TLabel: Nouvel opérateur d'agrégation par catégorisation dans les cubes de textes

Lamia Oukid*, Omar Boussaid**,
Nadjia Benblidia*, Fadila Bentayeb**

*Université de Blida 1 (Laboratoire LRDSI)
B.P. 270, Route de Soumaa; 09000 Blida, Algérie.
o.lamia@hotmail.fr, benblidia@yahoo.com
**Université de Lyon (ERIC, Lyon 2)
5, avenue Pierre Mondès France 69676 Bron Cedex, France.
{omar.boussaid, fadila.bentayeb}@univ-lyon2.fr

Résumé. L'analyse en ligne (OLAP) dans les cubes de textes nécessite la définition de nouveaux types d'opérateurs d'analyse appropriés aux données textuelles. En effet, les opérateurs d'agrégation classiques ont montré leur efficacité pour l'analyse en ligne des données numériques, mais ils sont inadaptés pour l'analyse des données textuelles. Dans cet article, nous proposons un nouvel opérateur d'agrégation par catégorisation nommé *TLabel* (*Text Label*) permettant d'agréger les données textuelles en plusieurs classes de documents. A chaque classe sera associée une étiquette (*Label*) qui représente le contenu sémantique des données textuelles de la classe grâce à une adaptation des techniques de fouille de textes à l'OLAP. Nous avons effectué une étude expérimentale sur notre opérateur *TLabel*. Les résultats préliminaires montrent l'intérêt de notre approche pour l'analyse en ligne des données textuelles.

1 Introduction

Les technologies d'entreposage de données et d'analyse en ligne (OLAP) ont largement fait leurs preuves pour l'analyse en ligne des données numériques. Néanmoins, une grande partie des données circulant dans les entreprises sont présentées sous forme de données textuelles (rapports, e-mails, etc.). Ces dernières restent peu exploitées par les systèmes décisionnels actuels. Permettre la prise en compte de ce type de données par les systèmes OLAP revient à définir de nouvelles techniques permettant d'intégrer la sémantique des données textuelles dans le processus d'analyse en ligne. Un des principaux challenges dans ce contexte est l'agrégation de données textuelles. En effet, l'agrégation des données numériques s'effectue à l'aide de fonctions d'agrégation classiques (somme, moyenne, min, max, etc.) qui ne sont pas adaptées aux données textuelles. La nature peu ou pas structurée de ces dernières les rend difficiles à analyser. Pour analyser en ligne des données textuelles, il est donc nécessaire de faire évoluer les cubes de données classiques vers des cubes de textes tout en proposant de nouveaux opérateurs permettant l'agrégation de ces données.

Les systèmes OLAP permettent une analyse navigationnelle à travers des cubes OLAP. Ils offrent la possibilité de passer d'une vue à une autre de manière interactive. Ils ont également montré leur efficacité pour l'analyse de gros volumes de données. L'analyse en ligne de données permet d'exprimer des requêtes complexes et de visualiser des résultats agrégés pertinents pour la prise de décision (Oukid et al., 2013a). D'autre part, la fouille de textes (FT) cherche à extraire des informations pertinentes à partir de grandes collections de documents par des techniques d'analyse appropriées (Felman et Sanger, 2007). Parmi les techniques de FT, nous retrouvons la classification automatique des documents. Cette dernière permet la création de groupes de documents qui sont rassemblés selon des relations de similarité qui existent entre eux. Nous pensons que ces techniques peuvent être exploitées afin de définir de nouvelles techniques d'analyse en ligne plus élaborées qui soient adaptées au texte.

L'analyse en ligne de données textuelles permet d'aller au delà d'une simple extraction de connaissances à partir des documents textuels. Elle doit permettre d'aller vers des analyses navigationnelles à travers les cubes de textes. Dans ce contexte, nous nous inspirons des techniques de FT afin de définir de nouveaux opérateurs OLAP, notamment des opérateurs d'agrégation, adaptés aux données textuelles.

Dans cet article, nous proposons un nouvel opérateur d'agrégation par catégorisation de textes nommé *TLabel* (*Text Label*), basé sur un couplage entre l'OLAP et la fouille de textes. En FT, la catégorisation de textes consiste à attribuer des catégories (sujets, thèmes) a une collection de documents textuels. Notre approche d'agrégation par catégorisation consiste à agréger le contenu sémantique des données textuelles dans les cubes de textes en groupes homogènes de documents étiquetés. Notre approche comprend deux étapes : La première consiste à agréger les données textuelles sous forme de groupes homogènes de documents. Pour ce faire, nous proposons une adaptation de l'algorithme *K-means* à l'OLAP nommée *OCluster* (*OLAP-Cluster*). Dans cette adaptation, nous nous basons sur une fonction de classement de documents développée dans nos travaux récents (Oukid et al., 2013a) et (Oukid et al., 2013b), qui permet de calculer la similarité entre documents dans les cubes de textes, par le biais d'une adaptation du modèle vectoriel (Salton et al., 1975) à l'analyse en ligne. La deuxième étape consiste en l'attribution d'une étiquette (*Label*) à chaque classe de documents obtenue. Cette étiquette agrège le contenu sémantique des documents d'une même classe. Pour calculer cette dernière, nous proposons une démarche qui consiste à calculer dans un premier temps un document représentant les données textuelles d'une même classe dénommé *DResume* (*Document Resume*). Par la suite, *DResume* est projeté sur une hiérarchie de concepts afin de définir l'étiquette de la classe, cette dernière est représentée par le concept le plus dominant dans *DResume*.

La suite de cet article est organisée comme suit. La section 2 présente quelques travaux pertinents de l'état de l'art sur l'analyse en ligne des données textuelles. La section 3 rappelle notre modèle de données *CXT-Cube* proposé dans un travail précédent, sur lequel est basé notre opérateur d'agrégation. La section 4 illustre notre opérateur d'agrégation par catégorisation *TLabel*. La section 5 présente quelques résultats expérimentaux sur notre opérateur *TLabel*. Enfin la section 6 conclut l'article et présente quelques perspectives.

2 État de l'art

Pour analyser les données textuelles, la tendance actuelle est de faire appel aux différents domaines qui s'intéressent aux traitements de textes telles que la recherche d'information (RI) et la FT. Dans cette section, nous présentons les principaux travaux qui proposent des approches d'analyse en ligne de données textuelles. Nous regroupons les travaux en deux familles. La première englobe les approches d'analyse en ligne des données textuelles (Text OLAP) basées sur des techniques issues de la RI. La deuxième famille regroupe les approches basées sur des techniques de la FT.

2.1 Approches Text OLAP basées sur des techniques de RI

Parmi les travaux qui utilisent des techniques issues de la RI pour définir des techniques d'analyse en ligne de données textuelles, nous citons les travaux de (Lin et al., 2008) qui proposent une nouvelle dimension à travers leur cube de données nommé *Text Cube* ; cette dimension comporte une hiérarchie de mots-clefs. Deux mesures sont proposées : la fréquence des termes (*term frequency Tf*) et l'index inversé (*inverted index IV*) (Salton et al., 1975). (Bringay et al., 2011) ont développé une fonction d'agrégation, qui recherche les groupes de mots les plus significatifs dans des tweets à travers une adaptation de la mesure *Tf-Idf*. Cette mesure permet la prise en compte des informations hiérarchiques dans les mesures. Les mots représentatifs des tweets sont calculés par rapport au niveau de granularité souhaité. (Pérez et al., 2007) proposent une combinaison entre entrepôts de données classiques et entrepôts de documents. Cette combinaison présente comme résultat, un cube contextualisé appelé *R-Cube*. Deux nouvelles dimensions sont proposées : (1) *Contexte*, comportant des fragments de textes jugés comme étant les plus pertinents vis-à-vis d'un contexte d'analyse et (2) *Pertinence*, contenant une valeur numérique, qui représente l'importance de chaque fait par rapport au contexte d'analyse. (Ravat et al., 2008) ont défini une fonction d'agrégation nommée *TOP-KWk*, qui retourne une liste des k mots-clefs ayant les plus grands poids dans une collection de documents. Ces mots-clefs sont pondérés par la méthode *Tf-Idf*.

2.2 Approches Text OLAP basées sur des techniques de FT

Dans cette famille de travaux, nous retrouvons (Zhang et al., 2011) qui ont défini un modèle de cube de textes appelé *MiTexCube*. Ce modèle permet une représentation compressée des cellules textuelles en micro-clusters dans une base de données multidimensionnelles. Chaque micro-cluster est représenté par un vecteur centroide de termes pondérés par la méthode *Tf-Idf* et sa taille. (Zhang et al., 2009) agrègent les données textuelles en plusieurs niveaux hiérarchiques de thèmes (Topics) en utilisent le modèle PLSA (*Probabilistic Latent Semantic Analysis*). Dans ce modèle nommé *Topic Cube*, deux mesures sont proposées : la distribution des mots d'un thème dans le document (*word distribution of a topic*) et la couverture du thème par le document (*topic coverage by documents*). (Mothe et al., 2003) proposent *DocCube*, un cube de textes pour l'exploration et la visualisation de documents basé sur la classification. La table de faits comprend un lien pondéré qui permet d'associer un fait à un document. le poids d'un lien obtenu en appliquant la méthode *Vector Voting* indique le degré de confiance de l'association d'un document avec les dimensions. Dans (Cody et al., 2002), Les auteurs

proposent un framework pour l'analyse de textes appelé *eClassifier*. A travers ce framework, les auteurs exposent une classification de documents et une attribution d'étiquettes aux classes de documents. Les étiquettes des classes sont définies par les termes couvrants comme suit : (1) Si un terme apparait dans au moins 90% des documents de la classe, cette dernière est étiquetée par ce terme. (2) Si plus d'un terme apparaît dans au moins 90% des documents de la classe, alors la classe est étiquetée pas l'ensemble de ces termes avec une limite égale à quatre. (3) Si aucun terme ne couvre au moins 90% des documents de la classe, alors la classe est étiquetée par le terme ayant la plus grande fréquence d'apparition suivi du terme ayant la plus grande fréquence d'apparition dans tous les documents, à condition que la combinaison de ces derniers couvre au moins 90% des documents de la classe. Sinon, le processus est répété. (4) Si aucun des top-4 termes n'est dans au moins 10% des documents de la classe, cette dernière est étiquetée « divers ».

2.3 Discussion et positionnement

Nous constatons que le couplage entre l'OLAP et les différentes techniques de recherche d'information et/ou de fouille de textes donne de bons résultats. La plupart des approches proposées utilisent *Tf-Idf* comme mesure. Or, cette dernière ne permet pas une réelle prise en compte de la sémantique des données textuelles. Par exemple, dans le cas de l'analyse de Curriculum Vitae (CVs), il est plus intéressant d'extraire les informations sur les compétences dans les documents plutôt que d'extraire les termes les plus fréquents. De plus, l'extraction des thématiques (*Topics*) en utilisant PLSA ou LDA comme dans (Zhang et al., 2009) ne donne pas toujours de bons résultats. (Pérez et al., 2007) considère le contexte des documents dans leurs modèle *R-Cube*. La dimension *Context* définie dans ce modèle est représentée par les fragments de textes les plus pertinents pour une requête d'analyse. Cependant, l'extraction de ces fragments de textes se fait en utilisant la mesure *Tf-Idf*. Cette dernière est insuffisante pour représenter les données en relation avec le contexte d'analyse. Cody et al. (2002) proposent un framwork pour l'analyse de données textuelles en utilisant la classification de documents. Mais ce dernier se base sur un cube de données classique qui reste limité en terme d'analyse de données textuelles. Nous constatons également que la plupart des travaux ne proposent pas des opérateurs d'analyse en ligne pour leurs cubes de textes.

Dans nos travaux, nous nous inspirons des techniques issues de la RI et de la FT afin de définir des techniques d'analyse en ligne adaptées aux données textuelles. Notre cube de textes *CXT-Cube* (*Contextual Text Cube model*) (Oukid et al., 2013b) est basé sur l'exploitation des informations contextuelles qui entourent les documents. Nous avons proposé un nouveau concept : les *dimensions contextuelles*. Ces dernières permettent d'observer les données textuelles selon différents facteurs contextuels. Notre cube de textes est associé à une mesure d'analyse textuelle permettant une meilleure prise en compte de la sémantique des données textuelles que celles basées sur la mesure statistique *Tf-Idf*. Cette mesure textuelle s'appuie sur les informations contextuelles des dimensions et une méthode de propagation de pertinence. Afin de permettre des analyses en ligne sur les cubes de textes, nous proposons dans cet article un opérateur d'agrégation inspiré des techniques de FT. Contrairement à Cody et al. (2002), l'étiquetage des classes de documents dans notre approche est effectué en se basant sur une hiérarchie de concepts, ce qui donne la possibilité d'effectuer des analyses navigationnelles sur les cubes de textes (*Drill-Down*, *Roll-Up*).

3 Background

Afin de permettre des analyses pertinentes sur les données textuelles, il est essentiel de faire évoluer les cubes de données classiques afin de supporter de nouveaux types d'analyse. Avant de présenter notre opérateur d'agrégation par catégorisation *TLabel*, nous présentons dans cette section une vue d'ensemble sur notre modèle de données.

Modèle de données CXT-Cube : Notre cube de textes contextuel, baptisé *CXT-Cube* (*Contextual Text Cube model*) (Oukid et al., 2013b) est une extension du cube de données classique afin de supporter de nouveaux types d'analyse sur les données textuelles. *CXT-Cube* comprend de nouveaux types de dimensions appelées *dimensions contextuelles* adaptées aux données textuelles. Chaque dimension correspond à un paramètre contextuel lié aux données textuelles. Notre modèle comprend également une nouvelle mesure d'analyse textuelle basée sur une adaptation du modèle vectoriel issu de la RI à l'analyse en ligne.

Dimensions contextuelles : Les dimensions contextuelles comprennent deux types :

- *Les dimensions sémantiques :* une dimension sémantique est extraite à partir d'une ontologie de domaine liée à la dimension. Chaque dimension sémantique correspond à une hiérarchie de concepts.
- *Les dimensions méta-données :* une dimension de type méta-donnée est définie pour représenter un type de méta-donnée lié aux données textuelles. Par exemple, pour la méta-donnée auteur d'un document, nous pouvons créer la dimension AUTEUR correspondante à ce facteur contextuel.

Mesure d'analyse textuelle : *CXT-Cube* comprend une nouvelle mesure d'analyse textuelle permettant la prise en charge des informations contextuelles des données textuelles. Notre mesure d'analyse est basée sur une représentation en modèle vectoriel des données textuelles.

Une mesure d'analyse textuelle M représente chaque document d par plusieurs vecteurs de concepts pondérés, un pour chaque dimension Dim_r du *CXT-Cube*.

$$M =< \overrightarrow{d_{Dim_1}}, \overrightarrow{d_{Dim_2}}, ..., \overrightarrow{d_{Dim_*}} >$$

où $\overrightarrow{d_{Dim_r}} =< w_{c_1}, w_{c_2}, ..., w_{c_n} >$ est le vecteur de concepts pondérés d'un document d dans un espace vectoriel spécifique à une dimension Dim_r ; $c_1, c_2, ..., c_n$ sont les concepts de la hiérarchie de concepts spécifique à Dim_r et w_{c_i} est le poids attribué au concept c_i.

Calcul de la mesure d'analyse : Le calcul des poids des concepts dans notre mesure d'analyse textuelle est basé à la fois sur leurs fréquences d'apparition dans le document et une méthode de propagation de pertinence (Oukid et al., 2013a) (Oukid et al., 2013b). Cette dernière permet une réattribution de scores à travers une hiérarchie de concepts pour une meilleure prise en compte de la sémantique des données textuelles. Par exemple, si les concepts *Java* et *Php* existent dans un document, nous pouvons conclure que ce dernier présente une compétence en *Programmation*. Ainsi, le poids du concept *Programmation* augmente grâce à la propagation de pertinence à travers la hiérarchie de concepts. La propagation de pertinence permet également la prise en compte de nouveaux concepts. Dans notre exemple, le concept *Programmation* aura un poids différent de zéro, même si ce dernier ne figure pas dans le document.

Dans ce qui suit, nous présentons un nouvel opérateur d'agrégation de données textuelles adapté à notre modèle *CXT-Cube*.

4 TLabel : Opérateur d'agrégation par catégorisation

La fouille de textes est définie comme un processus d'analyse exploratoire qui permet d'extraire des connaissances des données textuelles par des outils d'analyse (Felman et Sanger, 2007). Dans ce contexte, nous nous intéressons aux méthodes de classification automatique de documents textuels afin de définir de nouvelles techniques d'analyse en ligne de données textuelles. Néanmoins, une classification de textes dans ce contexte nécessite une adaptation, notamment pour la prise en compte des informations sémantiques contenues dans la mesure d'analyse textuelle de notre modèle *CXT-Cube*. Pour cela, nous proposons un nouvel opérateur baptisé *TLabel* (*Text Label*), qui agrège un ensemble de documents considérés comme similaires par une étiquette représentant le contenu sémantique de ces derniers. Pour atteindre cet objectif, nous proposons un couplage entre les techniques de fouille de textes et celles de l'OLAP.

Contrairement à la fouille de textes, l'analyse en ligne dépend de la requête d'analyse. Elle offre la possibilité au décideur d'effectuer une analyse navigationnelle à travers différent axes d'analyse et plusieurs niveaux hiérarchiques. Nous proposons une approche d'agrégation des données textuelles dans les cubes de textes qui comporte deux étapes : dans un premier temps, nous proposons d'agréger les données textuelles en plusieurs classes de documents homogènes en se basant sur une classification non supervisée. Par la suite, nous proposons une démarche qui permet d'attribuer une étiquette à chaque classe de documents en se basant sur un document représentant de la classe appelé *DResume* (*Document Resume*) et les hiérarchies sémantiques des dimensions sémantiques du *CXT-Cube*.

4.1 Classification de documents dans un cadre OLAP

A partir de notre mesure d'analyse textuelle définie dans notre modèle *CXT-Cube*, Les documents sont classés selon une méthode de classification. Notre choix s'est porté sur la méthode de *K-means* (Macqueen, 1967) qui recherche un partitionnement de documents basé sur une distance. Plusieurs variantes de cet algorithme existent, parmi lesquelles *Spherical K-means* (Dhillon et al., 2001) qui utilise la mesure *Cosinus* pour établir la distance entre documents. Nous proposons une adaptation de l'algorithme *K-means* à notre cube de textes. Pour calculer la distance entre des documents, nous utilisons l'algorithme *OCluster* (*OLAP-Cluster*) basé sur la fonction *ORank(d)*, que nous avons proposée dans (Oukid et al., 2013a), et qui est adaptée à notre mesure d'analyse textuelle.

Formalisation Soit $q =< q_1, q_2, ..., q_n >$ la requête d'analyse, tel que q_i , est la sous requête spécifique à la dimension Dim_i .

Soit l'ensemble de documents $D =< d_1, d_2, ..., d_* >$ retournés par la requête d'analyse q, où d_i est représenté par notre mesure d'analyse textuelle $M =< \overrightarrow{d_{Dim_1}}, \overrightarrow{d_{Dim_2}}, ..., \overrightarrow{d_{Dim_*}} >$. L'algorithme *OCluster* comprend les étapes suivantes :

1. Initialement, les k centroides $< ct_1, ct_2, ..., ct_k >$ sont définis par tirage aléatoire à partir de D.

2. Ensuite, pour chaque document d_r de D, calculer la distance entre ce dernier et chacun des centroides par la fonction $ORank(d)$:

$$ORank(d) = \frac{\sum_{i=1}^{n}(\alpha_i \times Sim(\overrightarrow{d_{Dim_i}}, \overrightarrow{ct_{Dim_i}}))}{n} \quad (1)$$

$Sim(\overrightarrow{d_{Dim_i}}, \overrightarrow{ct_{Dim_i}})$ est la similarité cosinus entre $\overrightarrow{d_{Dim_i}}$ et $\overrightarrow{ct_{Dim_i}}$ (Salton et al., 1975) :

$$Sim(\overrightarrow{d_{Dim_i}}, \overrightarrow{ct_{Dim_i}}) = \cos a = \frac{\sum_i w_{c_i} * wct_{c_i}}{\sqrt{\sum_i w_{c_i}^2 * \sum_i wct_{ci}^2}} \quad (2)$$

où w_{c_i} est le poids du concept c_i dans $\overrightarrow{d_{Dim_i}}$ et wct_{c_i} est le poids de c_i dans $\overrightarrow{ct_{Dim_i}}$.
α_i représente les préférences du décideur pour chaque dimension Dim_i du ***CXT-Cube*** ; α_i est calculé comme suit :

$$\alpha_i = Per_i \times n \quad (3)$$

où Per_i est le pourcentage d'importance accordé à Dim_i et n est le nombre de dimensions.

3. Construire un ensemble de classes $Cl =< cl_1, cl_2, ..., cl_k >$ par affectation de chaque document d_r au centre le plus proche.
4. Recalculer les centroides $< ct'_1, ct'_2, ..., ct'_k >$ des classes de documents. Pour chaque classe cl_j, ct'_j est représenté par plusieurs vecteurs $< \overrightarrow{ct'_{Dim_1}}, \overrightarrow{ct'_{Dim_2}}, ..., \overrightarrow{ct'_{Dim_n}} >$, un pour chaque dimension Dim_i. Tel que $\overrightarrow{ct'_{Dim_i}}$ est calculé comme suit :

$$\overrightarrow{ct'_{Dim_i}} = \frac{\sum_{i=1}^{N} \overrightarrow{d_{Dim_i}}}{N} \quad (4)$$

où N est le nombre de documents dans la classe cl_j.

5. Répéter les étapes 2 et 3 jusqu'à ce que deux itérations successives donnent la même distribution de documents dans les classes.

4.2 Étiquetage des classes de documents

Dans cette deuxième étape, nous proposons d'attribuer des étiquettes aux classes de documents précédemment définies. Afin de pouvoir attribuer une étiquette à chaque classe de documents, nous proposons d'exploiter les hiérarchies sémantiques de notre modèle ***CXT-Cube***. Nous proposons au décideur d'exprimer ses préférences sur les dimensions sémantiques du ***CXT-Cube***. Le décideur peut choisir une dimension sémantique sur laquelle il souhaite avoir l'étiquetage. Notre démarche comprend les étapes suivantes :

1. **Calcul de *DResume***
   Initialement, ***DResume*** (***Document Resume***) est calculé pour chaque classe de documents cl_j. ***DResume*** est considéré comme le représentant des documents d'une même classe.

DResume est exprimé, comme dans notre mesure d'analyse textuelle, par plusieurs vecteurs, un pour chaque dimension Dim_i :

$$DResume =< \overrightarrow{DResume_{Dim_1}}, \overrightarrow{DResume_{Dim_2}}, ..., \overrightarrow{DResume_{Dim_*}} >$$

$\overrightarrow{DResume_{Dim_i}}$ est calculé par la formule suivante :

$$\overrightarrow{DResume_{Dim_i}} = \frac{\sum_{i=1}^{N} \overrightarrow{d_{Dim_i}}}{N} \quad (5)$$

où N est le nombre de documents dans la classe cl_j.

2. **Attribution d'étiquettes aux classes de documents**
   L'étiquetage de documents se fait en se basant sur *DResume* et sur la hiérarchie sémantique choisie.
   Soit H la hiérarchie sémantique associée à une dimension sémantique Dim_s choisie pour l'étiquetage des classes de documents. Soit q_s la requête d'analyse spécifique à la dimension Dim_s, tel que le niveau des nœuds des concepts de q_s est différent de l_0 (niveau des nœuds feuilles dans H).
   Une étiquette est attribuée à chaque classe de documents cl_j. Celle-ci est représentée par le concept C le plus dominant de la classe i.e. ayant le plus grand poids dans $\overrightarrow{DResume_{Dim_s}} =< w_{c_1}, w_{c_2}, ..., w_{c_*} >$, tel que C appartient aux descendants directs des concepts de la requête d'analyse q_s spécifique à Dim_s ($fils(q_s)$) :

$$TLabel_{cl_j} = C/w_C = Max(w_{c_i})_{(i=1,...,k)} \wedge c_i \in fils(q_s)$$

3. **Vérification des étiquettes**
   Si deux classes de documents ou plus ont la même étiquette, ces dernières peuvent être regroupées en une seule classe ou ré-étiquetées comme suit :
   Soit cl_r et cl_m deux classes de documents ayant une même étiquette représentée par le concept C.
   - Pour chacune des classes cl_r et cl_m, le deuxième concept ayant le plus grand poids C^* dans $\overrightarrow{DResume_{Dim_s}}$ est calculé :
     $$TLabel'_{cl_j} = C^*/w_{C^*} = Max(w_{c_i})_{(i=1,...,k)} \wedge c_i \in fils(q_s) \wedge C^* \neq C$$
     où $fils(q_s)$ représente les nœuds fils des concepts de q_s dans H.
   - Si $TLabel'_{cl_r} = TLabel'_{cl_m}$, alors cl_r et cl_m sont regroupées en une seule classe de documents ayant comme étiquette le concept C.
   - Si cl_r et cl_m n'ont pas comme seconde étiquette le même concept C^*, alors ces dernières sont ré-étiquetées en combinant les deux concepts C et C^*.

4.3 Exemple d'agrégation de données textuelles en utilisant TLabel

Prenons l'exemple d'une analyse en ligne sur une collection de CVs. Les documents sont observés selon deux dimensions sémantiques : THÉMATIQUE, LOCALISATION et une dimension de type méta-donnée : TEMPS.

Supposons que nous ayons la requête d'analyse $q =< q_z, q_l, q_t >$ tel que : $q_z =<$ "$Informatique$" $>$, $q_l =<$ "$France$" $>$ et $q_t =< 2014 >$. q_z, q_l et q_t sont les sous-requêtes des dimensions THÉMATIQUE, LOCALISATION et TEMPS respectivement.

Le tableau 1 montre un exemple de données textuelles utilisées dans cet exemple. Cinq documents sont considérés, chacun d'eux est représenté selon notre mesure d'analyse textuelle par trois vecteurs $< \overrightarrow{d_{Dim_z}}, \overrightarrow{d_{Dim_l}}, \overrightarrow{d_{Dim_t}} >$, un pour chaque dimension du *CXT-Cube*.

Doc	$\overrightarrow{d_{Dim_z}}$	$\overrightarrow{d_{Dim_l}}$	$\overrightarrow{d_{Dim_t}}$
d_1	< Informatique (0.17), Programmation(0.03), Java(0.05), PHP(0.00), C(0.08), Base de données(0.09), Oracle(0.05), Mysql(0.05), Pl/Sql(0.00), Décisionnel(0.16), OLAP(0.17), ETL(0.00), Entrepot de données(0.15)>	<France(0.4), Rhone-Alpes(0.2), Lyon(0.4)>	2014(1.0)
d_2	< Informatique (0.18), Programmation(0.13), Java(0.25), PHP(0.17), C(0.08), Base de données(0.06), Oracle(0.08), Mysql(0.05), Pl/Sql(0.00), Décisionnel(0.00), OLAP(0.00), ETL(0.00), Entrepot de données(0.00)>	<France(0.3), Midi-Pyrenees(0.2), Toulouse(0.5)>	2014(1.0)
d_3	< Informatique (0.18), Programmation(0.08), Java(0.10), PHP(0.05), C(0.00), Base de données(0.16), Oracle(0.10), Mysql(0.05), Pl/Sql(0.10), Décisionnel(0.08), OLAP(0.00), ETL(0.05), Entrepot de données(0.05)>	<France(0.2), Rhone-Alpes(0.1), Lyon(0.2), Valence(0.5) >	2014(1.0)
d_4	< Informatique (0.11), Programmation(0.09), Java(0.09), PHP(0.00), C(0.08), Base de données(0.17), Oracle(0.05), Mysql(0.05), Pl/Sql(0.05), Décisionnel(0.11), OLAP(0.10), ETL(0.00), Entrepot de données(0.1)>	<France(0.2), Rhone-Alpes(0.1), Lyon(0.2), Grenoble(0.5)>	2014(1.0)
d_5	< Informatique (0.16), Programmation(0.10), Java(0.10), PHP(0.17), C(0.08), Base de données(0.01), Oracle(0.00), Mysql(0.05), Pl/Sql(0.00), Décisionnel(0.18), OLAP(0.00), ETL(0.05), Entrepot de données(0.10)>	<France(0.2), Rhone-Alpes(0.1), Lyon(0.5), Grenoble(0.2)>	2014(1.0)

TAB. 1 – *Données textuelles utilisées dans l'exemple*

Étape 1 : Construction de groupes de documents en utilisant *OCluster*

Comme expliqué précédemment, la première étape de notre processus d'agrégation par catégorisation de documents est la classification des documents en k classes.

Dans cet exemple, l'étiquetage des documents est fait par rapport aux compétences (dimension THÉMATIQUE). La figure 1 illustre un exemple d'une hiérarchie sémantique des compétences associées au *CXT-Cube*.

Prenons le cas où $k = 3$. Les classes de documents sont calculées en appliquant l'algorithme *OCluster*.

Étape 2 : Étiquetage des classes de documents

Afin d'affecter une étiquette à chaque classe de documents, *DResume* est calculé pour chacune d'elles en utilisant la formule 5. Le tableau 2 montre les résultats des résumés tex-

tuels. *DResume* est représenté par trois vecteurs, un pour chaque dimension du *CXT-Cube* : $< \overrightarrow{DResume_{Dim_z}}, \overrightarrow{DResume_{Dim_l}}, \overrightarrow{DResume_{Dim_t}} >$.

OCluster	$\overrightarrow{DResume_{Dim_z}}$	$\overrightarrow{DResume_{Dim_l}}$	$\overrightarrow{DResume_{Dim_t}}$
$cl_1 = \{d_1, d_5\}$	< Informatique (0.165), Programmation(0.065), Java(0.075), PHP(0.085), C(0.08), Base de données(0.05), Oracle(0.025), Mysql(0.05), Pl/Sql(0.00), **Décisionnel(0.17)**, OLAP(0.085), ETL(0.025), Entrepot de données(0.125)>	<France(0.3), Rhone-Alpes(0.15), Lyon(0.45), Grenoble(0.1)>	2014(1.0)
$cl_2 = \{d_3, d_4\}$	< Informatique (0.145), Programmation(0.085), Java(0.095), PHP(0.025), C(0.04), **Base de données(0.165)**, Oracle(0.075), Mysql(0.05), Pl/Sql(0.075), Décisionnel(0.095), OLAP(0.05), ETL(0.025), Entrepot de données(0.075)>	<France(0.2), Rhone-Alpes(0.1), Lyon(0.2), Grenoble(0.25)>	2014(1.0)
$cl_3 = \{d_2\}$	< Informatique (0.18), **Programmation(0.13)**, Java(0.25), PHP(0.17), C(0.08), Base de données(0.06), Oracle(0.08), Mysql(0.05), Pl/Sql(0.00), Décisionnel(0.00), OLAP(0.00), ETL(0.00), Entrepot de données(0.00)>	<France(0.3), Midi-Pyrenees(0.2), Toulouse(0.5)>	2014(1.0)

TAB. 2 – *Calcul de DResume pour les classes de documents*

Une fois *DRsume* calculé pour chaque classe de documents, une étiquette représentant le contenu des données textuelles est attribuée à chaque classe. La figure 1 montre un exemple d'extraction de l'étiquette spécifique à la classe de documents cl_1.

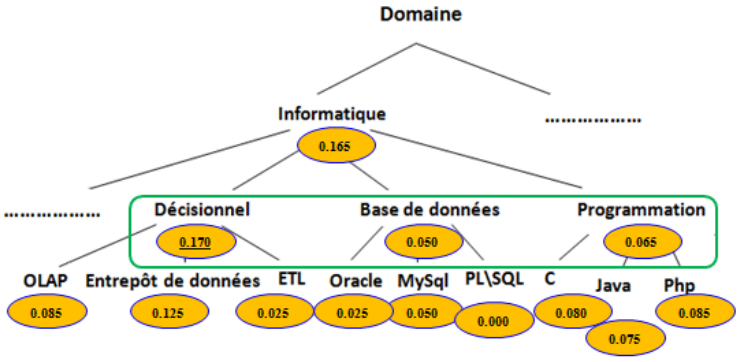

FIG. 1 – *Exemple illustratif de l'étiquetage de la classe de documents* cl_1

Le nœud *Informatique* (concept de la requête d'analyse q_z spécifique à la dimension THÉMATIQUE) comprend trois fils : *Décisionnel*, *Base de données* et *Programmation*. l'étiquette de la classe cl_1 est représentée par le nœud ayant le plus grand poids dans $\overrightarrow{DResume_{Dim_z}}$ (voir Tableau 2) et appartenant aux fils du concept *Informatique* (voir Figure 1). Dans cet exemple l'étiquette de la classe cl_1 est représentée par le concept *Décisionnel*. De la même manière, une étiquette est attribuée à chaque classe de documents.

La figure 2 montre les résultats générés par notre opérateur d'agrégation *TLabel*. Dans cet exemple, les documents sont agrégés en trois catégories : *Décisionnel*, *Base de données* et *Programmation*.

Opérateur d'agrégation *TLabel*				**THEMATIQUE**	
				Domaine	Informatique
LOCALISATION	Ville	**TEMPS**	Année		
	France		2014		

Classe	**Documents**	***Label***
Cl 1	{d1, d5}	Décisionnel
Cl 2	{d3, d4}	Base de données
Cl 3	{d2}	Programmation

FIG. 2 – *Exemple d'analyse en utilisant TLabel*

5 Expérimentations

Dans cette section, nous présentons les expérimentations pour valider notre opérateur d'agrégation par catégorisation *TLabel*. Nous avons développé un prototype en Java et nous avons mené une étude expérimentale sur un corpus de CVs.

5.1 Jeu de données

Nous avons appliqué notre modèle sur une collection d'environ 2000 documents représentant des CVs de candidats dans différents sous-domaines de l'informatique. Pour la construction des hiérarchies de concepts, nous avons utilisé le portail thématique de l'encyclopédie libre *Wikipédia* [1] et l'ontologie géographique *Geonames* [2].

5.2 Protocole d'expérimentation

Les expérimentations réalisées comportent les étapes suivantes :

1. Pré-traitements : c'est une phase de préparation et de nettoyage des données. Les principales tâches effectuées sont : La *Tokenisation* du texte, afin de récupérer les termes ; La

1. http://fr.wikipedia.org
2. http://www.geonames.org

conversion du texte en minuscule ; L'élimination des mots vides ; Enfin, la *Lemmatisation* des termes en utilisant l'outil morphosyntaxique *tree tagger* [3].

2. Alimentation des dimensions du *CXT-Cube* : dans notre exemple, *CXT-Cube* comprend deux dimensions sémantiques : THÉMATIQUE et LOCALISATION, et une dimension métadonnées : TEMPS. Pour chaque dimension sémantique, nous avons conçu une hiérarchies de concepts sous forme d'un arbre XML. Pour parcourir ce dernier, nous avons utilisé l'API java open source DOM4J [4].
3. Tests effectués : nous avons testé des requêtes d'analyse sur *CXT-Cube*. Afin d'évaluer la fonction *ORank(d)*, nous avons effectué des tests en proposant une comparaison des résultats d'agrégation sans et avec considération des préférences du décideur représentées par α_i (formule 3). Les résultats sont présentés dans (Oukid et al., 2013a) et (Oukid et al., 2013b). Nous exposons dans ce qui suit les résultats obtenus de l'application de notre opérateur *TLabel*.

5.3 Résultats

Nous avons choisi d'utiliser la dimension THÉMATIQUE comme base pour l'étiquetage. Cette dernière comprend une hiérarchie de compétences en informatique.

Le paramètre k peut être défini soit par le décideur, soit comme étant le nombre de descendants des concepts de la sous-requête spécifique à la dimension choisie pour l'étiquetage. Dans nos expérimentations, nous avons choisi d'agréger les données en 8 classes ($k = 8$).

Dans un premier temps, nous montrons les résultats de l'application de notre opérateur d'agrégation *TLabel* pour la requête d'analyse $q =< q_z, q_l, q_t >$ tel que :

$q_z =<$ "$Informatique$" $>$, $q_l =<$ "$France$" $>$, $q_t =< 2014 >$; q_z, q_l et q_t sont les sous-requêtes des dimensions THÉMATIQUE, LOCALISATION et TEMPS respectivement. Par la suite, nous montrons les résultats d'une opération de forage vers le bas (*Drill-Down*) sur la dimension THÉMATIQUE, pour observer les compétences en *"Programmation"*.

La première étape de notre processus d'agrégation de données textuelles par catégorisation consiste en la construction de groupes homogènes de documents. Le tableau 3 montre les résultats retournés par l'algorithme *OCluster* pour la requête d'analyse q.

$OCluster$	Cl_1	Cl_2	Cl_3	Cl_4	Cl_5	Cl_6	Cl_7	Cl_8
Nombre de documents	98	181	84	179	215	178	1	216
Total	1152							

TAB. 3 – *Résultats de OCluster sur le corpus de documents*

Nous avons obtenu huit classes de documents, chacune comportant un ensemble de documents considérés comme similaires.

Après la classification des documents, en utilisant *OCluster*, les classes de documents sont étiquetées. Le tableau 4 montre qu'à partir des huit classes de documents retournées par l'algorithme *OCluster*, trois catégories se sont distinguées, chacune regroupe un nombre de classes partageant une même étiquette.

3. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
4. http://dom4j.sourceforge.net

$TLabel$	Programmation-Réseau	Programmation-Conception	Programmation-Décisionnel
$OCluster$	Cl_1, Cl_7	$Cl_2, Cl_3, Cl_4, Cl_6, Cl_8$	Cl_5
Nb documents	99 (8 %)	838 (73 %)	215 (19 %)

TAB. 4 – *Résultats de l'étiquetage des classes de documents*

La figure 3 montre les résultats d'analyse en utilisant l'opérateur *TLabel* pour la requête d'analyse q.

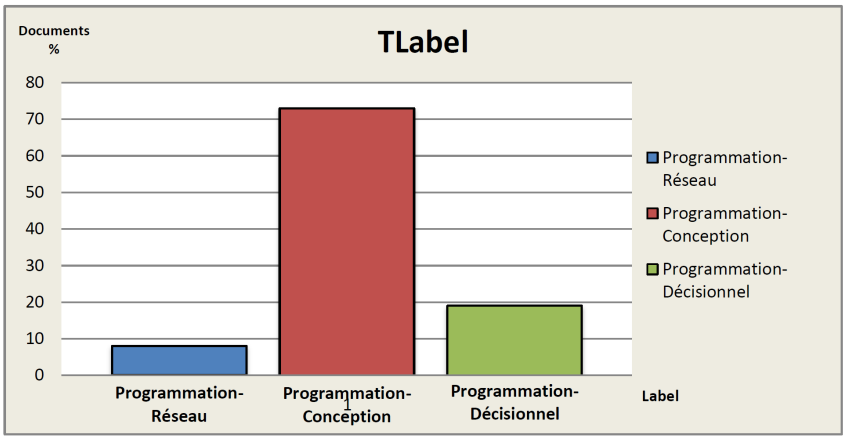

FIG. 3 – *Résultats d'analyse en utilisant TLabel*

A travers les résultats montrés dans la figure 3, nous observons qu'à partir des huit classes de départ, nous avons obtenu les trois catégories suivantes : *Programmation-Réseau*, *Programmation -Conception* et *Programmation-Décisionnel*. Les résultats d'analyse montrent que la majorité des candidats présentent des compétences en *Programmation* et en *Conception* (78%). Nous constatons également que le concept *Programmation* est la compétence principale qui apparait dans les CVs analysés.

Pour une vision plus détaillée du concept *Programmation*, nous proposons d'effectuer un *Drill-Down* sur la dimension THÉMATIQUE.

Les résultats d'analyse illustrés dans la Figure 4 montrent que trois compétences principales se sont distinguées : *C*, *Java* et *Php*. Ces compétences sont regroupées en trois catégories : *Php-C*, *C-Java* et *Java-C* (le concept en première position dans une étiquette représente la compétence principale de la catégorie). La catégorie ayant comme étiquette *C-Java* est celle qui comprend le plus grand nombre de documents (69 %).

Après cette étape d'agrégation de CVs par catégorisation, le décideur pourra agréger les données d'un même groupe en utilisant l'opérateur d'agrégation *ORank* (*OLAP Rank*) (Oukid et al., 2013a) et (Oukid et al., 2013b). Cette analyse met en évidence les candidatures les plus pertinentes au sein d'une catégorie qui l'intéresse ; ce qui facilitera au décideur ses prises de décision pour le recrutement.

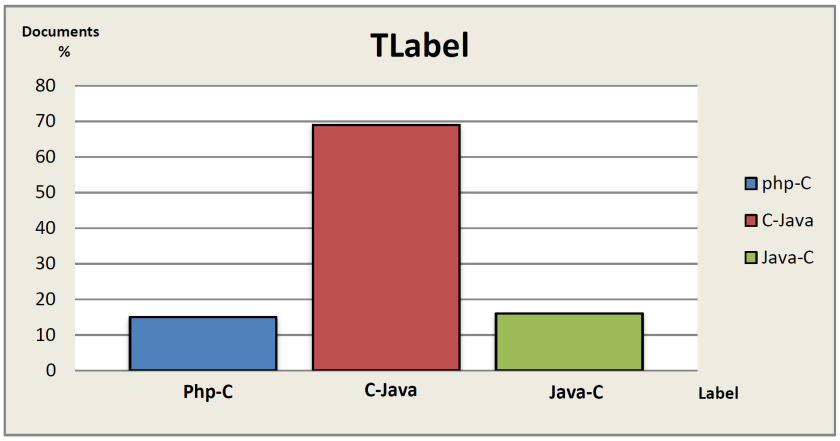

FIG. 4 – *Résultats du Drill-Down sur la dimension* THÉMATIQUE *en utilisant TLabel*

6 Conclusion

Dans cet article, nous avons proposé un opérateur d'agrégation par catégorisation *TLabel* (*Text-Label*) adapté à notre modèle *CXT-Cube*. Notre opérateur permet d'agréger les données textuelles en plusieurs catégories en se basant sur *OCluster* : une adaptation de l'algorithme *K-means* à l'OLAP. Dans cet algorithme, la similarité entre documents est calculée en utilisant la fonction d'agrégation *ORank(d)* adaptée à notre mesure d'analyse textuelle. Pour chaque classe de documents, *DResume* : un document qui représente le contenu sémantique des documents d'une même classe est calculé. A chaque classe de documents est associée une étiquette représentant son contenu. L'étiquetage est effectué en exploitant *DResume* et les hiérarchies sémantiques du *CXT-Cube*. Nous prévoyons dans le futur d'améliorer la formule de calcul de *DResume* pour une meilleure prise en compte de la sémantique des documents d'une même classe. Un étiquetage des classes de documents selon plusieurs hiérarchies sémantiques est aussi envisageable. Enfin, nous planifions d'évaluer notre approche sur des données volumineuses dans le cadre des *Big Data*, notamment en utilisant des modèles *NoSQL* orientés documents, afin de tester sa capacité au passage à l'échelle.

Références

Bringay, S., N. Béchet, F. Bouillot, P. Poncelet, M. Roche, et M. Teisseire (2011). Analyse de gazouillis en ligne. *Journées francophone sur les Entrepots de Données et l'Analyse en ligne 7*, 87–102.

Cody, W. F., J. T. Kreulen, V. Krishna, et W. S. Spangler (2002). The integration of business intelligence and knowledge management. *IBM Systems Journal 41*, 697–713.

Dhillon, I., J. Fan, et Y. Guan (2001). Efficient clustering of very large document collections. *Data Mining for Scientific and Engineering Applications*, 357–381.

Felman, R. et J. Sanger (2007). *The text mining handbook advanced approaches in analyzing unstructured data*. New York, NY, USA: Cambridge university press.

Lin, C. X., B. Ding, J. Han, F. Zhu, et B. Zhao (2008). Text cube: Computing ir measures for multidimensional text database analysis. *ICDM*, 905–910.

Macqueen, J. B. (1967). Some methods of classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

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

Oukid, L., O. Asfari, F. Bentayeb, N. Benblidia, et O. Boussaid (2013a). Cube de textes et opérateur d'agrégation basé sur un modèle vectoriel adaptée. *9èmes Journées francophone sur les Entrepots de Données et l'Analyse en ligne EDA'13 RNTI, B-9, Hermann*, 79–94.

Oukid, L., O. Asfari, F. Bentayeb, N. Benblidia, et O. Boussaid (2013b). Cxt-cube: Contextual text cube model and aggregation operator for text olap. *Proceeding of the sixteenth international workshop on data warehousing and OLAP DOLAP*, 27–32.

Pérez, J., R. Berlanga, M. Aramburu, et T. Pedersen (2007). R-cube: Olap cubes contextualized with documents. *IEEE International Conference 23*, 1477–1478.

Ravat, F., O. Teste, R. Tournier, et G. Zurfluh (2008). Top keyword: an aggregation function for textual document olap. *Intl Journal of data Warehousing and Mining*, 55–64.

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

Zhang, D., C. Zhai, et J. Han (2009). Topic modeling for olap on multidimensional text databases. *Statistical Analysis and Data Mining 2*, 378–395.

Zhang, D., C. Zhai, et J. Han (2011). Mitexcube: Microtextcluster cube for online analysis of text cells. *Conference on Intelligent Data Understanding*, 204–218.

Summary

Classical aggregation operators are efficient for aggregating numerical data but are not suitable for textual data. To alleviate this shortcoming, Online Analytical Processing (OLAP) in text cubes requires new analysis operators designed for textual data. In this paper, we propose a new aggregation operator based on categorization, named Text Label (*TLabel*), that allows the aggregation of textual data in several classes of documents. Each class is associated with a label representing the semantic content of textual data, thanks to a tailoring of text mining techniques to OLAP. The preliminary results of our experimental study show the interest of our approach for Text OLAP.