Cube de textes et opérateur d'agrégation basé sur un modèle vectoriel adapté

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

*Université Saad Dahlab Blida (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 Mendès France 69676 Bron Cedex, France
{ounas.asfari, fadila.bentayeb, omar.boussaid}@univ-lyon2.fr

Résumé. Les technologies d'entreposage de données et d'analyse en ligne (*On-Line Analytical Processing* OLAP) ont largement fait leurs preuves pour l'analyse de données structurées, mais elles sont inadaptées pour l'analyse des données textuelles, faute d'outils et de méthodes adaptés. Nous proposons dans cet article, un modèle de cube textuel nommé *TCube*, qui comporte plusieurs dimensions sémantiques, pour une meilleure prise en charge de la sémantique des données textuelles. Les attributs de chaque dimension sémantique sont regroupés dans une hiérarchie de concepts, extraite à partir d'une ontologie de domaine utilisée comme une ressource externe. Notre cube de textes comprend une mesure d'analyse textuelle qui s'appuie à la fois sur un modèle vectoriel adapté à l'analyse OLAP et sur une technique de propagation de pertinence. Il est également associé à un nouvel opérateur d'agrégation appelé *ORank*(*OLAP-Rank*) permettant d'agréger les données textuelles dans un environnement OLAP. Les résultats préliminaires de notre étude expérimentale montrent l'intérêt de notre approche.

1 Introduction

De nos jours, les technologies d'entrepôts de données et d'analyse en ligne (OLAP) sont efficaces pour traiter des données numériques. Néanmoins, une grande partie des données circulant dans les entreprises (texte, images, vidéo, etc.) reste hors de portée des systèmes décisionnels. Ces dernières sont appelées *données complexes*. La plupart d'entre elles représentent des données textuelles (rapports, e-mails, etc.). Dans cet article, nous nous intéressons à l'analyse OLAP de ces données textuelles.

L'intégration des informations issues de données textuelles dans un processus d'analyse en ligne représente un défi pour les systèmes décisionnels. D'autre part, l'agrégation des données numériques s'effectue à l'aide de fonctions d'agrégat (somme, moyenne, min, max, etc.). Or, ces dernières ne sont pas adaptées à l'agrégation de données textuelles. La nature non structurée de ces dernières les rend difficile à analyser. Par conséquent, les questions suivantes se posent :

comment peut-on représenter les données textuelles de façon multidimensionnelles dans un cube OLAP afin de les analyser ? Comment peut-on agréger ces données ? Pour répondre à ces interrogations, il est nécessaire de faire évoluer les schémas en étoile existants et définir de nouvelles techniques d'agrégation adaptées à la nature des données textuelles.

Les systèmes OLAP permettent de naviguer dans des cubes multidimensionnels. Ils offrent la possibilité de passer d'une vue à l'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 à la prise de décision. Cependant, pour traiter les données textuelles, il est souvent nécessaire d'avoir recours aux techniques de recherche d'information (RI). Ces dernières évaluent, par example, la pertinence des résultats par rapports aux mots-clefs exprimant l'information recherchée. Cette pertinence est souvent basée sur la fréquence des mots-clefs dans le document. Les résultats générés par les systèmes de RI restent donc limités à une extraction d'information à partir de documents et nécessitent un jugement de l'utilisateur. Par ailleurs, dans un système OLAP, on est concerné par une analyse navigationnelle qui, dans le cas d'une analyse de données textuelles, peut s'appuyer sur des opérateurs issus de la RI. Par exemple, nous prenons un corpus de documents représentant des Curriculum vitae (CVs) pour la sélection de candidatures lors d'un recrutement. Les réponses à une offre d'emploi représentent une grande masse de documents difficiles à gérer par le recruteur. Il devient nécessaire d'assister le recruteur pour une meilleure prise de décision. Une analyse OLAP, permet d'aller au-delà d'une simple recherche par mots-clefs comme dans les systèmes RI. Elle permet au décideur à travers la navigation dans le cube OLAP, d'observer les données selon plusieurs axes d'analyses organisés selon différent niveaux de hiérarchie. Le décideur pourra par exemple, observer les compétences en informatique pour l'année 2012 en France puis, à travers un forage vers le bas, observer celles en informatique décisionnelle pour l'année 2012 en France, etc.

Pour analyser en ligne des données textuelles, il est donc nécessaire de faire appel aux différents domaines qui s'intéressent aux traitements de texte, tels que la recherche d'information, la fouille de données et l'extraction d'information (Park et Song, 2011). Notre idée clé consiste alors à proposer une approche qui combine les techniques de RI et celles de l'analyse en ligne. L'intérêt de cette combinaison est de permettre l'analyse en ligne de gros corpus de documents pour l'aide à la décision et de faciliter la navigation dans les cubes de textes pour répondre aux besoins d'analyse. Les techniques de RI permettent donc d'une part, l'extraction d'informations pertinentes à partir des documents afin de construire des mesures d'analyse textuelles, et d'autre part, de définir des opérateurs OLAP adaptés aux données textuelles.

Dans cet article, nous proposons un modèle de cube de données textuelles, appelé *TCube*, comportant plusieurs dimensions sémantiques. Les attributs de ces dimensions sont regroupés dans des hiérarchies de concepts extraites à partir d'ontologies de domaines. Pour le calcul du poids des termes représentatifs d'un document dans la hiérarchie de concepts, nous proposons une technique de propagation de pertinence. Dans le modèle d'entrepôt de donnés textuelles, la table de faits va comporter une nouvelle mesure d'analyse textuelle, basée sur une représentation vectorielle des données textuelles. Nous associons à cette mesure, un nouvel opérateur d'agrégation de documents noté *ORank* (*OLAP-Rank*), grâce à une adaptation du modèle vectoriel à l'analyse OLAP. La suite de cet article est organisée comme suit. Dans la section 2, nous présentons un état de l'art des travaux pour l'analyse OLAP adapté aux données textuelles. La section 3 expose notre modèle de données multidimensionnel *TCube*. La section 4

illustre notre opérateur d'agrégation ORank. La section 5 fait état de nos expérimentations et présente nos résultats. Enfin, la section 6 conclut cet article et présente quelques perspectives de recherche de notre travail.

2 État de l'art

Des travaux récents proposent une extension du cube de données classique afin de supporter des analyses OLAP sur les données textuelles. Zhang et al. (2009) agrègent les données textuelles en plusieurs niveaux hiérarchiques de thèmes (*Topics*). Pour ce faire, les auteurs 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*). Cependant, l'extraction des thèmes par un modèle probabiliste ne génère pas toujours des résultats significatifs. 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. Lin et al. (2008) proposent une nouvelle dimension à travers leur cube de données nommé *Texte 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*). Zhang et al. (2011) ont défini un modèle de cube de textes (*MiTexCube*), qui 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* (Salton et al., 1975) et sa taille. Ces travaux utilisent des mesures d'analyse basées sur la méthode statistique *Tf-Idf*. Or, cette dernière ne permet pas une réelle prise en compte de la sémantique véhiculée dans les données textuelles dans un contexte d'analyse. Par exemple, dans le contexte d'une analyse OLAP de CVs, il est plus intéressant d'extraire les informations sur les compétences des candidatures plutôt que d'extraire les termes les plus fréquents.

Parmi les propositions d'opérateurs d'agrégation sur des données textuelles, nous retrouvons les travaux de Ravat et al. (2007) qui proposent une fonction d'agrégation appelée *AVG-KW*. Celle-ci permet de regrouper des mots-clefs en des mots-clefs plus généraux à l'aide d'une ontologie de domaine. Dans la suite de leurs travaux, Ravat et al. (2008) ont défini une deuxième fonction d'agrégation nommée *TOP-KWk*, qui retourne une liste des k mots-clefs avec les plus grands poids dans un document. Ces mots-clefs sont pondérés par la méthode *Tf-Idf*. Ben-Messaoud et al. (2004) proposent *OpAC*, un opérateur d'agrégation par classification. Cet opérateur permet une classification hiérarchique, selon l'ordre de proximité dans les dimensions, en utilisant la méthode de *Classification Ascendante Hiérarchique CAH*. 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.

Nous constatons que la plupart des travaux récents font appel aux différents domaines qui s'intéressent à l'analyse de textes, tels que la recherche d'information ou la fouille de textes,

afin de pouvoir analyser les données textuelles. Notre travail s'inscrit dans la tendance des travaux qui proposent un couplage entre l'OLAP et la recherche d'information. Toutefois, notre démarche est différente. Contrairement aux travaux de Zhang et al. (2009), la *dimension sémantique* du modèle *TCube* que nous proposons est extraite à partir d'une ontologie métier, où les concepts définis sont plus significatifs que ceux extraits par les méthodes *PLSA* ou *LDA*. De plus, la mesure que nous proposons diffère de celle de Lin et al. (2008) et de celle Zhang et al. (2011), qui se basent toutes les deux sur la méthode classique *Tf-Idf*. Notre mesure comprend des informations sémantiquement plus riches, grâce à une représentation multidimensionnelle des données textuelles et une technique de propagation de pertinence. Les opérateurs d'agrégation qui ont été proposés dans la littérature restent insuffisants. A notre connaissance aucun d'entre eux n'offre des outils permettant au décideur de disposer d'un classement de documents, intégrant les préférences du décideur par rapport aux dimensions du cube OLAP.

3 Modèle de données textuelles

Dans cette section, nous présentons une extension du cube de données classique pour modéliser un cube de textes, permettant la prise en compte de la sémantique des données textuelles.

3.1 Formalisation

Un cube de données $Cube$ permet de modéliser un sujet d'analyse appelé *Fait* et noté F défini selon plusieurs *dimensions* $Dim_r, r \in [1, *]$. Une dimension Dim_r est définie par des attributs $A =< a_1, a_2, ..., a_* >$, chacun pouvant être organisé en plusieurs niveaux hiérarchiques $< l_1, l_2, ..., l_* >$. F est associé à un ou plusieurs indicateurs d'analyse appelé *mesures*. Nous notons M une mesure de la table des faits. Un fait F associé à ses dimensions Dim_r composent un schéma en étoile $Cube = (F, Dim_1, Dim_2, ..., Dim_*, M_1, M_2, ..., M_*)$.

Notre cube de textes, baptisé *TCube* comprend de nouvelles dimensions sémantiques, chacune correspond à une hiérarchie arborescente de concepts.

Définition. Une *Dimension Sémantique* Dim_r est définie sur plusieurs niveaux hiérarchiques l_i. Chaque niveau $l_i =< c_1, c_2, ...c_n >$ comprend un ensemble de concepts $c_j, (j \in [1, n])$ extraits d'une ontologie de domaine.

Pour instancier notre modèle de cube *TCube*, nous prenons l'exemple d'une analyse OLAP sur une collection de CVs. Les documents sont observés selon trois *dimensions sémantiques* : THÉMATIQUE, TEMPS et LOCALISATION, chacune comportant une hiérarchie de concepts. Par exemple, la dimension THÉMATIQUE comporte une hiérarchie de concepts représentant des compétences de domaines. La figure 1 montre un sous-arbre de cette hiérarchie pour le domaine informatique, composé de 4 niveaux hiérarchiques, chacun décrit par un ensemble de concepts.

3.2 Mesure d'analyse textuelle

Dans cette section, nous présentons dans un premier temps une définition de notre mesure d'analyse textuelle, par la suite nous détaillons sa mise en œuvre.

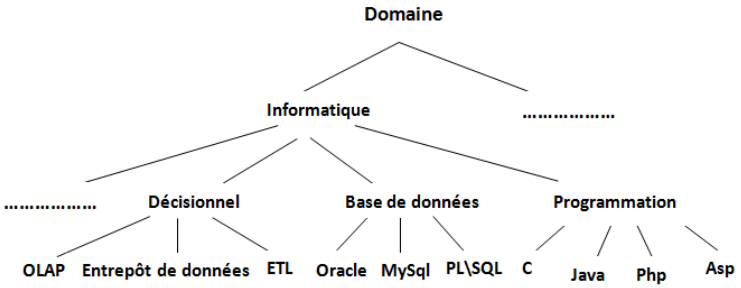

FIG. 1 – *Exemple d'une hiérarchie thématique*

3.2.1 Présentation générale

La sémantique véhiculée dans les documents textes reste faiblement exploitée dans le cadre d'une analyse OLAP. Le fait de s'appuyer sur les seules connaissances issues d'une dimension du cube OLAP ne permet pas une réelle prise en charge des informations sémantiques.

Nous proposons une mesure d'analyse textuelle basée sur une représentation en modèle vectoriel, habituellement utilisée dans le domaine de la RI. Toutefois, une représentation des documents par le modèle vectoriel dans le cadre d'une analyse OLAP nécessite une adaptation, notamment pour la prise en compte des informations sémantiques dans les dimensions du cube textuel. Pour cela, nous proposons une représentation du sujet d'analyse (document) en extrayant des informations sémantiques du document selon les différents axes d'analyse du cube OLAP, représentés par des hiérarchies de concepts. Nous proposons une technique de propagation de pertinence sur ces hiérarchies, pour une meilleur prise en compte de la sémantique des données textuelles. La propagation de pertinence est utilisée en RI, comme dans les travaux de Pinel-Sauvagnat et Boughanem (2006), et Asfari (2008).

Le modèle vectoriel est un modèle algébrique où les documents d'une collection sont représentés par des vecteurs dans un espace multidimensionnel, dont les dimensions sont les termes issus de l'indexation de la collection Salton et McGill (1983). Soit R l'espace vectoriel défini par l'ensemble des termes : $R < t_1, t_2, ..., t_n >$. Un document d est représenté par un vecteur de poids comme suit : $d < w_{t1}, w_{t2}, ..., w_{tn} >$.

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

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

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 et w_{c_i} est le poids attribué au concept c_i.

Dans le cube de textes de la figure 2, la mesure d'analyse est $M =< \overrightarrow{d_{Dim_{th}}}, \overrightarrow{d_{Dim_l}}, \overrightarrow{d_{Dim_t}} >$ où $\overrightarrow{d_{Dim_{th}}}, \overrightarrow{d_{Dim_l}}, \overrightarrow{d_{Dim_t}}$ sont les vecteurs relatifs aux dimensions : THÉMATIQUE (Dim_{th}), LOCALISATION (Dim_l) et TEMPS (Dim_t).

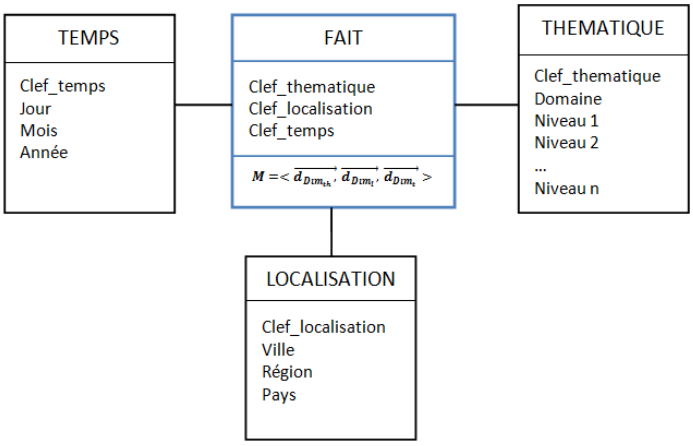

FIG. 2 – *Schéma en étoile de TCube*

La dimension THÉMATIQUE comprend une hiérarchie de thèmes (figure 1). La dimension LOCALISATION comporte trois attributs hiérarchisés qui sont "pays", "région" et "ville". De même pour la dimension TEMPS qui comprend les attributs "année", "mois" et "jour".

3.2.2 Propagation de pertinence pour la mesure d'analyse

Nous proposons une pondération des termes dans les documents en prenant en compte la sémantique des données textuelles. Les poids des termes ne sont pas calculés uniquement par leur fréquence d'apparition, mais aussi par la propagation de leur pertinence dans une hiérarchie de concepts.

La propagation de pertinence permet une ré-attribution de scores à travers une hiérarchie de concepts. Par exemple à travers les concepts *java*, *php*, *c* et *oracle* extraits d'un document, nous pouvons conclure que ce dernier présente une compétence en *informatique*. Grâce à la propagation de pertinence, le poids de ce dernier augmente (prend plus d'importance). La propagation de pertinence permet aussi la prise en compte de nouveaux concepts. Par exemple, un vecteur représentatif d'un document qui comprend les concepts *java*, *c*, *php* va aussi inclure le concept père *programmation*, même si ce dernier ne figure pas dans le document. Notre technique de propagation de pertinence comporte les étapes suivantes :

– Initialement les poids des termes d'un document existants dans la hiérarchie de concepts sont calculés par la méthode Tf *(Term Frequency)*, formule (1).

$$Tf_{t,d} = \frac{n_{t,d}}{N_d} \tag{1}$$

où $n_{t,d}$ est la fréquence d'apparition du terme t dans le document d et N_d est le nombre total de termes dans le document d.

– Ensuite, nous appliquons la propagation de pertinence sur la hiérarchie de concepts, qui permet d'enrichir le vecteur de concepts, en considérant de nouveaux concepts (les nœuds ancêtres des termes pondérés dans le document) et en donnant de nouveaux poids aux termes, et ce grâce à l'exploitation de la distance sémantique entre les différents

nœuds dans la hiérarchie. Le principe de notre méthode de propagation de pertinence consiste à propager des scores attribués à des nœuds feuilles à travers une structure d'arbre. Pour chaque nœud feuille qui a un poids différent de 0, les poids de ses ancêtres sont recalculés par la formule suivante :

$$Poids(n_k, n_{fi}) = Poids(n_k) + Poids(n_{fi})^{distance(n_k, n_{fi})+1} \quad (2)$$

où n_{fi} est le nœud feuille depuis lequel la propagation de pertinence est effectuée, n_k est le nœud ancêtre pour lequel le nouveau poids est calculé et $distance(n_k, n_{fi})$ est la distance sémantique entre les nœuds n_k et n_{fi} représentée par le nombre d'arcs qui les séparent.

La propagation de pertinence dépend de deux paramètres :

- Du nœud feuille n_{fi} à partir duquel la propagation de pertinence est effectuée : quand le poids du nœud feuille est important, la propagation de pertinence augmente.
- De la distance entre le nœud feuille n_{fi} et son ancêtre n_k : quand la distance entre les deux nœuds augmente, la propagation de pertinence diminue.

L'algorithme 1 illustre notre méthode de propagation de pertinence.

Exemple. Supposons que nous ayons une hiérarchie de domaines (figure 3 (a)) liée à la dimension THÉMATIQUE du cube de textes de la figure 2 avec les poids suivants : *mathématiques* (0.1), *informatique* (0.1), *analyse* (0), *statistique* (0.1), *base de données* (0), *programmation* (0), *oracle* (0.3), *mysql* (0), *plsql* (0), *c* (0.3), *java* (0.4), *php* (0.3), *asp* (0).

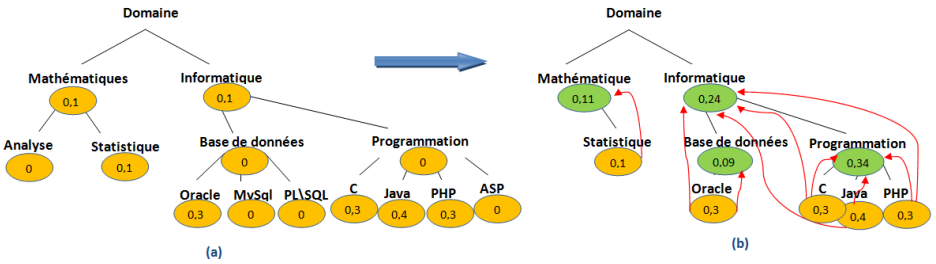

FIG. 3 – *Exemple illustratif de la propagation de pertinence dans une hiérarchie*

Les poids des nœuds *programmation*, *informatique*, *base de données* et *mathématiques* sont recalculés à partir des nœuds feuilles *c*, *java*, *php*, *oracle* et *statistique* (formule 2).

Après application de l'algorithme de propagation de pertinence, le poids du nœud (*programmation*) est passé de 0 à 0.34, par propagation de pertinence à partir de ses fils *c* (0.3), *java* (0.4) et *php* (0.3). Le poids du nœud *base de données* qui avait un poids égal à 0 est passé à 0.09, par propagation de pertinence à partir de son seul nœud fils ayant un poids différent de 0 *oracle* (0.3). Le poids du nœud *informatique* est passé de 0.1 à 0.24 par propagation de pertinence à partir des nœuds feuilles *oracle* (0.3), *c* (0.3), *java* (0.4) et *php* (0.3). Enfin, le poids du nœud *mathématiques* qui avait un poids égal à 0.1 est passé à 0.11 par propagation de pertinence à partir du nœud feuille *statistiques* (0.1) (voir figure 3(b)).

Résultat : $\overrightarrow{d_{Dim_{th}}}$ = *informatique* (0.24), *programmation* (0.34), *java* (0.4), *php* (0.3), *c* (0.3), *base de données* (0.09), *oracle* (0.3), *mathématiques* (0.11), *statistique* (0.1).

ENTRÉES:
Variable f : Booléen
$< c_1, c_2, ..., c_n >$: concepts de la hiérarchie relatifs à une dimension Dim_r
$d < t_1, t_2, ..., t_m >$: termes dans un document d
$< n_1, n_2, ..., n_n >$: nœuds de la hiérarchie de concepts
$< n_{f1}, n_{f2}, ..., n_{fx} >$: nœuds feuilles de la hiérarchie de concepts
nk : nœud dans la hiérarchie de concepts représentant un ancêtre d'un nœud feuille
SORTIES:
d_{Dim_r} : Vecteur de concepts pondérés relatif à une dimension Dim_r

Pour $i = 1$ à n **Faire**
  Pour $j = 1$ à m **Faire**
    Si $c_i.aquals(t_j)$ **Alors**
      $Poids(n_i) \leftarrow Tf_{t_j,d}$
    Finsi
  Fin pour
Fin pour
Pour $i = 1$ à x **Faire**
  $f \leftarrow Vrai$
  Tant que f **Faire**
    Si $Poids(n_{fi}) = 0$ **Alors**
      $Delete(n_{fi})$
      Si $HasNotFils(Parent(n_{fi}))$ **Alors**
        $n_{fi} \leftarrow Parent(n_{fi})$
      Sinon
        $f \leftarrow Faux$
      Finsi
    Sinon
      $f \leftarrow Faux$
    Finsi
  Fin tant que
Fin pour
Pour $i = 1$ à x **Faire**
  $nk \leftarrow Parent(n_{fi})$
  Tant que $nk <> Racine$ **Faire**
    $Poids(nk, n_{fi}) \leftarrow Poids(nk) + Poids(n_{fi})^{distance(nk,n_{fi})+1}$
    $Poids(nk) \leftarrow Poids(nk, n_{fi})$
    $n_k \leftarrow Parent(n_k)$
  Fin tant que
Fin pour

Algorithm.1 – Propagation de pertinence

Grâce à la propagation de pertinence, nous avons deux nouveaux concepts dans notre vecteur de poids : *programmation* et *base de données*. Ainsi, le concept *informatique* a pris davantage d'importance dans ce vecteur.

Avant la propagation de pertinence, le poids des nœuds *informatique* et *mathématiques* était à 0.1 chacun, ce qui fait qu'il était difficile de distinguer si le document (CV) concernait une compétence en mathématiques où bien en informatique. Après propagation de pertinence, le poids du nœud *informatique* est passé de 0.1 à 0.24 tandis que celui du nœud *mathématiques* est resté pratiquement le même (de 0.1 à 0.11). Ainsi, le décideur pourra conclure que le document concerne une compétence en informatique ayant des connaissances en mathématiques.

Les vecteurs $\overrightarrow{d_{Dim_l}}$ et $\overrightarrow{d_{Dim_t}}$ sont calculés de la même manière. La propagation de pertinence est effectuée sur des hiérarchies de concepts extraites à partir d'ontologies géographique et temporelle pour les dimensions LOCALISATION et TEMPS respectivement.

4 Opérateur d'agrégation de données textuelles

L'agrégation des données textuelles dans un environnement OLAP revient à définir de nouvelles techniques. Dans ce cadre, nous présentons dans cette section un nouvel opérateur d'agrégation adapté à notre mesure d'analyse textuelle.

4.1 Présentation de l'opérateur ORank

Nous proposons un nouvel opérateur d'agrégation basé sur une adaptation du modèle vectoriel à l'analyse OLAP. Notre opérateur agrège le contenu sémantique des documents, représenté par notre mesure d'analyse textuelle, en produisant un classement (*Rank*) en combinant les différents espaces vectoriels relatifs aux dimensions de *TCube*.

Le modèle vectoriel défini dans la section 3.2 permet de représenter les documents par des vecteurs dans un espace multidimensionnel comportant les termes issus de l'indexation de la collection. Dans les systèmes de RI, une requête est représentée dans un modèle vectoriel par un vecteur de termes. La similarité entre une requête et un document est calculée en comparant leurs vecteurs respectifs (Salton et McGill, 1983).

Soit R l'espace vectoriel défini par l'ensemble des termes : $R < t_1, t_2, ..., t_n >$, un document $d < w_{t1}, w_{t2}, ..., w_{tn} >$ et une requête $q < wq_{t1}, wq_{t2}, ..., wq_{tn} >$ sont représentés par des vecteurs de poids. où : w_{ti} et wq_{ti} sont les poids du terme ti dans le document d et dans la requête q respectivement ; n représente le nombre de termes dans l'espace. En général, la similarité entre le document d et la requête q est calculée par la mesure de *similarité Cosinus* comme suit :

$$Sim(d, q) = \cos a = \frac{\sum_i w_{ti} * wq_{ti}}{\sqrt{\sum_i w_{ti}^2 * \sum_i wq_{ti}^2}} \tag{3}$$

Soit une requête d'analyse $Q =< \overrightarrow{q_1}, \overrightarrow{q_2}, ..., \overrightarrow{q_*} >$ composée de plusieurs sous requêtes $\overrightarrow{q_r}$, chacune relative à une dimension Dim_r.

L'agrégat d'un document $d =< \overrightarrow{d_{Dim_1}}, \overrightarrow{d_{Dim_2}}, ..., \overrightarrow{d_{Dim_*}} >$ *ORank* est calculé par la formule suivante :

$$ORank(d) = \sum_{i=1}^{n} (\alpha_i \times Sim(\overrightarrow{d_{Dim_i}}, \overrightarrow{q_i}) \tag{4}$$

où $\overrightarrow{d_{Dim_i}}$ est la représentation du document d dans un espace vectoriel relatif à la dimension Dim_i, $Sim(\overrightarrow{d_{Dim_i}}, \overrightarrow{q_i})$ est la *similarité cosinus* entre le document $\overrightarrow{d_{Dim_i}}$ et la requête $\overrightarrow{q_i}$.

Pour mieux répondre aux requêtes d'analyse, nous intégrons dans notre fonction les préférences du décideur. A travers α_i nous proposons au décideur de définir ses préférences par un pourcentage, indiquant l'importance accordée à chaque dimension du cube ; α_i est calculé de la manière suivante :

$$\alpha_i = P_i \times n \tag{5}$$

où P_i est le pourcentage d'importance accordé à la dimension i et n est le nombre de dimensions.

Enfin, les documents sont classés selon un ordre décroissant de $ORank(d)$.

4.2 Exemple d'agrégation de données textuelles

L'analyse navigationnelle à travers les cubes OLAP fait un usage massif des agrégations, notamment les opérations de forage qui permettent de passer d'un niveau hiérarchique à un niveau supérieur ou inférieur. Dans cet exemple, nous agrégeons les documents à travers l'opérateur d'agrégation *ORank (OLAP-Rank)*. Nous utilisons une opération de forage vers le haut sur le cube *TCube* présenté dans la figure 2 .

Le tableau 1 montre les données textuelles utilisées dans l'exemple. Quatre documents sont sélectionnés, les colonnes 2, 3 et 4 représentent les informations extraites à partir des documents relatifs aux dimensions THÉMATIQUE, TEMPS et LOCALISATION.

Documents	THÉMATIQUE	TEMPS	LOCALISATION
CV 1	Informatique ; Statistique ; Décisionnel	10 Jan 2010	France ; Rhône-Alpes ; Lyon
CV 2	Programmation ; Informatique ; Java	25 Jan 2010	France ; Rhône-Alpes ; Lyon
CV 3	Informatique ; Base de données ; Oracle	18 Fév 2010	France ; Midi-Pyrénées ; Toulouse
CV 4	Statistique ; Décisionnel ; Programmation	22 Jan 2010	France ; Rhône-Alpes ; Grenoble

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

Dans la figure 4 (a), le décideur analyse les CVs de l'année 2010 par compétences, et en France, à travers une requête d'analyse $Q =< q_{th}, q_l, q_t >$. Où $q_{th} =<$ "*Décisionnel*", "*Base de données*", "*Programmation*"$>$, $q_l =<$"*France*"$>$ et $q_t =<2010>$ sont les sous requêtes d'analyse spécifiques aux dimensions THÉMATIQUE, LOCALISATION et TEMPS. Chaque document d est représenté dans trois espaces vectoriels $< \overrightarrow{d_{Dim_{th}}}, \overrightarrow{d_{Dim_l}}, \overrightarrow{d_{Dim_t}} >$, chacun relatif à une dimension du cube de textes. Les CVs sont agrégés à travers l'opérateur *ORank*.

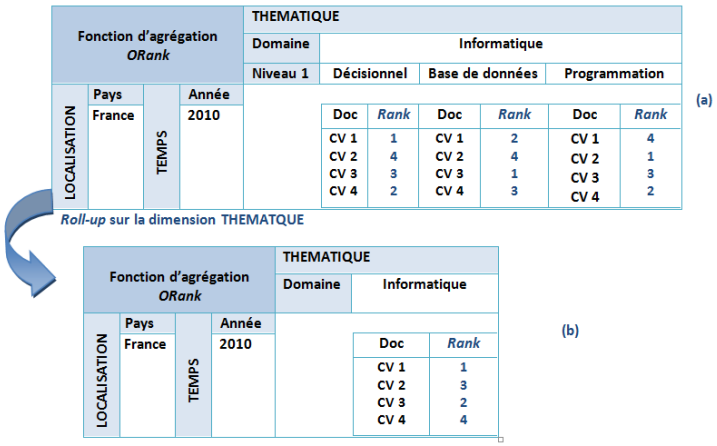

FIG. 4 – *Exemple d'une analyse navigationnelle sur un Cube de textes*

Le décideur obtient des agrégats sur trois compétences : *Décisionnel*, *Base de données* et *Programmation*. Par la suite, le décideur effectue un forage vers le haut (figure 4 (b)) pour avoir une vue plus générale sur la dimension THÉMATIQUE. Le niveau de détail dans la hiérarchie passe du niveau actuel l_i au niveau directement supérieur l_{i+1}. Ainsi les compétences *Décisionnel*, *Base de données* et *Programmation* sont regroupées en un seul groupe représentant les compétences dans le domaine *Informatique*. Les documents sont agrégés selon une nouvelle vue représentant les CVs de l'année 2010 pour le domaine *informatique* en *France*.

5 Expérimentations

Afin de valider notre approche, 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 1000 documents (CVs). Ces derniers représentent des candidatures pour un Master 2 en informatique décisionnelle, reçus lors d'une compagne de recrutement dans notre établissement.

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 conversion du texte en minuscule.
   - L'élimination des mots vides à partir d'une liste (*StopList*) (exemples : de, le, la, dans, etc.).
   - La *Lemmatisation* des termes en utilisant l'outil morphosyntaxique *tree tagger* [3]
2. Alimentation des dimensions du cube de textes : pour chaque dimension sémantique, nous avons conçu une hiérarchies de concepts sous forme d'un arbre XML. Pour parcourir un arbre XML, nous avons utilisé l'api java Open Source DOM4J [4].
   - Dimension THÉMATIQUE : elle est alimentée à partir du portail thématique de l'encyclopédie libre Wikipédia [5]. La hiérarchie extraite concerne la catégorie informatique qui comprend 43 sous catégories. Chacune d'elles comporte une arborescence de sous catégories.

1. http ://fr.wikipedia.org
2. http ://www.geonames.org
3. http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
4. http ://dom4j.sourceforge.net
5. http ://fr.wikipedia.org/wiki/Catégorie :Informatique

- Dimension LOCALISATION : elle est alimentée à partir d'une ontologie géographique *GeoNames*[6] qui est une collection géographique comportant environ 6 millions de localisations classées en 11 catégories différentes.
- Dimension TEMPS : cette dimension peut être représentée de différentes manières. En effet, un CV comprend plusieurs dates, la dimension TEMPS peut être représentée par les dates de réception des CVs, par les dates d'obtention des diplômes ou encore par les dates relatives aux compétences professionnelles (expériences), etc. Pour faciliter l'implémentation, nous avons choisi de représenter la dimension TEMPS par les dates de réception des CVs.

3. Tests effectués

   Nous avons réalisé deux types de tests, le premier concerne la mesure d'analyse textuelle proposée. Nous avons effectué une comparaison entre nos résultats et ceux d'une mesure basée sur une pondération *Tf* classique (Lin et al., 2008). Le deuxième test permettait d'évaluer l'opérateur d'agrégation *ORank*. Nous avons testé des requêtes d'analyse sur le *TCube*. Les détails sont présentés dans la section qui suit.

5.3 Résultats et discussions

Nous présentons dans le tableau 2 une comparaison des résultats que nous avons obtenus en appliquant, sur un document d pris aléatoirement, d'une part la mesure standard *TF* et d'autre part notre mesure d'analyse textuelle *M*.

Mesure *Tf*	Notre *mesure d'analyse textuelle M*
$M < \overrightarrow{d}$ = donnée (0.0348), base (0.0348), statistique (0.0290), informatique (0.0232), lyon (0.0232), analyse (0.0174), programmation (0.0174), compétence (0.0116), mention (0.0116), professionnel (0.0116), système (0.0116), lumière (0.0116). >	$M < \overrightarrow{d_{Dim_{th}}}$ = informatique (0.0239), statistique (0.0293), programmation (0.0180), réseau (0.0119), décisionnel (0.0065), php (0.0059), java (0.0059), r (0.0059), sql (0.0059), access (0.00598), base donnée (3.5856E-5), bureautique (3.5856E-5). $\overrightarrow{d_{Dim_l}}$ = lyon (0.0232), rhône-alpes (0.0005), france (0.0125E-3) $\overrightarrow{d_{Dim_t}}$ = 13 Avril 2010 (1.0) >

TAB. 2 – *Comparaison entre la mesure TF et notre mesure d'analyse textuelle M*

Dans la mesure d'analyse basée sur une pondération par la méthode *Tf*, les termes ayant les plus grand poids ne sont pas tous pertinents, exemples : *donnée*, *lumière*, *système*, etc. Notre mesure d'analyse textuelle basée sur l'exploitation de plusieurs hiérarchies de concepts pour l'extraction des informations sémantiques des documents et d'une technique de propagation de pertinence, permet une meilleure prise en charge de la sémantique des documents.

6. http ://www.geonames.org/ontology/

Pour évaluer notre opérateur d'agrégation, nous avons demandé à un décideur de poser des requêtes d'analyse.

Nous considérons une requête d'analyse où le décideur veut analyser les candidatures (CVs) pour : TEMPS = 2010, LOCALISATION = "Lyon" et THÉMATIQUE= "Informatique décisionnelle" sans mettre de préférences. Le tableau 3 montre les résultats sur un échantillon de 10 documents (CVs).

$ORank(d)$	Documents	$Sim(d_{Dim_{th}}, q_{th})$	$Sim(d_{Dim_l}, q_l)$	$Sim(d_{Dim_t}, q_i)$
1(2.6314)	CV 8	0.6314	1.0	1.0
2(2.5070)	CV 4	0.5070	1.0	1.0
3(2.4901)	CV 1	0.4901	1.0	1.0
4(2.4856)	CV 10	0.4856	1.0	1.0
5(2.4761)	CV 7	0.5477	0.9284	1.0
6(2.4648)	CV 9	0.4648	1.0	1.0
7(2.3281)	CV 6	0.3281	1.0	1.0
8(2.2817)	CV 2	0.5746	0.7071	1.0
9(2.2778)	CV 3	0.5707	0.7071	1.0
10(2.2734)	CV 5	0.3790	0.8944	1.0

TAB. 3 – *Résultats de l'opérateur d'agrégation ORank sur le corpus des CVs*

Les résultats affichés dans le tableau 3 montrent que notre opérateur d'agrégation combine les résultats des trois espaces vectoriels relatifs à chaque dimension du cube, tout en respectant le classement des documents dans les trois dimensions.

Prenant l'exemple du document CV 7, le décideur n'ayant pas de préférences dans cette requête :

$\alpha_{Dim_{th}} = \alpha_{Dim_l} = \alpha_{Dim_t} = 1 \Rightarrow ORank(d) = 1 \times 0.5477 + 1 \times 0.9284 + 1 \times 1.0 = 2.4761$

où $\alpha_{Dim_{th}}, \alpha_{Dim_l}, \alpha_{Dim_t}$ sont les coefficients liés aux dimensions THÉMATIQUE, LOCALISATION et TEMPS. Ce résultat fait que ce document (CV 7) est classé cinquième sur l'ensemble des documents.

Le décideur veut analyser la requête précédente, cette fois ci avec des préférences de 50% pour la dimension THÉMATIQUE et 25% pour chacune des dimensions TEMPS et LOCALISATION. Le tableau 4 montre les résultats de l'opérateur *ORank* sur le même échantillon.

En posant ses préférences, le décideur a accordé plus d'importance à la dimension THÉMATIQUE au détriment des deux autres. Notre opérateur d'agrégation *ORank* prend en compte les préférences du décideur et retourne un classement en faveur de la dimension THÉMATIQUE.

Par exemple, l'agrégation du document CV 7 est calculée cette fois ci comme suit :

- $\alpha_{th} = 0.5 \times 3 = 1.5$
- $\alpha_l = \alpha_t = 0.25 \times 3 = 0.75$
- $ORank(d) = 1.5 \times 0.5477 + 0.75 \times 0.9284 + 0.75 \times 1 = 2.2678$

Ce document a pris de l'importance en raison de la multiplication de sa similarité avec la requête pour la dimension THÉMATIQUE par le coefficient $\alpha_{th} = 1.5$ (formule 5). Tandis que la similarité du document avec la requête d'analyse pour les dimensions LOCALISATION et TEMPS sont les deux multipliées par des coefficients $\alpha_l = \alpha_t = 0.75$ inférieur à celui de la dimension THÉMATIQUE.

$ORank(d)$	Documents	$Sim(d_{Dim_{th}}, q_{th})$	$Sim(d_{Dim_l}, q_l)$	$Sim(d_{Dim_t}, q_i)$
1(2.4471)	CV 8	0.6314	1.0	1.0
2(2.2678)	CV 7	0.5477	0.9284	1.0
3(2.2605)	CV 4	0.5070	1.0	1.0
4(2.2351)	CV 1	0.4901	1.0	1.0
5(2.2284)	CV 10	0.4856	1.0	1.0
6(2.1972)	CV 9	0.4648	1.0	1.0
7(2.1422)	CV 2	0.5746	0.7071	1.0
8(2.1363)	CV 3	0.5707	0.7071	1.0
9(1.9921)	CV 6	0.3281	1.0	1.0
10(1.9893)	CV 5	0.3790	0.8944	1.0

TAB. 4 – *Résultats de l'opérateur d'agrégation ORank avec préférences*

En comparant les résultats montrés dans les tableaux 3 et 4, nous pouvons voir que les préférences du décideur ont permis de changer le classement. le CV 7 qui était classé cinquième pour la première requête sans préférences est devenu deuxième après introduction des préférences.

Après examination des CVs agrégés par notre opérateur *ORank*, le décideur a jugé les résultats d'analyse comme étant satisfaisants par rapport à ses requêtes d'analyse.

6 Conclusion

Dans cet article, nous avons proposé une approche d'analyse OLAP de données textuelles. Nous avons défini un cube de textes nommé *TCube*. Nous avons associé à notre cube de textes une mesure d'analyse textuelle où, chaque document est représenté par des vecteurs de concepts pondérés, un pour chaque dimension. Notre cube, comporte un nouveau type de dimension pour représenter les données textuelles appelé *dimension sémantique*. Celle-ci est représentée par une hiérarchie de concepts, à travers laquelle une propagation de pertinences est effectuée. Nous avons proposé un opérateur d'agrégation noté *ORank*, qui agrège les documents par un classement, par rapport aux espaces vectoriels qui les représentent. Nous avons effectué des tests sur une collection de CVs reçus lors d'une compagne de recrutement dans notre établissement. Les résultats d'analyse obtenus étaient jugés satisfaisants par le recruteur.

En perspectives, une amélioration de l'implémentation de la dimension TEMPS est envisagée. Pour cela, nous pensons à une extraction des dates à partir des documents, à travers une segmentation thématique de ces derniers. De plus, nous prévoyons d'effectuer d'autres expérimentations, d'une part pour mesurer la qualité des termes retenus par notre mesure d'analyse textuelle, d'autre part, pour évaluer la qualité des résultats d'analyse. Enfin, nous planifions une évaluation de notre approche sur de gros volumes de données textuelles.

Références

Asfari, O. (2008). Modèle de recherche contextuelle orientée contenu pour un corpus de documents xml. *Conference on Information Retrieval and Applications, CORIA 2008*, 377–384.

Ben-Messaoud, R., O. Boussaid, et S. Rabaseda (2004). A new olap aggregation based on the ahc technique. *Workshop on Data Warehousing and OLAP 7*, 65–72.

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.

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.

Park, B. et I. Song (2011). Toward total business intelligence incorporating structured and unstructured data. *International Workshop on Business intelligencE and the WEB BEWEB 2*, 12–19.

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

Pinel-Sauvagnat, K. et M. Boughanem (2006). Conférence en recherche information et applications. *Propositions pour la pondération des termes et l'évaluation de la pertinence des éléments en recherche d'information structurée 6*, 77–97.

Ravat, F., O. Teste, et R. Tournier (2007). Olap aggregation function for textual data warehouse. *International Conference on Enterprise Information Systems 1*, 151–156.

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. et M. McGill (1983). *Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

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

Data warehousing technologies and On-Line Analytical Processing (OLAP) have generated methodologies for the analysis of structured data. However, they are not appropriate to handle textual data analysis. In this paper, we propose a text cube model for textual data, called *TCube*, which includes several semantic dimensions to better consider the semantics of text data. The attributes of each semantic dimension are grouped into a hierarchy of concepts extracted from domain ontology used as an external source. Our text cube model includes a new text analysis measure based on both an OLAP-adapted vector space model together with relevance propagation technique. It is also associated with a new aggregation function denoted ORank (*OLAP-Rank*) for a textual data aggregation in OLAP environment. The preliminary results of our experimental study show the importance of our approach.