65–74, 1997.
107. Ralph Kimball. *The data warehouse toolkit : Practical Techniques for Building Dimensional Data Warehouses.* John Wiley and Sons, 1996.
108. D. Boukraa, O. Boussaid, F. Bentayeb, and D Zegour. Modle multidimensionnel d'objets complexes : Du modele d'objets aux cubes d'objets complexes. *Ingénierie des Systèmes d'Information*, 16, 2011.
109. Ronan Tournier, vGeneviève Pujolle, Franck Ravat, and Olivier Teste. Fonctions d'agrégation pour l'analyse en ligne (OLAP) de données textuelles. fonctions top_kwk et avg_kw

opérant sur des termes. *Ingénierie des Systèmes d'Information*, 13(6) :61–84, 2008.

110. Sandra Bringay, Nicolas Béchet, Flavien Bouillot, Pascal Poncelet, Mathieu Roche, and Maguelonne Teisseire. Analyse de gazouillis en ligne. In *Actes des 7èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne, Clermont-Ferrand, France, EDA 2011, Juin 2011*, pages 87–102, 2011.
111. Reiner Kraft, Chi Chao Chang, Farzin Maghoul, and Ravi Kumar. Searching with context. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 477–486, New York, NY, USA, 2006. ACM.
112. Lamia Oukid, Nadjia Benblidia, Fadila Bentayeb, Ounas Asfari, and Omar Boussaid. Contextualized text OLAP based on information retrieval. *IJDWM*, 11(2) :1–21, 2015.