Bien cube, les données textuelles peuvent s'agréger !

Sandra Bringay, Anne Laurent, Pascal Poncelet, Mathieu Roche, Maguelonne Teisseire

Bien cube, les données textuelles peuvent s'agréger !

Sandra Bringay*,**, Anne Laurent*,
Pascal Poncelet*, Mathieu Roche*, Maguelonne Teisseire*,***

*LIRMM – CNRS, 161 rue Ada, Montpellier, France
{bringay,laurent,poncelet,mroche,teisseire}@lirmm.fr
**Univ. Montpellier 3
***CEMAGREF – UMR TETIS, maguelonne.teisseire@cemagref.fr

Résumé. La masse des données aujourd'hui disponibles engendre des besoins croissants de méthodes décisionnelles adaptées aux données traitées. Ainsi, récemment de nouvelles approches fondées sur des cubes de textes sont apparues pour pouvoir analyser et extraire de la connaissance à partir de documents. L'originalité de ces cubes est d'étendre les approches traditionnelles des entrepôts et des technologies OLAP à des contenus textuels. Dans cet article, nous nous intéressons à deux nouvelles fonctions d'agrégation. La première propose une nouvelle mesure de TF-IDF adaptative permettant de tenir compte des hiérarchies associées aux dimensions. La seconde est une agrégation dynamique permettant de faire émerger des groupements correspondant à une situation réelle. Les expériences menées sur des données issues du serveur HAL d'une université confirment l'intérêt de nos propositions.

1 Introduction

Avec le développement de l'Internet, de plus en plus de documents textuels sont disponibles. Extraire de la connaissance ou analyser et interroger de tels volumes de données est un enjeu important et de nombreux travaux de recherche se sont intéressés à ces problématiques. Ainsi, par exemple, les travaux menés autour de la fouille de textes ont proposé de nouvelles approches pour classer automatiquement des documents (Sebastiani (2002)), rechercher les nouvelles tendances (Saga et al. (2009)) ou extraire de l'information dans des données textuelles (Chang et al. (2006)). Plus récemment, de nouvelles approches fondées sur des cubes de textes proposent d'utiliser les technologies OLAP pour analyser et extraire de la connaissance. L'un des avantages de ces approches est notamment de pouvoir utiliser des opérateurs comme $Roll$-Up ou $Drill$-$Down$ pour naviguer au travers des hiérarchies et ainsi agréger les données en fonction des requêtes utilisateurs.

De manière à illustrer les problématiques que nous étudions dans cet article, considérons, par exemple, les documents extraits de dépêches concernant le virus de la grippe $A(H_1N_1)$. En étudiant les différents articles, il est aisé de constater que plusieurs catégories de documents peuvent apparaître : articles sur le vaccin, articles sur de nouveaux cas déclarés, articles sur les recommandations ou même articles généraux. Dans un processus d'aide à la décision, si nous désirons retrouver les mots caractéristiques de chaque catégorie, nous pouvons utiliser

des entrepôts de données. Dans un tel contexte, il est indispensable d'extraire pour chacune des catégories les termes les plus représentatifs en tenant compte du fait qu'il peut exister une hiérarchie entre les différentes catégories (c-à-d. la catégorie "vaccin" peut être divisée en "vaccin en Europe", "vaccin en Asie", "vaccin aux Etats-Unis", etc). Dans cet exemple, nous considérons qu'il existe une hiérarchie disponible. Toutefois une telle connaissance n'est pas forcément aisée à obtenir et sa définition n'est pas toujours caractéristique d'une réalité. Par exemple, pourquoi établir une distinction entre "vaccin en Europe" et "vaccin aux Etats Unis" ? Cette distinction est d'autant plus complexe à effectuer lorsque l'utilisateur ne sait pas au préalable qu'il peut exister dans les documents des spécificités propres aux régions.

Dans cet article, notre contribution est double. D'une part, nous proposons un nouveau modèle de données qui permet de construire un entrepôt de données textuelles afin de répondre aisément aux demandes des décideurs via des requêtes OLAP en tenant compte des hiérarchies existantes. D'autre part, nous étendons ce modèle à la définition automatique de dimensions générées à partir des documents étudiés sur lesquels le décideur pourra également naviguer.

Le reste de l'article est organisé de la manière suivante. Dans la section 2, nous présentons notre problématique à partir d'une base exemple qui illustrera les différents concepts introduits. Dans la section 3, nous décrivons les travaux antérieurs liés à ce contexte. La section 4 détaille notre proposition. Les expérimentations menées sont décrites dans la section 5. Enfin nous concluons cet article en présentant quelques perspectives.

2 Problématique

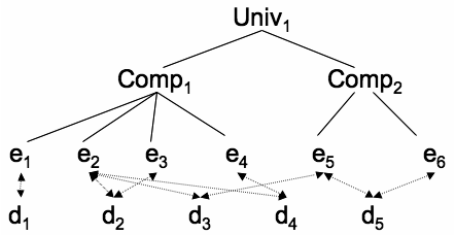

FIG. 1 – *La hiérarchie associée à l'Enseignement*.

Supposons que nous ayons une hiérarchie liée aux enseignements définie de la manière suivante : des enseignants appartiennent à une composante et une université est composée de plusieurs composantes (UFR, IUT, etc). La figure 1 décrit une telle représentation dans laquelle $Comp_1$ est composée de quatre enseignants, $Comp_2$ est composée de deux enseignants et les deux composantes appartiennent à une même Université $Univ_1$. Sur cette figure, les différentes descriptions de cours (documents textuels) associées aux enseignants sont également représentées. Nous pouvons ainsi constater sur la figure que l'enseignant e_1 est attaché seul à la description d_1 d'un cours alors que l'enseignant e_2 enseigne avec un autre intervenant d'une autre composante (enseignant e_5) pour effectuer le cours d_3. Nous considérons par la suite qu'un document propre à un enseignement est décrit par un ensemble de mots-clés. Par exemple, le document d_5 est décrit par les mots-clés m_{15}, m_{16}, m_{17}, m_{18} et m_{19}. Le tableau 1 décrit les différentes caractéristiques des enseignements.

Doc.	enseignant	composante	Univ.	Liste de mots clés
d_1	e_1	$Comp_1$	$Univ_1$	$\{m_1,m_2,m_3,m_4,m_5\}$
d_2	e_2	$Comp_1$	$Univ_1$	$\{m_6,m_7,m_8,m_9,m_{10}\}$
	e_3	$Comp_1$	$Univ_1$	
d_3	e_2	$Comp_1$	$Univ_1$	$\{m_6,m_7,m_8,m_{11},m_{12}\}$
	e_5	$Comp_2$	$Univ_1$	
d_4	e_2	$Comp_1$	$Univ_1$	$\{m_6,m_{13},m_{14},m_{11},m_{10}\}$
	e_4	$Comp_1$	$Univ_1$	
d_5	e_5	$Comp_2$	$Univ_1$	$\{m_{15},m_{16},m_{17},m_{18},m_{19}\}$
	e_6	$Comp_2$	$Univ_1$	

TAB. 1 – *Liste de mots clés, d'enseignants et des composantes par document*

Rappelons que le but de nos travaux est de proposer des mots-clés représentatifs en fonction des différents niveaux de hiérarchies existants (enseignant, composante, etc) dans un contexte d'entrepôt de données. Ces mots-clés seront présents dans les cellules de l'entrepôt.

De manière classique, nous pouvons regrouper les enseignants suivant les composantes en appliquant la hiérarchie existante. Ceci permet de sélectionner les mots les plus discriminants à mettre en relief pour le décideur. Ainsi, si nous prenons le niveau "enseignant", nous pouvons constater par exemple que pour le décideur les mots représentatifs pour l'enseignant e_2 sont : m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13} et m_{14}. Par contre si nous nous intéressons au niveau "composantes", nous constatons que les mots représentatifs diffèrent. Ainsi, notre première contribution dans cet article est d'utiliser et d'adapter des mesures issues du domaine de la Recherche d'Information dans un contexte d'entrepôt de données. Ces mesures proposent d'utiliser les connaissances liées à une organisation hiérarchique existante afin de sélectionner les mots-clés les plus discriminants en fonction du niveau de la hiérarchie interrogée.

Considérons à nouveau les différents mots utilisés dans les documents. Si nous examinons l'enseignant e_1 qui appartient à $Comp_1$, nous pouvons constater qu'il ne partage aucun de ses enseignements avec les membres de sa composante alors que e_2 et e_5, de composantes différentes, partagent par contre des enseignements. Notre objectif dans la seconde contribution de cet article est de faire émerger ce type de comportement et donc de permettre au décideur de connaître les regroupements réels des enseignants indépendamment de toute hiérarchie existante.

3 Travaux Antérieurs

Les entrepôts de données ont été introduits au début des années 1990 (Codd et al. (1993)) pour répondre aux besoins grandissants des décideurs. Ceux-ci souhaitaient alors être munis de bases de données non pas dédiées au stockage robuste de leurs données pour répondre à des requêtes simples et répétitives (bases de données transactionnelles) mais plutôt à une représentation de leurs données en vue de prendre les meilleures décisions et répondre à des requêtes non répétitives et plus complexes. Le modèle multidimensionnel a alors été proposé pour répondre à ce besoin, et permet d'étudier un ensemble d'indicateurs (ou mesures) en fonction de plusieurs dimensions, chaque dimension pouvant être munie d'une ou plusieurs

hiérarchies. Les opérateurs OLAP permettent de naviguer de manière intuitive dans de telles données multidimensionnelles (par exemple pour visualiser les données à différents niveaux de hiérarchies).

Quelques travaux récents se sont intéressés à intégrer les données textuelles dans un contexte d'entrepôt de données. Dans ce cadre, des méthodes d'agrégation adaptées aux données textuelles ont été proposées. Par exemple, les travaux de Keith et al. (2005) proposent d'utiliser des approches de TALN (Traitement Automatique du Langage Naturel) pour agréger les mots ayant la même racine ou les mêmes lemmes (connaissances morpho-syntaxiques). Les auteurs proposent également de rassembler les mots sur la base de classifications sémantiques généralistes existantes (WordNet et Roget). Outre l'utilisation de connaissances morpho-syntaxiques et sémantiques pour agréger les données textuelles, d'autres travaux utilisent des approches numériques issues du domaine de la Recherche d'Information (RI) pour agréger les données textuelles (Pujolle et al. (2008); Lin et al. (2008); Pérez-Martínez et al. (2008)). Ainsi, Lin et al. (2008) agrège les documents sur la base des mots-clés présents dans ces derniers en utilisant une hiérarchie sémantique des mots présents dans l'entrepôt et des mesures issues de la RI. De telles méthodes issues de la RI sont aussi utilisées dans les travaux de Pérez-Martínez et al. (2008) qui consistent à prendre en compte une dimension "contexte" et "pertinence" pour construire un entrepôt de données textuelles appelé R-Cube. Certaines approches proposent d'ajouter une nouvelle dimension spécifique. Par exemple, dans (Zhang et al. (2009)) les auteurs ajoutent une dimension 'topic' et appliquent l'approche PLSA (Hofmann (1999)) pour extraire les thèmes représentatifs des documents dans cette nouvelle dimension. Enfin, Pujolle et al. (2008) proposent d'agréger des parties de documents afin d'offrir au décideur des mots-clés caractéristiques propres à cette agrégation. Dans ce cadre, les auteurs utilisent une première fonction pour sélectionner les mots-clés les plus significatifs en utilisant la mesure $TF\text{-}IDF$ classique issue du domaine de la RI. L'objectif de nos travaux est assez similaire à cette dernière approche. Toutefois, nous souhaitons étendre la prise en compte de la hiérarchie dans les documents rendus aux décideurs. En d'autres termes, nous souhaitons ne retourner que les mots-clés significatifs par rapport à un niveau donné. Par exemple, dans le cas des mots-clés significatifs, leur agrégation *Avg-Kw* qui utilise un $TF\text{-}IDF$ ne permet de connaître que les mots significatifs pour des chercheurs mais ne permet pas de prendre en compte une hiérarchie existante. Pujolle et al. (2008) proposent également d'utiliser une ontologie légère. Toutefois, là aussi, l'objectif est différent car ils s'intéressent à la représentation de mots génériques par rapport aux mots du domaine. Dans notre cas, nous souhaitons extraire les mots clés significatifs par rapport à une hiérarchie existante. En outre, nous souhaitons pouvoir effectuer des regroupements significatifs pour le décideur même s'il n'existe pas de hiérarchie fixe définie.

4 Contribution

Dans cette section nous présentons notre approche. Dans un premier temps, nous décrivons le modèle de données utilisé. Nous présentons ensuite l'agrégation adaptative selon le niveau existant puis l'agrégation dynamique. Pour chacune de ces approches nous présentons tout d'abord le principe général puis nous explicitons sa mise en œuvre.

4.1 Le modèle de données

Dans cette section, nous définissons un modèle de données pour représenter les cubes de textes. Une table de faits F est définie sur le schéma $D = \{T, \ldots, T_n, M\}$ où T_i $(i = 1, .., n)$ correspondent aux dimensions (Pérez-Martínez et al. (2008)) et M correspond à une mesure. Les différentes mesures utilisées sont décrites dans les sections suivantes. Chaque dimension T_i est définie sur un domaine $D = dom(T_i)$ partitionné en un ensemble de catégories (ou niveaux de granularité) C_j. On a donc $D = \cup_j C_j$. D doit être muni d'un ordre partiel $\sqsubseteq_D$ permettant de comparer les valeurs du domaine D. Chaque catégorie représente les valeurs associées à un niveau de granularité. Nous notons $e \in D$ pour préciser que e est une valeur de dimension de D, s'il existe une catégorie $C_j \subseteq D$ telle que $e \in \cup_j C_j$. Notons que deux catégories particulières sont distinguées et sont présentes sur toutes les dimensions : $\perp_D$ et $\top_D$ $\in C_D$ correspondant respectivement au niveau de plus fine et de plus forte granularité. Dans le cadre de notre approche, l'ordre partiel défini sur les domaines des dimensions correspond à l'inclusion ensembliste des mots clés associés aux valeurs de dimension considérées. Ainsi, soient deux valeurs $e_1, e_2 \in \cup_j C_j$, on a $e_1 \sqsubseteq_D e_2$ si e_1 est logiquement contenu dans e_2.
Par exemple, la dimension Enseignement de la figure 1 possède les catégories $\perp_{enseignement}$ $= Enseignant \leq Composante \leq Universite \leq \top_{enseignement}$. Les valeurs de dimensions sont $dom(Enseignement) = \{e_1, e_2, e_3, Comp_1, Comp_2, Univ_1, ...\}$ réparties dans ces catégories (niveaux de granularité) de la manière suivante : $Enseignant = \{e_1, e_2, e_3\}$, $Composante = \{Comp_1, Comp_2\}, Univ = \{Univ_1, ...\}$. L'ordre partiel $\sqsubseteq_D$ sur les valeurs des dimensions peut bien entendu être généralisé aux catégories : pour $C_1, C_2 \in C_D$, on a alors $C_1 \leq_D C_2$ si $\exists e_1 \in C_1, e_2 \in C_2$ $tels\ que$ $e_1 \sqsubseteq_D e_2$. Par exemple, nous avons : $e_1 \sqsubseteq_D Comp_1 \sqsubseteq_D Univ_1 \sqsubseteq_D \top$. La prise en compte de hiérarchie dynamique est telle que toutes les catégories de cette dimension doivent respecter l'ordre partiel défini. Notre modèle permet de prendre en compte différentes dimensions (*e.g.* le temps tel que : $\perp_{temps} = mois \leq semestre \leq année \leq \top_{temps}$) et bien entendu une dimension correspondant aux différentes informations extraites des documents. Cette dimension est définie de la manière suivante : il n'existe pas de hiérarchie sur cette dimension et chaque valeur correspond aux mots-clés associés aux documents. Ainsi, pour un enseignant donné, le contenu d'une cellule correspond à la fréquence d'apparition d'un mot dans un document. Par exemple pour l'enseignant e_1 elle peut contenir la fréquence d'apparition du mot m_1.

4.2 Agrégation Adaptative selon le niveau hiérarchique

4.2.1 Présentation générale

Dans nos travaux, nous nous appuyons sur une hiérarchie originale adaptée aux données textuelles. Dans cette hiérarchie, les nœuds sont les éléments que nous souhaitons agréger et les feuilles sont les descripteurs (mots-clés) de ces éléments. Pour chaque agrégation, le but que nous nous fixons est de sélectionner les descripteurs pertinents. Cette sélection dépendra du niveau et des nœuds que nous désirons agréger. Nous proposons dans ce cas une mesure fondée sur la mesure TF-IDF bien connue en Recherche d'Information. Nous montrerons dans la section 4.2.2 de quelle manière nous avons adapté cette mesure à la problématique des entrepôts de données. Notre approche s'appuie sur l'utilisation d'une hiérarchie du domaine, ce qui est relativement classique dans la littérature (voir section 3). Cependant, l'originalité de

notre approche réside dans la méthode d'agrégation qui dépend du niveau traité, d'où le nom d'*agrégation adaptative*. Ainsi, d'un niveau à l'autre les descripteurs pertinents extraits pour caractériser une cellule de notre entrepôt peuvent se révéler extrêmement différents.

En reprenant l'exemple que nous avons décrit en section 2, les mots-clés pour décrire un IUT (par exemple, le mot-clé "Réseaux") n'est pas toujours adapté pour discriminer les enseignements effectués par les intervenants dans une telle composante. En d'autres termes, un tel mot-clé est caractéristique pour décrire un IUT spécialisé par rapport à différentes composantes mais ne permet pas de discriminer les cours d'un IUT "Réseaux et Télécommunication". De la même manière, en utilisant une hiérarchie de laboratoires (laboratoire, équipe, chercheur) un terme très discriminant pour décrire une équipe (par exemple, le terme 'data-mining') n'est pas nécessairement pertinent pour distinguer les membres d'une équipe de fouille de données.

4.2.2 Méthode mise en œuvre

Dans notre processus, la première étape consiste à fusionner chaque feuille correspondant aux attentes de l'utilisateur. Ceci revient à fusionner les mots (de manière booléenne et/ou fréquentielle) des différents documents. Le but de cette étape est de lister tous les mots situés dans les documents correspondant à un niveau donné (par exemple, "enseignant", "composante", "Université"). Si l'utilisateur souhaite axer sa recherche au niveau de l'enseignant e, les mots-clés des articles écrits par e forment le vecteur de cet enseignant. Nous pouvons appliquer ce même principe au niveau des composantes ($Roll\text{-}up$).

À titre d'exemple, en nous appuyant sur la figure 1 et le tableau 1, e_2 est associé aux documents d_2 et d_3. Nous construisons alors un espace vectoriel dont la dimension correspond au nombre de mots rencontrés (ou sélectionnés) dans l'ensemble des documents. Le processus appliqué est illustré dans le tableau 2 qui représente les vecteurs de documents de manière booléenne et fréquentielle.

Représentation booléenne

Mots	d_2	d_3	d_4	ens. e_2
m_1	0	0	0	0
m_2	0	0	0	0
...				
m_6	1	1	1	1
m_7	1	1	0	1
m_8	1	1	0	1
m_9	1	0	0	1
m_{10}	1	0	1	1
m_{11}	0	1	1	1
m_{12}	0	1	0	1
...				
m_{19}	0	0	0	0

Représentation fréquentielle

Mots	d_2	d_3	d_4	ens. e_2
m_1	0	0	0	0
m_2	0	0	0	0
...				
m_6	fr_6^2	fr_6^3	fr_6^4	$fr_6^2 + fr_6^3 + fr_6^4$
m_7	fr_7^2	fr_7^3	0	$fr_7^2 + fr_7^3$
m_8	fr_8^2	fr_8^3	0	$fr_8^2 + fr_8^3$
m_9	fr_9^2	0	0	fr_9^2
m_{10}	fr_{10}^2	0	fr_{10}^4	$fr_{10}^2 + fr_{10}^4$
m_{11}	0	fr_{11}^3	fr_{11}^4	$fr_{11}^3 + fr_{11}^4$
m_{12}	0	fr_{12}^3	0	fr_{12}^3
...				
m_{19}	0	0	0	0

TAB. 2 – *Vecteur de mots clés relatif à l'enseignant* e_2

À partir des vecteurs constitués, nous sélectionnons les termes les plus discriminants par rapport au niveau d'éléments souhaités (par exemple, les enseignants). Pour effectuer une telle

sélection, nous nous appuyons sur la mesure $TF\text{-}IDF$ que nous adaptons à notre problématique. Traditionnellement, la mesure $TF\text{-}IDF$ donne un poids plus important aux mots caractéristiques d'un document (Salton et al. (1975)). Ainsi, pour attribuer un poids de $TF\text{-}IDF$, il est nécessaire, dans un premier temps, de calculer la fréquence d'un terme (*Term Frequency*). Celle-ci correspond au nombre d'occurrences de ce terme dans le document considéré. Ainsi, pour le document d_j et le terme t_i, la fréquence du terme dans le document est donnée par l'équation suivante :

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

où $n_{i,j}$ est le nombre d'occurrences du terme t_i dans d_j. Le dénominateur correspond au nombre d'occurrences de tous les termes dans le document d_j. La fréquence inverse de document (*Inverse Document Frequency*) mesure l'importance du terme dans l'ensemble du corpus. Elle consiste à calculer le logarithme de l'inverse de la proportion de documents du corpus qui contiennent le terme et est définie de la manière suivante :

$$IDF_i = log_2 \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

où $|D|$ représente le nombre total de documents dans le corpus et $|\{d_j : t_i \in d_j\}|$ représente le nombre de documents où le terme t_i apparaît (c.à-d. $n_{i,j} \neq 0$). Enfin, le poids s'obtient en multipliant les deux mesures :

$$TF - IDF_{i,j} = TF_{i,j} \times IDF_i$$

Dans notre cas, nous ne calculons pas les termes représentatifs par rapport au nombre de documents mais plutôt par rapport au niveau de granularité souhaité. Ainsi, dans notre cas, la formule représentant un IDF *adaptatif* est donnée ci-dessous :

$$IDF_i^k = log_2 \frac{|E^k|}{|\{e_j^k : t_i \in e_j\}|}$$

où $|E^k|$ représente le nombre total d'éléments de type k (dans notre exemple, $k = \{$*Enseignant*, *Composante*, *Université*$\}$) qui correspond au niveau de la hiérarchie que le décideur souhaite agréger. $|\{e_j : t_i \in e_j\}|$ est relatif au nombre d'éléments de type k dans lequel le terme t_i apparaît. Cette mesure permet d'attribuer un poids adapté au niveau d'agrégation décidé par l'utilisateur. Ainsi, nous calculons ce poids $TF\text{-}IDF_i^k$ pour chacun des mots t_i. Nous pouvons ainsi conserver les n mots ayant les poids les plus élevés.

Exemple 1 *Nous donnons dans la table 3 l'IDF des mots au regard du niveau traité (Composante, Enseignant) illustré en section 2. Ceci permet de mettre en relief, dans la table 4, les mots caractéristiques de chaque niveau. Par exemple, le mot m_{10} est assez caractéristique de la composante 1 (il ne représente jamais un mot-clé de la composante 2). Cependant, un tel mot-clé est utilisé par de nombreux enseignants (e_2, e_3, e_4), il n'est donc pas caractéristique pour décrire l'enseignement effectué par les enseignants de l'Université. Enfin, pour chaque niveau, nous pouvons calculer le TF-IDF de chacun des mots. Le résultat d'un tel calcul est donné dans la table 5. Pour simplifier la présentation des résultats, nous appliquons une fréquence (TF) de 1 aux mots-clés présents ce qui revient à utiliser une représentation booléenne.*

Bien cube, les données textuelles peuvent s'agréger !

Mot	**Valeur** $IDF^{enseignant}$
m_1	$log_2(6/1) = 2.58$
m_2	$log_2(6/1) = 2.58$
...	...
m_6	$log_2(6/4) = 0.58$
m_7	$log_2(6/3) = 1.00$
...	...
m_{11}	$log_2(6/3) = 1.00$
m_{12}	$log_2(6/2) = 1.58$
...	...

Mot	**Valeur** $IDF^{composante}$
m_1	$log_2(2/1) = 1$
m_2	$log_2(2/1) = 1$
...	...
m_6	$log_2(2/2) = 0$
m_7	$log_2(2/2) = 0$
...	...
m_{11}	$log_2(2/2) = 0$
m_{12}	$log_2(2/2) = 0$
...	...

TAB. 3 – *Exemple de calcul de l'IDF à différents niveaux (enseignant, composante)*

$IDF^{enseignant}$	**Mots**
2.58	$m_1, m_2, ...$
1.58	$m_{12}, ...$
1.00	$m_7, m_{11}, ...$
0.58	$m_6, ...$

$IDF^{composante}$	**Mots**
1	$m_1, m_2, ..$
0	$m_6, m_7, m_{11}, m_{12}, ...$

TAB. 4 – *Mots discriminants triés au niveau "Enseignant" (gauche) - Mots discriminants triés au niveau "Composante" (droite)*

4.3 Agrégation Dynamique

4.3.1 Présentation générale

Dans cette section, nous proposons de mettre en place une mesure dynamique afin d'agréger les éléments. Ce problème d'agrégation dynamique est par exemple étudié dans le contexte des taxonomies dynamiques (e.g., Sacco (2000)). Le fait de nous appuyer sur les seules connaissances de la hiérarchie ne permet pas toujours une agrégation de qualité (représentant une situation réelle). Par exemple, dans le cadre d'une Université, un enseignant peut enseigner des cours très différents de la thématique de sa propre composante. A contrario, les enseignants de deux composantes différentes peuvent se révéler extrêmement proches. Dans ce cas, nous proposons d'effectuer un regroupement sur la base des données textuelles qui permettent de mieux décrire les éléments à agréger.

Enseignant	$TF\text{-}IDF^{Enseignant}$
e_1	$P(m_1) = 1 \times 2.58 = 2.58$ $P(m_2) = 1 \times 2.58 = 2.58$ ...
e_2	$P(m_6) = 1 \times 0.58 = 0.58$ $P(m_7) = 1 \times 1.00 = 1.00$ ...
e_3	$P(m_6) = 1 \times 0.58 = 0.58$ $P(m_7) = 1 \times 1.00 = 1.00$ ...
e_4	...

Composante	$TF\text{-}IDF^{Composante}$
$composante_1$	$P(m_1) = 1 \times 1 = 1$ $P(m_2) = 1$ ... $P(m_6) = 0$ $P(m_7) = 0$ ...
$composante_2$	$P(m_6) = 1 \times 0 = 0$ $P(m_7) = 0$ ... $P(m_{12}) = 0$ ...

TAB. 5 – *TF-IDF au niveau "Enseignant" et "Composante".*

4.3.2 Méthode mise en œuvre

De la même manière que l'agrégation adaptative (section 4.2), la première étape consiste à fusionner les termes des feuilles (c'est-à-dire les documents). Ceci permet de constituer des vecteurs de mots-clés pour chaque niveau (Enseignant, Composante, Université) en appliquant une représentation de *type Salton* (Salton et al. (1975)). Notons que nous pouvons sélectionner un nombre donné de mots-clés (les plus fréquents) pour limiter les informations contenues dans l'entrepôt. Par la suite, nous appliquons une méthode de clustering (Jain et al. (1999)) pour rassembler les éléments qui partagent les mêmes mots. Bien entendu, le résultat d'agrégation obtenu peut différer du regroupement fondé sur une hiérarchie statique comme nous le montrerons dans nos expérimentations (section 5). Les méthodes de clustering que nous utilisons (k-means dans nos expérimentations) s'appuient sur des mesures classiques afin de calculer la proximité entre deux vecteurs : Jaccard (représentation booléenne), cosinus, distance euclidienne (représentation fréquentielle), etc.

Avec les approches de clustering utilisées, pour chaque regroupement formé, nous obtenons un vecteur moyen. Nous sélectionnons alors les mots caractéristiques (c'est-à-dire ayant le poids le plus élevé) de ces vecteurs moyens. Notons que pour cette représentation dynamique, nous pouvons bien sûr accorder des poids booléens, fréquentiels ou de type $TF\text{-}IDF$ pour construire nos vecteurs.

Notons que l'agrégation/désagrégation des éléments pour effectuer une analyse de type OLAP peut s'effectuer par la variation du paramètre k de l'algorithme de clustering appliqué (ici k-means). En effet, le fait de choisir le nombre de groupes à obtenir permet d'agréger/désagréger ($Drill\text{-}down$ et $Roll\text{-}up$) les éléments pour une meilleure analyse.

Exemple 2 *En considérant l'exemple précédent, nous effectuons un regroupement fondé sur un algorithme de type k-means en appliquant différentes valeurs de k (nombre de regroupements différent). Les résultats des regroupements avec les vecteurs moyens associés sont donnés dans les tableaux 6 et 7. Lorsque nous utilisons le même nombre de groupes que la hiérarchie statique (deux composantes), nous remarquons que le regroupement proposé par notre algorithme est différent (cf tableau 6). Ce regroupement reflète davantage la réalité propre aux cours donnés par les enseignants. Avec la hiérarchie statique nous avons : Classe 1 = {e_1, e_2, e_3, e_4}, Classe 2 = {e_5, e_6} alors que pour la hiérarchie dynamique : Classe 1 = {e_1}, Classe 2 = {e_2, e_3, e_4, e_5, e_6}. En effet, le regroupement formé met en exergue le fait que l'enseignant e_1 est isolé par rapport à sa propre composante. Les vecteurs moyens des groupes sont donnés dans le tableau 6. Pour chaque regroupement formé, nous retenons n mots-clés représentatifs (c'est-à-dire, ayant le score le plus élevé). À titre d'exemple, le mot le plus représentatif du groupe formé des enseignants {e_2, e_3, e_4, e_5, e_6} est le mot-clé m_6. Ce dernier a un poids de 0.8 (le mot m_6 est utilisé pour décrire les cours de 4 enseignants sur les 5 du groupe formé dynamiquement).*

5 Expérimentations

De manière à évaluer notre proposition, différentes expérimentations ont été réalisées. Les données utilisées correspondent à des articles publiés dans un laboratoire (LIRMM) sur l'année 2009 (305 articles) et référencés dans la base de données HAL (archive ouverte pluridisciplinaire). Dans ce contexte, nous utilisons une hiérarchie dont l'élément le plus élevé est le

Bien cube, les données textuelles peuvent s'agréger !

Mots	0	1
m_1	0	1
m_2	0	1
m_3	0	1
m_4	0	1
m_5	0	1
m_6	0.8	0
m_7	0.6	0
m_8	0.6	0
m_9	0.4	0
m_{10}	0.6	0
m_{11}	0.6	0
m_{12}	0.4	0
m_{13}	0.4	0
m_{14}	0.4	0
m_{15}	0.4	0
m_{16}	0.4	0
m_{17}	0.4	0
m_{18}	0.4	0
m_{19}	0.4	0

Mots	0	1	2	3
m_1	0	1	0	0
m_2	0	1	0	0
m_3	0	1	0	0
m_4	0	1	0	0
m_5	0	1	0	0
m_6	1	0	1	0.5
m_7	0	0	1	0.5
m_8	0	0	1	0.5
m_9	0	0	1	0
m_{10}	1	0	1	0
m_{11}	1	0	0.5	0.5
m_{12}	0	0	0.5	0.5
m_{13}	1	0	0.5	0
m_{14}	1	0	0.5	0
m_{15}	0	0	0	1
m_{16}	0	0	0	1
m_{17}	0	0	0	1
m_{18}	0	0	0	1
m_{19}	0	0	0	1

Mots	0	1	2	3	4
m_1	0	1	0	0	0
m_2	0	1	0	0	0
m_3	0	1	0	0	0
m_4	0	1	0	0	0
m_5	0	1	0	0	0
m_6	1	0	1	0.5	1
m_7	0	0	1	0.5	1
m_8	0	0	1	0.5	1
m_9	0	0	1	0	1
m_{10}	1	0	1	0	1
m_{11}	1	0	1	0.5	0
m_{12}	0	0	1	0.5	0
m_{13}	1	0	1	0	0
m_{14}	1	0	1	0	0
m_{15}	0	0	0	1	0
m_{16}	0	0	0	1	0
m_{17}	0	0	0	1	0
m_{18}	0	0	0	1	0
m_{19}	0	0	0	1	0

TAB. 6 – *Utilisation de l'algorithme k-means de Weka pour différentes valeurs de k : 2 groupes (gauche), 4 groupes (centre), 5 groupes (droite)*

N.	Instances
0	5 (83%)
1	1 (17%)

N.	Instances
0	1 (17%)
1	1 (17%)
2	2 (33%)
3	2 (33%)

N.	Instances
0	1 (17%)
1	1 (17%)
2	1 (17%)
3	2 (33%)
4	1 (17%)

TAB. 7 – *Les différents regroupements obtenus pour : 2 groupes (gauche), 4 groupes (centre), 5 groupes (droite). Les instances correspondent au nombre d'éléments (enseignants) présents dans chaque groupe.*

laboratoire qui possède plusieurs équipes. Ces dernières sont composées de chercheurs. Les documents traités correspondent aux résumés d'articles. Le but de nos expérimentations est de comparer les agrégations effectuées en utilisant une hiérarchie statique (section 4.2) ou dynamique (section 4.3). Par manque de place, nous ne nous focalisons ici que sur l'agrégation dynamique. Dans un premier temps, nous appliquons l'étiqueteur grammatical TreeTagger[1] sur notre corpus afin de ne retenir que les noms extraits à partir des résumés d'articles. Pour chaque document, nous retenons les m ($m = 10, 20, 30$) mots-clés ayant le plus grand nombre d'occurrences et constituons des vecteurs associés à chaque document.

Les résultats reportés ici concernent un sous-ensemble du laboratoire : 3 équipes représentant 84 chercheurs. Ce sous-ensemble nous permet d'analyser manuellement les résultats obtenus qui sont synthétisés dans le tableau 8. Nous pouvons alors comparer le regroupement dynamique par rapport à un rassemblement fixé par une hiérarchie existante. Nous avons mené nos expérimentations en faisant varier le nombre m de mots sélectionnés pour caractériser un chercheur. Par ailleurs, nous faisons varier le nombre de clusters formés. Les résultats du

[1] http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

m	10	20	30
k=2	1 NSG (11 instances) 1 RG (73 instances)	1 NSG (11 instances) 1 RG (73 instances)	1 NSG (11 instances) 1 RG (73 instances)
k=3	2 NSG (11 et 7 instances) 1 RG (66 instances)	2 NSG (11 et 3 instances) 1 RG (70 instances)	2 NSG (11 et 3 instances) 1 RG (70 instances)
k=4	3 NSG (11, 7, 1 instances) 1 RG (65 instances)	4 NSG (42, 28, 11, 3 instances)	3 NSG (11, 8, 2 instances) 1 RG (63 instances)
k=5	4 NSG (11, 7, 3, 1 instances) 1 RG (62 instances)	5 NSG (34, 28, 11, 8, 3 instances)	4 NSG (11, 8, 7, 2 instances) 1 RG (56 instances)

TAB. 8 – *Modification des groupes proposés par l'agrégation dynamique (algorithme k-means selon différentes valeurs de k). NSG formé = Nouveau Sous-Groupe formé. RG = Rassemblement de groupes.*

tableau 8 montrent que les groupes formés automatiquement sont assez différents de la répartition réelle des chercheurs dans différentes équipes. L'analyse manuelle montre que de nombreux sous-groupes spécifiques sont formés ; ces derniers représentent des ensembles de chercheurs travaillant sur un sous-thème des trois équipes de recherche. À titre d'exemple, avec $k = 3$ et un nombre $m = 20$ de mots sélectionnés, un cluster de 3 individus est créé. Celui-ci correspond à trois chercheurs qui travaillent sur une thématique de "fouille de textes" dans une équipe ayant un axe de recherche plus général. Les autres expérimentions menées avec l'ensemble des 511 chercheurs répertoriés dans les archives HAL 2009 du laboratoire confirment ces observations. Ainsi, l'application d'une agrégation dynamique en fixant par exemple la dimension propre à l'année permet de mettre en relief des informations nouvelles et intéressantes pour les décideurs.

Comparativement à une représentation classique, notre entrepôt de données textuelles nécessite de stocker m mots par document. Ce choix du paramètre m expérimenté dans cette section aura une influence importante quant à la taille finale de notre entrepôt de textes. Notons qu'après avoir effectué les agrégations, un sous-ensemble de ces mots sera retourné à l'utilisateur.

6 Conclusion

Dans cet article, nous avons proposé deux nouvelles fonctions afin d'agréger des données textuelles d'un entrepôt. Nos fonctions permettent de proposer au décideur les mots les plus significatifs issus de ces agrégations. La première fonction s'appuie sur une mesure issue de la Recherche d'Information que nous avons étendue afin de prendre en compte les informations d'une hiérarchie existante. La seconde fonction proposée effectue une agrégation dynamique sur la base d'algorithmes de clustering. Nos expérimentations ont montré que les agrégations effectuées en utilisant ces deux types de fonctions se révèlent différentes ce qui permet d'apporter des informations originales à partir des données textuelles de notre entrepôt. Dans nos futurs travaux, nous souhaitons appliquer et expérimenter ces méthodes d'agrégation pour des données particulières telles que les données d'opinions. Ceci demandera une recherche de descripteurs linguistiques plus précis (syntagmes adjectivaux par exemple) pour caractériser les données textuelles.

Bien cube, les données textuelles peuvent s'agréger !

Références

Chang, C.-H., M. Kayed, M. R. Girgis, et K. F. Shaalan (2006). A survey of web information extraction systems. *IEEE Trans. Knowl. Data Eng. 18*(10), 1411–1428.

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

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *In Proc. of Uncertainty in Artificial Intelligence, UAI'99*, pp. 289–296.

Jain, A. K., M. N. Murty, et P. J. Flynn (1999). Data clustering : A review. *ACM Comput. Surv. 31*(3), 264–323.

Keith, S., O. Kaser, et D. Lemire (2005). Analyzing large collections of electronic text using olap. Technical Report TR-05-001, UNBSJ CSAS.

Lin, C. X., B. Ding, J. Han, F. Zhu, et B. Zhao (2008). Text Cube : Computing IR Measures for Multidimensional Text Database Analysis. In *In Proc. of Int. Conf. on Data Mining (ICDM'08)*, pp. 905–910.

Pérez-Martínez, J. M., R. B. Llavori, M. J. A. Cabo, et T. B. Pedersen (2008). Contextualizing data warehouses with documents. *Decision Support Systems 45*(1), 77–94.

Pujolle, G., F. Ravat, O. Teste, et R. Tournier (2008). 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.

Sacco, G. (2000). Dynamic taxonomies : A model for large information bases. *IEEE Transactions on Knowledge and Data Engineering 12*(3), 468–479.

Saga, R., H. Tsuji, et K. Tabata (2009). Loopo : Integrated text miner for fact-graph-based trend analysis. In *HCI (9)*, pp. 192–200.

Salton, G., A. Wong, et C. S. Yang (1975). A vector space model for automatic indexing. *Commun. ACM 18*(11), 613–620.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv. 34*(1), 1–47.

Zhang, D., C. Zhai, et J. Han (2009). Topic cube : Topic modeling for olap on multidimensional text databases. In *In Proc. of the SIAM Int. Conference on Data Mining*, pp. 1123–1134.

Summary

With the development of the Internet, the amount of textual information grows explosively and it is more and more desirable to provide the end user with new tools for analysing and extracting knowledge from such amount of data. Recently, new approaches have been defined in order to enhance tradionnal Datawarehouse and OLAP technologies by handling text documents. In this paper, we focus on two new aggregative functions. The former is based on an extension of the classical TF-IDF measure to take into account existing hierarchies. The latter proposes to dynamically define a hierarchy in order to emphasize real situation extracted from texts. Experiments conducted on articles stored in the HAL repository show the efficiency of our proposals.