Top_Keyword : agrégation de mots-clefs dans un environnement d'analyse en ligne (OLAP)

Franck Ravat, Olivier Teste
Ronan Tournier, Gilles Zurfluh

IRIT SIG/ED, UMR5505, 118 rte. de Narbonne,
F31062 Toulouse CEDEX 9, France
{ravat, teste, tournier, zurfluh}@irit.fr
http://www.irit.fr

Résumé. Depuis plus d'une décennie, les travaux de recherche sur OLAP et les bases de données multidimensionnelles ont produit des méthodes, des outils et des moyens d'analyse de données numériques. L'accroissement de la disponibilité des documents numériques entraîne un besoin pour l'ajout de documents XML principalement constitués de données textuelles au sein de bases de données multidimensionnelles et d'un environnement adapté à leur analyse. En réponse à ce besoin, cet article présente une nouvelle fonction d'agrégation permettant l'agrégation de données textuelles au sein d'un environnement OLAP, au même titre que les fonctions d'agrégation arithmétique traditionnelles le permettent pour des données numériques. La fonction TOP_KEYWORD (ou TOP_KW) résume un ensemble de documents par leurs termes les plus significatifs, en employant une fonction de pondération issue de la recherche d'information : *tf.idf*.

1 Introduction

Les systèmes d'analyse en ligne OLAP (On-Line Analytical Processing) permettent aux analystes d'améliorer le processus de prise de décision. Ces systèmes facilitent la consultation et l'analyse de données économiques, statistiques ou scientifiques agrégées et historisées via une structuration adaptée au sein de bases de données multidimensionnelles (Colliat, 1996). Les systèmes d'aide à la décision, emploient des bases de données multidimensionnelles (BDM), qui permettent aux décideurs d'avoir une vision des performances d'une entreprise. Pour modéliser les BDM, des structures multidimensionnelles ont été définies permettant la représentation de sujets d'analyse, appelés *faits* et d'axes d'analyse, appelés *dimensions* (Kimball, 1996). Les faits sont des regroupements d'indicateurs d'analyse appelés *mesures*. Les dimensions sont composées d'attributs, agencés de manière hiérarchique, qui modélisent les différents niveaux de détails (granularité) des axes d'analyse.

Lors d'une analyse OLAP multidimensionnelle, les données représentant un sujet sont analysées en fonction de différents niveaux de détails ou niveaux de granularité. Le processus d'analyse agrège les données en fonction des niveaux de granularité sélectionnés via une

fonction d'agrégation (par ex. somme, moyenne, maximum…). Les opérations de forage (drill down et roll up), qui sont parmi les opérations les plus utilisées par les décideurs, font un usage intensif de ces fonctions d'agrégation. Ces opérations permettent au décideur de changer le niveau de granularité utilisé pour afficher les données analysées. Ainsi, lors du changement de niveau, les données sont à nouveau agrégées par l'emploi de la fonction d'agrégation selon le nouveau niveau de granularité. Par exemple, dans la figure FIG. 1, un décideur analyse le *nombre de mots-clefs* employés par *auteur* et par *mois*. Afin d'avoir une vision plus globale, le décideur change le niveau de détails de l'analyse et effectue un forage vers le haut (roll up) changeant le niveau de détails *mois* en *années*. Par conséquent, les valeurs mensuelles sont agrégées en valeurs annuelles pour chaque couple (*auteur*, *année*).

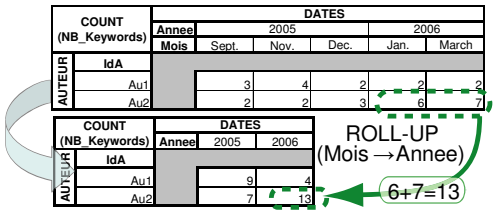

FIG. 1 – *Analyse multidimensionnelle du nombre de mots-clefs par auteur et par mois, suivi d'un forage vers le haut (roll up) vers le niveau de granularité années.*

L'analyse multidimensionnelle basée sur des BDM factuelles numériques est une tâche bien maîtrisée de nos jours (Sullivan, 2001). Ces BDM sont souvent construites sur des données transactionnelles issues des systèmes d'information (SI) des entreprises. Cependant, seul 20% des données d'un SI sont des données transactionnelles et peuvent être traitées (Tseng et Chou, 2006). Les 80% restants, la « paperasserie électronique », restent hors de portée de la technologie OLAP faute d'outils et de méthodes adaptées à la gestion de données textuelles. Ne pas prendre en compte ces données mène inévitablement à l'omission d'informations pertinentes durant un important processus de prise de décision voire l'inclusion de données non pertinentes générant ainsi des analyses approximatives ou erronées (Tseng et Chou, 2006).

Récemment, le format XML[1] a fourni un vaste environnement permettant l'échange et la diffusion de documents au sein des systèmes d'information des entreprises ou bien sur le Web. Les données textuelles en un format XML sont désormais des sources envisageables pour les systèmes OLAP.

La problématique de cet article peut se résumer comme suit : lors de l'analyse de données issues de documents XML principalement constitués de texte, des indicateurs textuels sont employés. Mais, comment effectuer leur agrégation alors que les fonctions d'agrégation disponibles sont des fonctions arithmétiques (somme, moyenne…).

1.1 Contexte : état de l'art

Pour analyser des données issues de documents principalement constitués de données textuelles, plusieurs approches ont été proposées :

[1] XML, Extensible Markup Language, de http://www.w3.org/XML/.

Premièrement l'analyse multidimensionnelle de documents au sein d'un environnement OLAP. Les propositions de (McCabe *et al*., 2000), (Mothe *et al*., 2003), (Keith *et al*., 2005) et (Tseng et Chou, 2006) envisagent toutes d'analyser des documents dans un environnement multidimensionnel classique. Toutefois l'ensemble de ces propositions se limitent à des indicateurs numériques et ne permettent pas l'analyse du contenu des documents.

Première étape vers l'adaptation de l'agrégation de données au sein des systèmes OLAP, à l'instar des opérateurs CUBE (Gray *et al*., 1996), SKYLINE (Börzsönyi *et al*., 2001) ou encore OPAC (Messaoud *et al*., 2004), des propositions ont commencé par aborder des opérateurs adaptés aux données structurées via le format XML. C'est ainsi qu'un opérateur d'agrégation XML a été proposé (Wang *et al*., 2003) et (Wang *et al*., 2005) suivi très récemment par une adaptation de l'opérateur CUBE pour des donnée XML (Wiwatwattana *et al*., 2007). Ces nouveaux opérateurs permettent l'analyse de documents XML. Mais le contenu de ces documents XML n'est pas principalement constitué de texte, aussi ne permettent-ils pas l'analyse du contenu de documents XML principalement constitués de texte.

Afin de répondre plus précisément à la problématique d'analyse du contenu de documents XML principalement constitués de données textuelles, dans (Park *et al*., 2005), les auteurs proposent un ensemble de fonctions d'agrégation adaptées à ce type de données. Ces fonctions sont inspirées du domaine de la fouille de texte. Toutefois, ces fonctions ne sont ni détaillées, ni formalisées, ni implantées. Récemment, en s'inspirant des propositions de (Park *et al*., 2005), nous avons proposé une fonction d'agrégation qui permet d'effectuer une « pseudo-moyenne » à partir de plusieurs mots-clefs (Ravat *et al*., 2007a). Toutefois, cette fonction d'agrégation nécessite l'emploi d'une ontologie de domaine qui n'est pas nécessairement disponible. En outre, le principe d'agrégation résume un ensemble de mots-clefs par un ensemble plus général entraînant une perte de sémantique (bien que cette dernière soit paramétrable).

En conclusion, l'intégration des méthodes et outils d'analyse adaptés à des données textuelles issues de documents XML au sein de l'environnement OLAP n'en est qu'à l'âge de la préhistoire…

1.2 Objectifs et contributions

Dans le but de créer un environnement adapté à l'analyse de données textuelles issues de documents XML, nous poursuivons nos précédents travaux (Ravat *et al*., 2007a) et (Tournier, 2007). Nous envisageons de proposer une nouvelle méthode pour permettre l'agrégation de données issues de documents XML principalement constitués de données textuelles.

En nous basant sur les fonctions déjà existantes, nous avons proposé une fonction, AVG_KW, qui s'inspirait de la fonction d'agrégation classique de moyenne (Ravat *et al*., 2007a). Dans la même ligne directrice, nous proposons une nouvelle fonction d'agrégation inspirée cette fois de la fonction MAXIMUM_K qui retourne les K valeurs numériques les plus élevées. Cette fonction, TOP_KEYWORD_K (ou TOP_KW_K), restitue à l'utilisateur les K principaux mots-clefs d'un ensemble de mots-clefs à agréger.

Le reste de l'article est constitué comme suit : la section suivante (section 2) présente le modèle conceptuel sur lequel repose notre proposition et la section 3 expose la fonction qui permet d'ordonnancer les termes en fonction de leur représentativité dans les documents à agréger. Enfin la section 4 définit la fonction d'agrégation TOP_KW.

2 Modèle conceptuel

Les modèles existants sont limités pour l'analyse de données textuelles issues de documents XML (Ravat *et al*., 2007b). Pour des raisons de simplicité, nous proposons d'employer une modification du modèle en constellation (Kimball, 1996) déjà présenté dans (Ravat *et al*., 2007a). Toutefois, notez bien que ce que ce choix nous fait gagner en simplicité en terme de présentation du modèle, il nous le fait perdre sur la flexibilité et la représentation adaptée aux données issues de documents XML.

Bien que le modèle en constellation pourvu de mesures textuelles ne permette pas une représentativité maximale de données issues de documents XML, nous nous en contentons dans le présent article pour sa simplicité. Toutefois, pour une meilleure représentativité, l'emploi d'un modèle multidimensionnel mieux adapté est recommandé tel que le modèle en galaxie (Ravat *et al*., 2007b). Il est à noter que l'implantation de la fonction d'agrégation reste identique, que ce soit avec le modèle en constellation modifié ou bien avec le modèle en galaxie.

2.1 Définition formelle

Un schéma en constellation textuel est employé pour modéliser une analyse de contenus de documents où ce contenu est modélisé en tant que sujet d'analyse. Comme dans un schéma en constellation classique, un fait modélise un sujet d'analyse et une dimension modélise un axe d'analyse.

Un *schéma en constellation textuel CT* est défini par $CT = (F^{CT}, D^{CT}, Star^{CT})$ où :
- $F^{CT} = \{F_1,\ldots,F_m\}$ est un ensemble de faits ;
- $D^{CT} = \{D_1,\ldots,D_n\}$ est un ensemble de dimensions ;
- $Star^{CT} = F^{CT} \rightarrow 2^{D^{CT}}$ est une fonction associant chaque fait à ses dimensions associées[2]. Un *schéma en étoile textuel* est une constellation où F^{CT} est un singleton.

Un *fait F* est défini par $F = (M^F, I^F, IStar^F)$ où :
- $M^F = \{M_1,\ldots,M_n\}$ est un ensemble de *mesures* ;
- $I^F = \{i^F{}_1,\ldots,i^F{}_q\}$ est un ensemble d'instances du fait ;
- $IStar^F : I^F \rightarrow I^{D1} \times \ldots \times I^{Dn}$ est une fonction qui associe respectivement les instances du fait F aux instances des dimensions D_i liées.

Une *mesure M* est définie par $M = (m, F_{AGG})$ où :
- m est la mesure ;
- $F_{AGG} = \{f_1,..f_x\}$ est un ensemble de fonctions d'agrégation compatibles avec l'additivité de la mesure, $f_i \in$ {SUM, AVG, MAX…}.

Les mesures peuvent être additives, semi-additives ou non-additives (Kimball, 1996) et (Horner *et al*., 2004).

Une *dimension D* est définie par $D = (A^D, H^D, I^D)$ où :

[2] la notation 2^D représente l'ensemble des parties de l'ensemble D.

- $A^D = \{a^D_1,\ldots,a^D_u\}$ est un ensemble d'*attributs* (paramètres et attributs faibles) ;
- $H^D = \{H^D_1,\ldots,H^D_x\}$ est un ensemble de *hiérarchies* représentant l'agencement des attributs ;
- $I^D = \{i^D_1,\ldots,i^D_p\}$ est un ensemble d'instances de la dimension.

Une *hiérarchie H* est définie par $H = (Param^H, Weak^H)$ où :

- $Param^H = <p^H_1, p^H_2,\ldots, p^H_{np}, All>$ est un ensemble ordonné d'attributs, appelés *paramètres* (avec $\forall\ k\in[1..np]$, $p^H_k\in A^D$ et une racine commune $p^H_1=a^D_1$, $\forall H\in H^D$) ;
- $Weak^H : Param^H \rightarrow 2^{A^D - Param^H}$ est une application spécifiant l'association d'attributs faibles aux paramètres.

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

2.2 Les mesures dans l'environnement OLAP

Pour répondre aux spécificités des collections de documents, nous définissons une extension du concept classique de mesure.

2.2.1 Différents types de mesures

Nous distinguons ainsi deux types de mesures : les mesures numériques et les mesures textuelles.

Une *mesure numérique* est exclusivement composée de données numériques. Elle est soit *additive* (toutes les fonctions d'agrégation traditionnelles peuvent être employées) ; soit *semi-additives* et représente des instantanés (des températures, des quantités de stock…). Avec des mesures semi-additives, les fonctions d'agrégation sont limitées. F_{AGG} permet la spécification des fonctions d'agrégations compatibles avec la nature additive ou semi-additive de la mesure.

Une *mesure textuelle* est une mesure dont les données textuelles sont à la fois non numériques et non additives. Le contenu d'une mesure textuelle peut représenter un mot, un paquet de mots, un paragraphe voire un document complet. Nous distinguons plusieurs types de mesures textuelles :

- Une *mesure textuelle brute* est une mesure dont le contenu correspond au contenu complet d'un document ou bien d'un fragment de document (par exemple le contenu d'un article au format XML privé des balises XML qui le structurent).
- Une *mesure textuelle élaborée* est une mesure dont le contenu est issu d'une mesure textuelle brute et ayant subi un certain nombre de prétraitements. Une *mesure textuelle* de type *mot-clef* est une mesure textuelle élaborée. Ce type de mesure est obtenu, par exemple, après application de traitements sur une mesure textuelle brute tel que le retrait des mots vides et le maintien des mots les plus significatifs vis-à-vis du contexte du document.

2.2.2 Mesures et fonctions d'agrégation

L'environnement OLAP propose diverses fonctions d'agrégation basiques. Toutefois, en fonction du type de mesure, elles ne peuvent pas nécessairement toutes être employées. Le

Top_Keyword : fonction d'agrégation OLAP

Tableau TAB. 1 résume les combinaisons compatibles entre fonctions d'agrégation et types de mesure. L'environnement OLAP classique dispose des fonctions arithmétiques et génériques suivantes :

- Fonctions arithmétiques : SUM, AVG, MIN, MAX (retournant respectivement la somme, la moyenne, le minimum, le maximum d'un ensemble de valeurs).
- Fonctions génériques : COUNT, LIST (la fonction de comptage qui compte le nombre d'instances et LIST qui est la fonction identité n'agrégeant aucune valeur et retournant la liste des valeurs à agréger).

A ces fonctions, notre environnement propose les fonctions d'agrégation adaptées aux données textuelles suivantes :

- Fonctions textuelles : AVG_KW (Ravat *et al.*, 2007a) et TOP_KW (définie ci-après).

Type de mesure	Fonctions applicables	Exemple
Numérique, additive	Fonction arithmétiques et génériques	Une quantité d'articles
Numérique, semi-additive	Avg, Min, Max et génériques	Une temperature
Textuelle, brute	Top_Kw, génériques	Le contenu d'un article
Textuelle, mot-clef	Avg_Kw, génériques	Les mots-clefs d'un fragment de document

TAB. 1 *Les différents types de mesures et les différentes fonctions d'agrégation associées.*

2.3 Exemple

Pour observer les activités d'un institut de recherche, un décideur analyse les sujets dont traite une collection d'articles scientifiques publiés par des auteurs à une certaine date (cf. FIG. 2). Le sujet d'analyse, le fait *ARTICLES*, dispose de trois indicateurs d'analyse (mesures) : une mesure numérique (*Tx_Accept*, le taux d'acceptation correspondant à l'article), une mesure textuelle brute (*Texte*) et une mesure textuelle élaborée de type mot-clef (*Mots_Clefs*). Les notations graphiques sont inspirées de (Golfarelli *et al.*, 1998). Le fait *ARTICLES* est relié pour sa part aux dimensions *DATES* et *AUTEURS*.

Le système dispose des fonctions d'agrégation arithmétiques classiques (SUM, AVG, MIN et MAX), des fonctions d'agrégation générique (COUNT et LIST) ainsi que de deux fonctions d'agrégation textuelles (AVG_KW et TOP_KW).

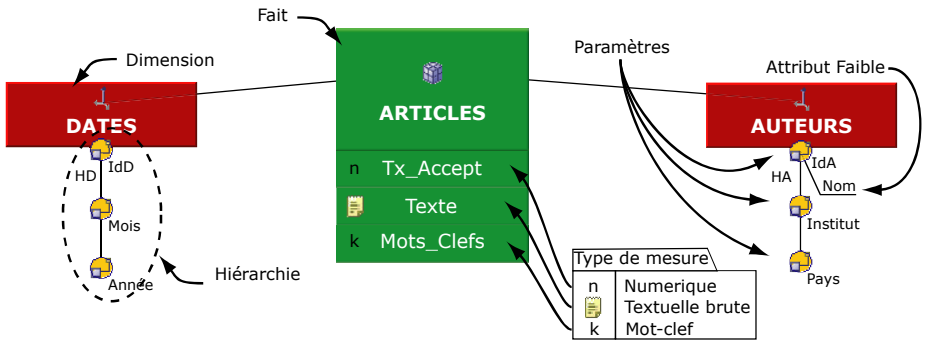

FIG. 2 – *Exemple de constellation textuelle permettant l'analyse de données textuelles.*

Formellement, le schéma en constellation textuel CT_1 présenté en FIG. 2 est le suivant :

- $CT1 = (F^{CT1}, D^{CT1}, Star^{CT1})$ avec $F^{CT1} = \{ARTICLES\}$, $D^{CT1} = \{DATES, AUTEURS\}$ et $Star^{CT1} = \{ARTICLES \rightarrow \{DATES, AUTEURS\}\}$.
- $ARTICLES = (M^{ARTICLES}, I^{ARTICLES}, IStar^{ARTICLES})$ avec $M^{ARTICLES} = \{Tx\_Acccept, Texte, Mots\_Clefs\}$, $I^{ARTICLES}$ est la liste des instances du fait *ARTICLES* et $IStar^{ARTICLES} : I^{ARTICLES} \rightarrow I^{DATES} \times I^{AUTEURS}$. Avec les mesures suivantes :
  - $Tx\_Acccept = (tx\_accept, \{AVG, MIN, MAX\})$
  - $Texte = (texte, \{$TOP_KW$\})$
  - $Mots_Clefs = (mots_clefs, \{AVG_KW\})$
- $DATES = (A^{DATES}, H^{DATES}, I^{DATES})$ avec $A^{DATES} = \{IdD, Mois, Année\}$, $H^{DATES} = \{HD\}$ et I^{DATES} est la liste des instances de la dimension *DATES*. Avec la hiérarchie suivante :
  - $HD = (Param^{HD}, Weak^{HD})$ avec $Param^{HD} = <IdD, Mois, Année>$ et $Weak^{HD} = \{\}$.
- $AUTEURS = (A^{AUTEURS}, H^{AUTEURS}, I^{AUTEURS})$ avec $A^{AUTEURS} = \{IdA, Nom, Institut, Pays\}$, $H^{AUTEURS} = \{HA\}$ et $I^{AUTEURS}$ est la liste des instances de la dimension *AUTEURS*. Avec la hiérarchie suivante :
  - $HA = (Param^{HA}, Weak^{HA})$ avec $Param^{HA} = <IdA, Institut, Pays>$ et $Weak^{HA} = \{IdA \rightarrow \{Nom\}\}$.

Notez que les fonctions d'agrégation génériques (COUNT et LIST) ne sont pas spécifiées dans les ensembles de fonctions d'agrégation compatibles avec l'additivité des mesures.

3 Ordonnancement de termes d'un fragment de texte

Le principe de la fonction d'agrégation TOP_KW est simple : il s'agit, à l'instar de la fonction MAX_K qui retourne les *k* plus grands nombres d'un ensemble de nombres à agréger, de fournir les *k* mots les plus représentatifs d'un fragment de texte. Pour ce faire, il est nécessaire d'ordonner les mots qui composent un document en fonction de leur représentativité vis-à-vis d'un ensemble de documents.

L'évaluation de la représentativité d'un terme au sein du bloc de texte qui le contient est une problématique bien connue de la recherche d'information (Baeza-Yates et Ribeiro-Neto, 1999). A l'instar des travaux de recherche de ce domaine, nous emploierons une fonction pour pondérer les termes en fonction de leur représentativité et ainsi les ordonner en fonction de cette représentativité.

3.1 Calcul d'un poids de « représentativité » : fonction *tf.idf*

Pour notre problématique nous employons une fonction qui permet d'assigner un poids à chaque terme en fonction de son contexte. Notre choix s'est porté sur l'une des plus simples d'entre elles: *tf.idf*. Cette fonction est le produit entre la représentativité d'un terme dans un document (tf : term frequency) avec l'inverse de sa représentativité dans l'ensemble des documents disponibles (idf : inverse document frequency).

$$tf(t) = \frac{n(t)}{\sum_{frag} n} \text{ et } idf(t) = \log \frac{(nb\_doc)}{(nb\_doc(t))} \quad \text{Eq. 1}$$

$$tf(t) \times idf(t) \quad \text{Eq. 2}$$

Dans l'équation précédente, *tf*(*t*) est le nombre de fois où le terme *t* est présent dans un fragment de texte normalisé par rapport au nombre total de termes contenus dans le fragment (ceci permet de réduire le biais introduit par de très long fragments par rapport à des fragments très courts).

La quantité *idf*(*t*) est l'inverse du nombre de documents contenant le terme *t* par rapport au nombre de documents disponibles. L'emploi d'une fonction logarithme est utile pour aplanir la courbe et ainsi réduire les conséquences de grandes valeurs.

Pour plus de détails, nous renvoyons le lecteur sur (Robertson, 2004) pour une étude détaillée et récente de la fonction.

3.2 Adaptation au contexte décisionnel

En recherche d'information, il est nécessaire de connaître la représentativité d'un terme par rapport à l'ensemble des documents qui le contient. Dans notre cas, les termes n'ont pas à être les plus représentatifs vis-à-vis de l'ensemble de la collection mais vis-à-vis de l'ensemble des fragments à agréger via la fonction.

Aussi, au sein de l'*idf*, les valeurs *nb_doc* (le nombre total de documents de la collection) et *nb_doc*(*t*) (le nombre de documents de la collection qui contiennent le terme *t*) sont adaptées à notre contexte. Il ne s'agit plus d'un nombre par rapport à la collection, mais par rapport au groupe de documents à agréger par la fonction.

Ainsi dans notre approche :

- *nb_doc* est le nombre de fragments de documents à agréger par la fonction ;
- *nb_doc*(*t*) est le nombre de fragments de documents à agréger contenant le terme *t*.

Notez que pour chaque cellule d'une table multidimensionnelle, la fonction d'agrégation est appliquée. Chaque cellule représente un certain nombre de documents ou de fragments de documents. L'ensemble des documents sur lequel travaille la fonction n'est donc pas la collection entière (comme en recherche d'information), mais c'est plutôt l'ensemble des documents (ou fragments de documents) de chaque cellule. Plus de précisions sont fournies dans la section suivante

3.3 Application à l'interface de restitution

La restitution d'une analyse se fait par l'intermédiaire d'une table multidimensionnelle (cf FIG. 3). Les valeurs disposées en lignes et en colonnes produisent des cellules c_{ij}, correspondant au croisement entre la i^{ème} ligne et la j^{ème} colonne. Chaque cellule contient les valeurs agrégées des indicateurs analysés.

À chaque cellule c_{ij}, correspond :

- un ensemble de documents D_{ij} (des articles scientifiques dans notre cas) ;

- un nombre total de documents d_{ij} ;
- un nombre total de termes n_{ij} contenu dans les d_{ij} documents (les mots vides de sens ne sont pas pris en compte).

De plus, à chaque cellule c_{ij}, pour chaque terme t correspond :

- un nombre d'occurrences de t : $n_{ij}(t)$ dans les documents de c_{ij} ;
- un nombre de documents $d_{ij}(t)$ qui contiennent t parmi ceux de c_{ij} ($d_{ij}(t) \leq d_{ij}$).

Ainsi, nous obtenons la formule suivante pour chaque terme contenu dans la cellule c_{ij} :

$$tf.idf(t) = \frac{n_{ij}(t)}{n_{ij}} \times \log \frac{d_{ij} + 1}{d_{ij}(t)} \qquad \text{Eq. 3}$$

Notez que dans certains cas, $d_{ij} = d_{ij}(t)$, c'est-à-dire que le terme t est contenu dans l'ensemble des documents. Le log est alors nul. Afin d'éviter ce cas et d'avoir une pondération du terme t nulle, 1 est ajouté à d_{ij}.

La formule, appliquée à un groupe de documents dont l'ensemble des termes a été extrait, permet l'obtention d'une liste ordonnée de termes. À chaque cellule c_{ij} correspond un ensemble de documents D_{ij}. Les termes sont extraits de chacun des documents de D_{ij}. Puis, ils sont ordonnés dans une liste par application de la formule *tf.idf* adaptée (cf. Eq. 3) : $L_{ij} = < t_1, \ldots, t_n >$. Sur la liste L_{ij} est appliquée la fonction d'agrégation TOP_KW.

4 Fonction d'agrégation : TOP_KW

A partir de l'ordonnancement des termes obtenus par l'application de la fonction *tf.idf* légèrement modifiée, nous définissons une fonction d'agrégation qui permet de restituer les k principaux termes.

4.1 Spécification formelle de la fonction

A partir des documents, un ensemble ordonné est créé par l'application de Eq. 3. Le résultat de l'ordonnancement est injecté en entrée à la fonction TOP_KW_K. Cette dernière agrège un ensemble de n termes (des mots-clefs) en un sous ensemble de k termes les plus représentatifs vis-à-vis des n termes. T représente l'ensemble des termes des documents.

$$\begin{array}{rccc} Top\_Kw_k : & T^n & \longrightarrow & T^k \\ & \langle t_1, \ldots, t_n \rangle & \mapsto & (t_1, \ldots, t_k) \end{array} \qquad \text{Eq. 4}$$

Avec :

- en entrée : une liste ordonnée de n termes $<t_1,\ldots,t_n> \mid p(t_1) \geq \ldots \geq p(t_n)$, telle que les termes sont classés par ordre de poids décroissant.
- en sortie les k premiers termes de la liste dont les poids sont les plus élevés.

Toutefois, entre les données textuelles représentant le contenu d'un document et la liste ordonnée de termes certains prétraitements sont effectués.

4.2 Prétraitements

Afin de permettre l'exécution de la fonction d'agrégation TOP_KW, une mesure textuelle brute est traitée afin de permettre d'éliminer des éléments susceptibles de parasiter les résultats. Lors du calcul des poids par un processus d'indexation, similaire en tout point à ceux employés en recherche d'information (Baeza-Yates et Ribeiro-Neto, 1999), il est nécessaire d'éliminer tous les termes susceptibles de biaiser le calcul des poids.

Il s'agit principalement du retrait des mots vides de sens (articles, prépositions, pronoms…). Ce retrait s'effectue à partir de listes telles que celles de l'université de Glasgow[3]. Toutefois, il faut noter que la fonction *tf.idf* pénalise naturellement les mots très voire trop représentés dans l'ensemble des documents. Mais le retrait est tout de même nécessaire afin d'éviter les parasitages. Le retrait est effectué en une passe sur l'ensemble des documents (en ignorant les balises XML), pour chaque bloc de texte (par exemple, pour chaque paragraphe), un nouveau bloc de texte privé des mots vides.

Une autre source de parasitage est l'emploi hors contexte de certains termes. Ceci peut être limité en appliquant un filtre très limitatif aux termes retenus. Il est possible d'employer une ontologie de domaine et de ne conserver que les termes représentatifs du domaine (les termes présents dans l'ontologie). Toutefois, cette méthode n'a pas été envisagée car elle n'est pas robuste. Par exemple, en prenant une ontologie avec les termes des systèmes d'aide à la prise de décision, le terme « *fait* » serait retenu dans la phrase suivante « *en s'appuyant sur le fait que*... » alors qu'il a un signification différente de celle de l'ontologie (un sujet d'analyse).

Remarque importante : une méthode complémentaire au retrait de mots vides est la lemmatisation. Il s'agit de remplacer les termes par leur forme canonique. Avec par exemple, l'élimination des pluriels par la suppression de tous les « s » terminaux. Ou encore le remplacement de tous les verbes par leur infinitif avec, entre autres, la suppression des terminaisons en « …ing » pour l'anglais. Pour éviter d'introduire trop de biais nous avons aussi renoncé à employer ce procédé. En effet selon les études effectuées en fouille de texte (Stavrianou *et al*., 2007), la lemmatisation, bien que très efficace en recherche d'information, l'est beaucoup moins pour l'analyse de texte.

4.3 Exemple

Soit l'analyse des 2 principaux termes ($k = 2$) d'articles scientifiques en fonction de l'auteur et de l'année de publication (cf. FIG. 3). Cette analyse conduit à l'application de la fonction d'agrégation TOP_KW_2 sur quatre regroupement d'articles : les quatre cellules $\{c_{11}, c_{12}, c_{21}, c_{22}\}$ de la table multidimensionnelle correspondant au couples (*Au1*, *2005*), (*Au1*, *2006*), (*Au2*, *2005*) et (*Au2*, *2006*). Au final, la table multidimensionnelle est constituée de quatre cellules, chacune contenant les 2 termes les plus représentatifs de l'ensemble de documents agrégés pour chacune d'elle.

Il est à noter que la spécification de la valeur de k dépend principalement de la quantité d'informations analysées. Les résultats étant affichés dans une table multidimensionnelle, cette dernière a tendance à être surchargée si trop d'informations y sont restituées. Ainsi k dépends du nombre de cellules de la table. Par exemple, dans une table n'affichant que quatre cellules (cf. FIG. 3), il est possible de spécifier $k = 10$, car au final l'utilisateur ne lit

[3] Stop word list, de http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words

« que » 40 termes (10 par cellule). Dans une table avec beaucoup plus de cellules, *k* sera revu à la baisse et probablement inférieur à 5.

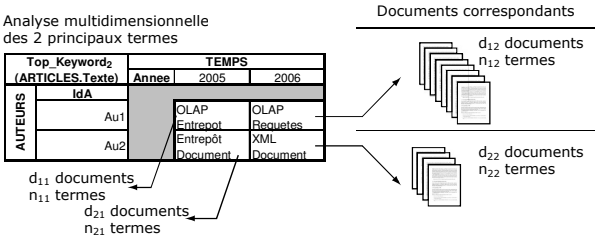

FIG. 3 – *Exemple d'analyse employant la fonction TOP_KW2.*

5 Conclusion

Dans cet article, nous avons présenté une nouvelle fonction d'agrégation pour un environnement OLAP adapté à l'analyse de données extraites de documents XML principalement constitués de données textuelles. Cette fonction d'agrégation permet d'obtenir une vision synthétique d'un ensemble de documents en sélectionnant les *k* mots-clefs les plus représentatifs. La fonction d'agrégation TOP_KEYWORD$_K$ (ou TOP_KW$_K$) s'appuie sur la fonction de pondération *tf.idf*. Cette fonction de pondération permet d'ordonnancer les mots-clefs d'un ensemble de documents ou de fragments de documents. Ainsi la fonction d'agrégation sélectionne les *k* premiers mots-clefs.

Nous implantons actuellement la fonction d'agrégation au sein d'un environnement OLAP basé sur une base de données multidimensionnelle implantée à la fois en ROLAP (OLAP Relationnel) et en XML au sein du SGBD Oracle 10*g*2. L'ensemble est piloté à partir d'une interface client en Java (jdk 1.6).

Plusieurs perspectives sont envisageables. Premièrement, comme en recherche d'information, l'un des inconvénients majeurs de l'emploi d'une fonction de pondération est la nécessité de disposer de fichiers inverses et d'indexes qui permettent un accès rapide aux poids pour permettre d'effectuer les calculs rapidement. Aussi il est nécessaire d'envisager un processus de matérialisation de vues pour accélérer le traitement de la fonction d'agrégation. Deuxièmement, en recherche d'information, il existe la notion de réinjection de pertinence qui consiste à ajouter des termes à une requête afin d'accroître l'effet des termes employés. De manière similaire, nous envisageons d'employer la réinjection de pertinence pour ajouter des termes à ceux que retourne la fonction d'agrégation. Ainsi le résultat final sera un ensemble de termes plus précis. Troisièmement, il existe de nombreuses variantes voire des alternatives complètes à la fonction *tf.idf* (Robertson, 2004), nous envisageons une étude comparative de ces différentes fonctions afin d'optimiser l'implantation de la fonction d'agrégation. Enfin, afin d'enrichir notre proposition d'un environnement OLAP permettant l'analyse de données issues de documents XML principalement constitués de données textuelles, nous envisageons de définir et d'implanter les autres fonctions d'agrégation proposées dans (Park *et al*., 2005).

Top_Keyword : fonction d'agrégation OLAP

Références

Baeza-Yates, R., et B. Ribeiro-Neto (1999). *Modern Information Retrieval*. Addison Wesley.

Börzsönyi, S., D. Kossmann et K. Stocker, (2001). The Skyline Operator. Dans *17th Intl. Conf. on Data Engineering* (*ICDE*), IEEE Computer Society, p. 421–430.

Colliat, G. (1996). OLAP, Relational and Multidimensional database systems. *SIGMOD Record*, vol.25(3), ACM Press, p. 64–69.

Golfarelli, M., D. Maio, S. Rizzi (1998). The Dimensional Fact Model: A Conceptual Model for Data Warehouses. invited paper, *Intl. Journal of Cooperative Information Systems* (*IJCIS*), vol.7(2-3), World Scientific Publishing, p. 215–247.

Gray, J., A. Bosworth, A. Layman, et H. Pirahesh (1996). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. *12th Int. Conf. on Data Engineering* (*ICDE*), IEEE Computer Society, p. 152–159.

Horner, J., I.-Y. Song, et P.P. Chen (2004). An analysis of additivity in OLAP systems. *7th ACM Int. Workshop on Data Warehousing and OLAP* (*DOLAP 2004*), ACM Press, p. 83–91.

Keith, S., O. Kaser, et D. Lemire (2005). Analyzing Large Collections of Electronic Text Using OLAP. *APICS 29th Conf. in Mathematics, Statistics and Computer Science*, Acadia University, p. 17–26.

Kimball, R. (1996). *The data warehouse toolkit*. John Wiley and Sons (2nd ed. 2003).

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

Messaoud, R.B., O. Boussaid et S. Rabaséda (2004). A new OLAP aggregation based on the AHC technique. *7th ACM Intl. Workshop on Data Warehousing and OLAP* (*DOLAP*), ACM Press, p. 65–72.

Mothe J., C. Chrisment, B. Dousset, et J. Alau (2003). DocCube: Multi-dimensional visualisation and exploration of large document sets. *Journal of the American Society for Information Science and Technology* (*JASIST*), vol.54(7), Wiley Periodicals, p. 650–659.

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

Ravat, F., O. Teste et R. Tournier (2007). OLAP Aggregation Function for Textual Data Warehouse. *International Conference on Enterprise Information Systems* (*ICEIS 2007*), Funchal, Madeira - Portugal, 12-17 juin 2007, Vol. DISI, INSTICC Press, p. 151–156.

Ravat, F., O. Teste, R. Tournier et G. Zurfluh (2007). A Conceptual Model for Multidimensional Analysis of Documents. *26th Intl. Conf. on Conceptual Modeling* (*ER*), LNCS 4801, Springer, p. 550–565.

Robertson, S. (2004). Understainding Inverse Document Frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5), Emerald Publishing Group, p. 503–520.

Stavrianou, A., P. Andritsos et N. Nicoloyannis (2007). Overview and Semantic Issues of Text Mining. *SIGMOD Record*, 36(3), ACM Press, p.23–34.

Sullivan D. (2001). *Document Warehousing and Text Mining*. Wiley John & Sons, 2001.

Tournier, R. (2007). Analyse en ligne (OLAP) de documents. Thèse de doctorat, Université Toulouse 3, Paul Sabatier.

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

Wang, H., J. Li, Z. He et H. Gao, (2003). Xaggregation: Flexible Aggregation of XML Data. *4th Intl. Conf. on Advances in Web-Age Information Management* (*WAIM*), LNCS 2762, Springer, p. 104–115.

Wang, H., J. Li, Z. He et H. Gao (2005). OLAP for XML Data. *5th Intl. Conf. on Computer and Information Technology* (*CIT*), IEEE Computer Society, p. 233–237.

Wiwatwattana, N., H.V. Jagadish, L.V.S. Lakshmanan et D. Srivastava (2007). X^3: A Cube Operator for XML OLAP. *23rd Intl. Conf. on Data Engineering* (*ICDE*), IEEE Computer Society, p. 916–925.

Summary

For more than a decade, research on OLAP and multidimensional databases has generated methodologies, tools and resources management systems for the analysis of numeric data. With the growing availability of digital documents, there is a need for incorporating such XML text-rich documents within multidimensional databases as well as an adapted framework for their analysis. This paper presents a new aggregation function that allows aggregating textual data in an OLAP environment as traditional arithmetic functions would do on numeric data. The TOP_KEYWORD function (or TOP_KW for short) summarises a set of documents by their most significant terms, using a weighing function from information retrieval: *tf.idf*.