Enrichissement sémantique de documents XML représentant des tableaux

Fatiha Saïs[1], Hélène Gagliardi[1]
Ollivier Haemmerlé[1,2], Nathalie Pernelle[1]

[1]LRI (UMR CNRS 8623 - Université Paris-Sud) / INRIA (Futurs), Bâtiment 490,
F-91405 Orsay Cedex, France
{prénom.nom}@lri.fr
http ://www.lri.fr/iasi
[2]UMR INAP-G/INRA BIA, 16 rue Claude Bernard,
F-75231 Paris Cedex 05, France
Ollivier.Haemmerle@inapg.fr
http ://www.inapg.inra.fr/ens_rech/mathinfo/index.html

Résumé. Ce travail a pour objectif la construction automatique d'un entrepôt thématique de données, à partir de documents de format divers provenant du Web. L'exploitation de cet entrepôt est assurée par un moteur d'interrogation fondé sur une ontologie. Notre attention porte plus précisément sur les tableaux extraits de ces documents et convertis au format XML, aux tags exclusivement syntaxiques. Cet article présente la transformation de ces tableaux, sous forme XML, en un formalisme enrichi sémantiquement dont la plupart des tags et des valeurs sont des termes construits à partir de l'ontologie.

Mots-clés : extraction de connaissances, entrepôt, ontologie, XML, Web.

1 Introduction

Le travail que nous présentons dans cet article est mené, en collaboration avec quatre partenaires[1], dans le cadre du projet e.dot (Entrepôt de Données Ouvert sur la Toile). Ce projet vise à permettre la construction automatique d'entrepôts thématiques de données stockées au format XML, alimentés par des données extraites du Web. Le domaine d'application choisi est la prévention du risque microbiologique (listeria, salmonelle, etc.) dans les aliments. Ce domaine présente un enjeu de santé publique mais également un enjeu industriel majeur. Notre entrepôt de données XML permet de compléter une base de données relationnelle et une base de graphes conceptuels préexistantes, contenant des données scientifiques et industrielles. Ces deux bases, interrogées de manière uniforme par le système MIEL [Buche *et al.*2004], ont été développées dans le cadre du projet Sym'Previus [sym] qui vise à construire un outil de prévision du comportement des germes pathogènes dans les aliments. La version étendue de MIEL interrogeant les deux bases existantes et l'entrepôt XML s'appelle MIEL++.

L'enchaînement des tâches qui constituent le projet e.dot est le suivant : à partir du Web, l'extraction automatique de documents de formats divers (html, pdf, excel, etc.) est réalisée par un robot explorateur du Web appelé *crawler*. Ces documents sont

[1]IASI-Gemo (LRI), Verso-Gemo (INRIA-Futurs), INAP-G/INRA et la société Xyleme

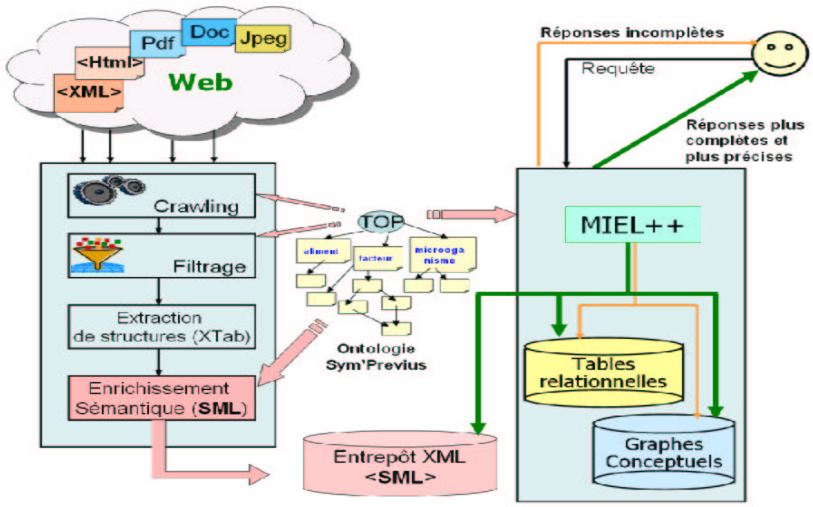

FIG. 1 – Architecture fonctionnelle du projet e.dot

alors traités par un module de filtrage qui permet de ne conserver que des documents présentant un niveau de pertinence satisfaisant par rapport au domaine d'application. Nous nous focalisons sur les documents qui contiennent des tableaux de données ; nous avons en effet constaté, lors de discussions avec des experts du domaine d'application, que les tableaux présents dans les publications scientifiques contiennent généralement des informations synthétiques et fiables. Ces structures de tableaux sont extraites et représentées au format XTab, qui est une représentation XML permettant de décrire des structures génériques de tableaux indépendantes du domaine et du format du document initial. Ces documents sont ensuite enrichis sémantiquement et stockés dans l'entrepôt.

C'est précisément cette phase d'enrichissement sémantique – uniquement fondée sur l'ontologie Sym'Previus – que nous allons présenter. Elle permet d'interpréter les valeurs et les lignes du tableau en fonction des termes et des relations sémantiques présents dans l'ontologie. Ce traitement automatique conduit parfois à des absences d'interprétation ou des incertitudes, mais l'objectif est de conserver le plus d'informations possibles du domaine dans l'entrepôt thématique, même si leur interprétation est imparfaite. Pour cela, nous avons défini le formalisme de représentation SML – *Semantic Markup Language* – fondé sur le langage XML, qui permet de conserver de telles interprétations. Le document SML résultant de l'enrichissement permet au moteur d'interrogation de fournir à l'utilisateur des informations complémentaires qui peuvent lui permettre de résoudre ces problèmes d'interprétation ou d'estimer la validité des réponses fournies.

Nous présentons brièvement, dans la partie 2, le format XTab et l'ontologie Sym'Previus. Nous expliquons ensuite sur un exemple simple, les objectifs de la tâche d'enrichissement sémantique. Nous étudions alors la façon dont les relations sémantiques présentes dans le tableau sont extraites et comment la représentation XML finale du document est construite, alimentée et exploitée par le moteur d'interrogation. Enfin, nous concluons et traçons quelques perspectives.

2 Prérequis

Un document XTab et l'ontologie du domaine sont fournis en entrée du module d'enrichissement sémantique. Nous les présentons maintenant.

2.1 Le format XTab : représentation générique de tableaux

Afin de fournir en entrée au module d'enrichissement sémantique des documents présentant une certaine homogénéité, une première étape consiste à transformer les tableaux d'origine – extraits à partir de documents html, pdf, etc provenant du Web – en documents XML. Le format XTab défini dans le cadre du projet e.dot [e.dot2004] permet de représenter un tableau en XML, en le structurant en lignes, chaque ligne étant elle-même structurée en un ensemble de cases. Cette représentation est indépendante du domaine d'application choisi. Les informations contenues dans un document XTab sont essentiellement le titre du tableau, les titres des colonnes et les valeurs des cases du tableau. La transformation des documents hétérogènes au format XTab est effectuée par le module *Any2XTab*. Les tableaux de structures plus complexes (imbrication de tableaux par exemple) nécessitent l'utilisation d'heuristiques afin de les ramener à une structure tabulaire simple. Ces heuristiques ne sont pas présentées dans cet article. La conversion au format XTab du tableau de la figure 2 est présentée en figure 3.

Article	**Valeur de pH**
Champignon d'ormeau	5.00
Crabe	6.60

FIG. 2 – *pH approximatif des produits alimentaires*

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<tableau><titre> <titre-tableau>pH approximatif des produits alimentaires</titre-tableau>
<titre-col>Article</titre-col> <titre-col>Valeur de pH</titre-col></titre> <nb-col>2</nb-col>
<contenu>
<ligne><case>champignon d'ormeau</case> <case>5.00</case> </ligne>
<ligne><case>crabe</case> <case>6.60</case> </ligne>
</contenu></tableau>
```

FIG. 3 – *Représentation d'un tableau en XTab*

2.2 Présentation de l'ontologie

Dans le cadre du projet Sym'Previus [sym], une *ontologie* contenant l'ensemble des connaissances du domaine du risque alimentaire a été développée. Elle est notamment composée :

1. d'une *taxonomie de termes* du domaine, hiérarchisés selon la relation de spécialisation. Cette taxonomie contient 428 termes couramment utilisés dans le domaine de la prévention du risque microbiologique dans les aliments (noms d'aliments, de germes pathogènes, de facteurs expérimentaux...) ;
2. d'un ensemble de termes rattachés aux termes de la taxonomie par une relation de proximité sémantique (*"produits surgelés"*, qui n'appartient pas à la taxonomie, est considéré comme sémantiquement proche de *"surgelés"* qui y appartient) ;
3. d'un ensemble de traductions en anglais de termes de la taxonomie ;

4. d'un schéma relationnel contenant les signatures des relations entre termes de la taxonomie ; $attrs(r)$ est l'ensemble des attributs de la signature de la relation r. Par exemple, la relation *alimentFacteurMicroorganisme* a pour signature *(aliment, facteur, microorganisme).*

3 Présentation de la tâche d'enrichissement

Afin de pouvoir exploiter des données contenues dans les tableaux extraits de documents provenant du Web, et notamment de les interroger par le biais de notre moteur de requêtes qui se fonde sur l'ontologie Sym'Previus, il est nécessaire de les exprimer en utilisant au maximum le vocabulaire du domaine puisé dans l'ontologie.

Nous avons donc fait le choix d'enrichir les documents XTab en leur ajoutant des tags et des valeurs provenant de l'ontologie. Plus précisément, nous avons défini le formalisme de représentation SML – *Semantic Markup Language* – qui permet de conserver la structure en lignes du tableau, de la même manière que dans les documents XTab. Par contre, les lignes ne sont plus représentées par un ensemble de cases mais par un ensemble de relations portant sur les colonnes du tableau. Une première étape du travail consiste donc à identifier de telles relations.

```
<tableau> <titre><titre-tableau>pH approximatif des produits alimentaires </titre-
tableau>
<titre-col> Article </titre-col> <titre-col> Valeur de pH </titre-col>... </titre>
<contenu>
<ligneRel> <alimentPH>
<aliment><finalVal>Champignon</finalVal>
<origineVal>champignon d'ormeau </origineVal> </aliment> <ph><finalVal/>
<origineVal>5.00</origineVal></ph> </alimentPH> </ligneRel>
<ligneRel> <alimentPH>
<aliment><finalVal>Crabe</finalVal><origineVal>crabe</origineVal></aliment>
<ph> <finalVal/><origineVal>6.60</origineVal> </ph>
</alimentPH> </ligneRel>
</contenu> </origine> </tableau>
```

FIG. 4 – *Représentation simplifiée du tableau de la figure 3 en SML*

Considérons par exemple la relation *alimentPH* qui appartient à l'ontologie Sym'Previus, et dont la signature est $(aliment, pH)$; cette relation permet de mettre en relation un nom d'aliment avec une valeur de pH. L'objectif de notre travail d'enrichissement sémantique est, à partir d'un document XTab, comme par exemple le document présenté dans la figure 3, d'obtenir un document SML tel que présenté dans celui de la figure 4. Dans ce document SML, la relation *alimentPH* reconnue dans le tableau en XTab et qui porte sur la première et la deuxième colonne, est représentée après son instanciation par les valeurs qu'elle représente.

Cette phase d'instanciation consiste essentiellement à chercher à associer un (ou plusieurs) termes(s) de la taxonomie de l'ontologie à chacune des valeurs apparaissant dans les cases du tableau. Dans le meilleur des cas, la valeur appartient à la taxonomie. Dans le cas contraire, nous utilisons des opérateurs de mise en correspondance ou – de *mapping* – afin de trouver un ou plusieurs termes de la taxonomie proches de cette valeur.

En ce qui concerne les valeurs présentes dans la première colonne, la valeur *crabe* a été trouvée "telle quelle" dans l'ontologie. Par contre, la valeur *Champignon d'ormeau* n'a pas été trouvée dans l'ontologie. La valeur *Champignon*, trouvée grâce à

un opérateur de mapping, apparaît comme valeur de l'élément SML $< finalVal >$. La valeur d'origine est conservée afin d'être visualisable lors d'une interrogation : elle apparaît comme valeur de l'élément $< origineVal >$.

La représentation d'un document SML (par exemple le document de la figure 4) est conforme à la DTD (Définition de Type de Document) SML définie dans [e.dot2004]. Cette DTD décrit la nature et l'organisation des éléments apparaissant dans les instances de ces documents.

4 Extraction des relations à partir des documents XTab

L'extraction des relations contenues dans un tableau s'effectue en deux étapes. Nous cherchons tout d'abord à identifier chaque colonne en lui associant un terme de la taxonomie. Ensuite, nous tentons de reconnaître les relations du schéma relationnel de l'ontologie dans lesquelles interviennent des termes identifiés à la première étape.

4.1 Identification des colonnes

L'identification des colonnes d'un tableau consiste en l'attribution à chacune d'elles d'un terme appelé *terme-attribut* qui représente ses valeurs.

Définition 1 *Un* terme-attribut *est un terme de la taxonomie apparaissant au moins une fois comme attribut dans la signature d'une relation du schéma relationnel de l'ontologie. Nous notons* attrsSchRel *l'ensemble des termes-attributs.*

L'association d'un terme-attribut à une colonne s'opère en cherchant dans un premier temps à repérer des valeurs présentes dans la colonne qui appartiennent à la taxonomie ou dont le libellé est inclus dans un terme de la taxonomie. On cherche alors les termes-attributs qui subsument la plupart des valeurs. Tout d'abord, on n'associe un terme-attribut à la colonne que si la proportion des valeurs qu'il subsume est supérieure à un seuil donné. On dit alors que le terme-attribut est candidat :

$$TACandidat(C, seuil) = \{t \mid t \in attrsSchRel \text{ et } \frac{|desc(t, C)|}{|C|} \geq seuil\}$$

avec *desc(t,C)* un prédicat définissant l'ensemble des valeurs[2] de la colonne C subsumées par le terme t.

Parmi ces termes-attributs candidats, on ne retient que les termes-attributs les plus spécifiques qui subsument le plus de valeurs. On note par *TARep* l'ensemble de ces *termes-attributs représentants* :

$$TARep(C, seuil) = \{t \mid t \in TACandidat(C, seuil),$$
$$\neg\exists\ t' \text{ tel que } t' \in TACandidat(C, seuil) \text{ et } |desc(t', C)| > |desc(t, C)|,$$
$$\neg\exists\ t'' \text{ tel que } t'' \in TACandidat(C, seuil) \text{ et } |desc(t'', C)| = |desc(t, C)| \text{ et } t'' \preceq t\}$$

S'il y a plusieurs termes-attributs représentants, nous prenons le premier trouvé. Mais notre expérience nous a montré que ce cas de figure est très rare si le seuil est suffisamment élevé. Si par le mécanisme présenté ci-dessus, la recherche de termes-attributs

[2] $v \preceq t$: il ne s'agit pas forcément de la valeur v elle-même ; il peut s'agir également d'un terme trouvé dans l'ontologie grâce à un opérateur de mapping d'inclusion (voir la définition des opérateurs de mapping dans la section 5.2)

est infructueuse, nous cherchons à identifier un terme-attribut dans le titre de la colonne du tableau. Nous cherchons d'abord dans les valeurs des colonnes plutôt que dans leurs titres, parce que l'expérience nous a montré que les titres des colonnes, s'ils sont disponibles, contiennent fréquemment des ambiguïtés. Au contraire, si l'on est capable d'identifier un nombre de valeurs suffisamment important, l'identification du bon terme-attribut pour la colonne est vraisemblable. En cas d'échec, nous conservons la colonne dans le document SML en lui associant le nom générique *attribut*.

Produit	**Qte**	**Lipides**	**Nombre de calories**
merlan au citron	100 g	7.8 g	92 kcal
crabe de terre	150 g	11.25 g	192 kcal
poulet	250 g	18.75 g	312 kcal

FIG. 5 – *Compositions nutritionnelles des aliments*

Considérons le tableau de la figure 5 : les valeurs *crabe de terre* et *poulet* peuvent être rapprochées des termes de l'ontologie dont le plus petit généralisant commun est le terme-attribut *Aliment*. En supposant que le seuil soit fixé à 0.5, le terme-attribut associé à la première colonne du tableau est *Aliment* (2/3 étant supérieur au seuil). Par contre, du fait de la présence des valeurs numériques et de l'abréviation *Qte*, aucun terme-attribut ne peut être associé à la deuxième colonne ; l'attribut générique *attribut* lui est alors associé. En revanche, les titres des colonnes *Lipides* et *Nombre de calories* ont permis d'identifier les deux termes-attributs *lipide* et *calorie* par un mécanisme d'inclusion de chaînes de caractères.

Définition 2 *un schéma* TabSch *d'un tableau* Tab*, noté* TabSch(Tab)*, consiste en un ensemble fini de couples (colonne, terme-attribut) où chaque couple représente pour une colonne et pour un seuil donné le terme-attribut lui correspondant.*

$TabSch(Tab) = \{(c,t) \mid t \in TARep(c, seuil)\ ou\ [(t = attribut)\ et\ TARep(c, seuil) = \emptyset]\}$

Dans le tableau de la figure 5 : *TabSch(Tab2)={(1,aliment) (2,attribut) (3,lipide),(4,calorie)}*

4.2 Identification des relations contenues dans le schéma du tableau

Nous présentons maintenant le mécanisme qui permet d'associer une ou plusieurs relations du schéma relationnel à un schéma du tableau.

Cas d'une relation totalement représentée : nous disons qu'une relation du schéma relationnel est *totalement représentée* dans le tableau si tous ses attributs subsument des termes-attributs apparaissant dans le schéma du tableau.

Cas d'une relation partiellement représentée : nous disons qu'une relation est *partiellement représentée* dans le schéma du tableau si elle ne l'est pas totalement, mais qu'au moins deux[3] de ses attributs subsument des termes-attributs apparaissant dans le schéma du tableau.

Cela signifie que tous les attributs de la relation ne figurent pas dans le schéma du tableau. Nous avons fait ce choix pour pouvoir prendre en compte les cas suivants :

[3]nous considérons la relation binaire comme la relation minimale pouvant être instanciée

- des colonnes non identifiées peuvent correspondre aux attributs non reconnus de la relation ;
- des attributs de relation peuvent correspondre à des valeurs constantes du tableau considéré (un tableau intitulé "croissance de la **listeria** dans des produits alimentaires" ne contiendra vraisemblablement pas de colonne "**microorganisme**") ;
- les experts ne formulent pas la même liste d'arguments pour une même relation sémantique.

Dans le cas où une relation est partiellement représentée, l'attribut qui n'appartient pas au schéma du tableau apparaît dans le document SML avec des valeurs nulles. Nous avons vu qu'il s'agissait parfois d'une valeur constante dans la légende ou le titre du tableau. Nous pouvons donc utiliser les opérateurs de mapping pour repérer cette valeur et la propager dans toutes les lignes du tableau en SML.

Aucune relation représentée : dans le cas où aucune des relations du schéma relationnel n'est (totalement ou partiellement) représentée dans le schéma du tableau, une relation générique appelée *relation* est générée dans le document SML résultat. Nous avons fait ce choix afin de pouvoir exploiter les informations du tableau même si aucune relation n'a été identifiée. Cela permet de garder dans le résultat de l'extraction les liens sémantiques existant entre certaines informations.

Dans cette même optique qui consiste à conserver des informations non identifiées, nous avons décidé d'adjoindre à chacune des relations identifiées, dans le document SML, l'ensemble des attributs génériques du schéma du tableau. Ceci est fait dans le cas des relations totalement ou partiellement représentées.

Supposons que l'ontologie contienne les relations suivantes : *alimentLipide*, *alimentCalorie*, *alimentPH*, ... et que les relations *alimentLipide* et *alimentCalorie* aient été normalisées par les experts du domaine aux 100g d'aliment. Dans le tableau de la figure 5, nous extrayons les relations suivantes :

- *alimentLipide*, totalement représentée par les valeurs des $1^{ère}$ et $3^{ème}$ colonnes ;
- *alimentCalorie*, totalement représentée par les valeurs des $1^{ère}$ et $4^{ème}$ colonnes.

Á chacune de ces relations, nous ajoutons un attribut générique qui contiendra les valeurs de la deuxième colonne qui n'a pas été identifiée.

En l'absence de l'attribut générique, par exemple la troisième ligne du tableau peut être interprétée par : "***100g*** *(normalisé par les experts) de poulet correspondent à 312 kcal*", alors que si on considère l'attribut générique, l'interprétation sera : "***250g*** *de poulet correspondent à 312 kcal*". Ainsi, le fait de ne pas afficher cette information non identifiée dans le résultat d'une requête portant sur une relation, peut conduire à une mauvaise interprétation de la relation sémantique entre les données, d'où l'intérêt des attributs génériques.

5 Alimentation et interrogation du tableau représenté en SML

Nous présentons maintenant l'ensemble des indicateurs que nous avons définis afin de rendre compte de la validité de l'interprétation des informations contenues dans les tableaux. Ces indicateurs permettent également d'indiquer la présence de renseignements complémentaires provenant du document original, renseignements qui pourront être visualisés lors de l'interrogation pour limiter les erreurs d'interprétation. Nous

montrons également comment les relations extraites sont instanciées par les valeurs du tableau. Enfin nous présentons sur un exemple comment un moteur d'interrogation exploite les indicateurs pour interroger les documents SML.

5.1 Des indicateurs de traitement exploitables par le moteur d'interrogation

Nous présentons les quatre principaux indicateurs de traitement qui sont des attributs XML attachés aux lignes ou aux attributs des relations. Le premier indicateur est lié à la structure des relations (présence ou absence d'attributs supplémentaires).

additionalAttr : il permet d'informer de la présence d'un ou plusieurs attributs supplémentaires représentant les colonnes du tableau qui n'ont pas pu être associées aux relations identifiées. Il est ajouté aux tags de ligne du document SML.
Par exemple dans le tableau de la figure 5, cet indicateur permet au moteur d'interrogation d'exploiter l'attribut générique *attribut* associé à la colonne *Qte*.

Les trois indicateurs suivants permettent de préciser les traitements de mise en correspondance entre la valeur d'origine et les termes de l'ontologie ; ils peuvent donc être exploités pour évaluer le risque d'une erreur de mapping. Ils sont ajoutés aux tags attributs des relations du document SML.

indOnto : il permet d'indiquer le traitement (inclusion, intersection,...) effectué lors de la recherche d'un terme correspondant à la valeur d'origine du tableau. Plusieurs opérateurs de mapping pouvant exister dans l'application, cet indicateur permet de moduler un degré de confiance relatif à l'enrichissement, en fonction de l'opérateur.

indTrans : si une traduction est nécessaire, nous exploitons en priorité celles de l'ontologie ; si cette étape échoue, il est possible d'utiliser un traducteur externe tel que SYSTRAN. Cet indicateur représente le type de traducteur utilisé. Ainsi, un traducteur externe a traduit la valeur Apple par "Apple" car il a considéré qu'il s'agissait de la marque de matériel informatique à cause de la majuscule. Cet exemple montre l'intérêt de l'indicateur *indTrans* qui peut influencer le degré de confiance que l'on affecte au résultat de cet enrichissement sémantique.

indProc : il indique si un traitement (traduction, inclusion ou intersection) a été nécessaire pour trouver un terme de l'ontologie correspondant à la valeur d'origine du tableau. Il peut être exploité pour visualiser la valeur d'origine le cas échéant.

Ces indicateurs de traitement sont conservés dans le document SML et peuvent être exploités par le moteur d'interrogation pour, par exemple, s'adapter et trouver d'autres réponses alternatives pour l'utilisateur en cas d'insatisfaction.

5.2 Instanciation des relations

Pour instancier les relations identifiées par les valeurs du tableau, nous recherchons pour chaque valeur le ou les termes de l'ontologie qui lui sont proches sémantiquement. Nous procédons par comparaison syntaxique de chaque valeur avec les termes de l'ontologie, grâce à des *opérateurs de mapping*. Pour définir ces opérateurs, nous considérons les valeurs et les termes comme des ensembles de mots lemmatisés[4] dans lesquels on

[4]Lemmatisation : c'est une procédure ramenant un mot portant des marques de flexion (e.g la forme conjuguée d'un verbe) à sa forme de référence dite *lemme* (e.g *les verbes à l'infinitif*).

supprime les mots vides[5].

Trois opérateurs de mapping ont été utilisés. Ils s'appuient sur des tests de comparaison entre ensembles de mots M_v et M_t correspondant respectivement à la *valeur* d'origine du tableau et au *terme* de l'ontologie auquel est comparée cette valeur.

1. **Égalité :** si ($M_v = M_t$) alors la valeur existe telle quelle dans l'ontologie.
2. **Inclusion :** si ($M_v \subset M_t$ ou $M_t \subset M_v$) alors *t correspond* à la valeur *v*.
3. **Intersection :** si ($M_v \cap M_t \neq \emptyset$) alors *t correspond* à la valeur *v*.

Pour effectuer un mapping, les trois opérateurs sont appliqués dans cet ordre et nous nous arrêtons au premier opérateur dont l'application a réussit.

Nous présentons, dans la figure 6, un extrait du document SML, généré d'une manière *automatique* et représentant le résultat de l'enrichissement sémantique du document XTab de la figure 5. Nous mettons l'accent sur les indicateurs de traitement au fur et à mesure de la description. Dans ce document SML nous trouvons les éléments suivants :

```
<tableau> <titre-tableau>Compositions nutritionnelles des aliments </titre-tableau>
<titre-col> Produit </titre-col> <titre-col>Qte</titre-col>
<titre-col>lipides</titre-col><titre-col>nombre de calories</titre-col><nb-col> 4 </nb-col>
<contenu>
<ligneRel additionalAttr="yes">
<alimentLipide><aliment indOnto="intersection" indTrans="none" indProc="yes">
<finalVal>merlan à la provençale</finalVal> <finalVal>citron vert</finalVal>
<finalVal>filets de merlan</finalVal> <origineVal>merlan au citron</origineVal></aliment>
<lipide indOnto="notFound" indTrans="none" indProc="no"> <finalVal/>
<origineVal> 7.8 g</origineVal> </lipide>
<attribut indOnto="notFound" indTrans="none" indProc="no"> <finalVal/>
<origineVal> 100 g</origineVal></attribut>
</alimentLipide> <alimentCalorie> ... </alimentCalorie> </ligneRel>

<ligneRel additionalAttr="yes">
<alimentLipide> <aliment indOnto="inclusion" indTrans="none" indProc="yes">
<finalVal>crabe</finalVal> <origineVal>crabe de terre </origineVal> </aliment>
<lipide indOnto="notFound" indTrans="none" indProc="no"> <finalVal/>
<origineVal> 11.25 g</origineVal> </lipide>
<attribut indOnto="notFound" indTrans="none" indProc="no"> <finalVal/>
<origineVal> 150 g</origineVal></attribut>
</alimentLipide> <alimentCalorie> ... </alimentCalorie> </ligneRel>

<ligneRel additionalAttr="yes"> <alimentLipide >
<aliment indOnto="complete" indTrans="none" indProc="no">
<finalVal>poulet</finalVal> <origineVal>poulet </origineVal> </aliment> ... </alimentLipide>
<alimentCalorie> ... </alimentCalorie> </ligneRel>
</contenu> <id>http ://www.i-dietetique.com/</id> </tableau>
```

FIG. 6 – *Représentation SML du tableau des compositions nutritionnelles des aliments*

tableau : l'élément racine du document SML qui encapsule les éléments *id*, *titre-tableau*, *nb-col* et *contenu*.

Nous présentons l'élément principal du tableau qui est l'élément ***contenu*** : il représente le contenu du tableau structuré sous forme d'un ensemble de lignes – ou ***ligneRel***. Chaque ligne représente l'ensemble des relations sémantiques du schéma relationnel identifiées dans le tableau (comme par exemple ***alimentLipide et alimentCalorie***).

[5]Mots vides : il désigne tout élément d'un texte n'ayant pas de référence concrète ou notionnelle dans la réalité. Il s'agit des articles, des conjonctions, des prépositions, etc.

La représentation SML d'une relation consiste en l'ensemble de ses attributs apparaissant dans sa signature décrite dans le schéma relationnel de l'ontologie (e.g. *alimentLipide*(*aliment*, *lipide*)). Chaque attribut de relation représente l'ensemble des valeurs d'une colonne du tableau à laquelle cet attribut a été affecté. Chaque valeur est représentée par l'ensemble des termes de l'ontologie dont la sémantique est proche de la valeur considérée. Ces termes sont représentés dans un tag XML *finalVal*, comme par exemple *crabe*, terme dont la sémantique est proche de la valeur *crabe de terre*. Nous gardons également la valeur d'origine dans un tag XML *origineVal*.

Pour trouver des termes de l'ontologie correspondant aux valeurs du tableau, nous les mettons en correspondance par le biais des opérateurs de mapping, dont le type est conservé dans l'indicateur de traitement *indOnto*. Quand il s'agit d'un opérateur d'inclusion ou d'intersection nous mettons l'indicateur de traitement *indProc* à la valeur *yes*, comme par exemple le terme *"crabe"* qui est inclus dans la valeur *crabe de terre*. Il est donc retenu comme terme correspondant à cette valeur. Il peut y avoir plusieurs termes de l'ontologie correspondant à une même valeur[6] ,comme par exemple *"merlan à la provençale"*, *"citron vert"* et *"filets de merlan"*, trois termes de l'ontologie qui correspondent à la valeur *"merlan au citron"*. L'intérêt de l'ajout des indicateurs de traitement *indOnto* et *indProc* est reflété, par exemple, par la capacité à identifier des termes de l'ontologie dont la proximité sémantique est peu significative (e.g. *citron vert*) et pouvant être considérés comme correspondant à une valeur (e.g. *merlan au citron*).

5.3 Interrogation des documents SML

Pour interroger les données contenues dans les documents SML, des requêtes en XQuery ont été écrites. Elles s'appuient sur la DTD SML. Voici un exemple de requête où l'utilisateur cherche pour 100 g d'aliment *"crabe"* la quantité de *lipide* correspondante. L'évaluation de cette requête consiste à rechercher dans le document SML les sous-arbres – fragments SML – dont l'élément père est *alimentLipide* et contenant un élément *finalVal* ayant comme valeur *"crabe"*.

Les indicateurs *indOnto* et *indProc* sont exploités pour vérifier la validité de l'enrichissement sémantique apporté sur les données. Comme l'indicateur *additionnalAttr* vaut yes, le moteur d'interrogation affiche l'information complémentaire **150g**, mais également la valeur d'origine *"crabe de terre"* puisque *indProc* indique qu'un traitement a été effectué sur la valeur d'origine. Le résultat de l'évaluation de cette requête sur le document de la figure 6 est présenté en figure 7.

<tableau>
<titre> Compositions nutritionnelles des aliments </titre>
<aliment> **crabe de terre**</aliment> <lipide>**11.25 g**</lipide>
<validité>inclusion</validité>
<attributSupp>**150 g**</attributSupp> <catégorie>inconnue</catégorie> </tableau>

FIG. 7 – *Une structure possible du résultat de cette requête*

[6]des heuristiques permettant de sélectionner qu'un sous-ensemble des termes proposés par l'opérateur de mapping appliqué, sont utilisées.

6 Conclusion

Nous proposons dans cette approche une méthode permettant de réaliser un enrichissement sémantique automatique d'informations structurées sous forme de tableaux provenant du Web. Cet enrichissement est uniquement guidé par une ontologie. Nous avons souhaité que cet enrichissement sémantique soit automatique ce qui entraîne d'éventuelles imperfections dans le résultat. En effet, ce traitement n'identifie pas toujours toutes les données (relations, attributs ou valeurs) ; notre représentation permet alors de les conserver en vue d'une interrogation moins spécifique (à l'aide de mots-clés par exemple) ou pour être visualisés par le moteur d'interrogation pour améliorer l'interprétation de l'utilisateur. Ensuite, en cas d'ambiguïté, il est possible d'associer plusieurs termes de l'ontologie à une valeur ou d'associer plusieurs relations sémantiques pour un même ensemble de colonnes. Pour permettre au moteur d'interrogation d'adapter ses réponses ou d'évaluer leur validité, nous conservons la trace des traitements effectués grâce à un ensemble d'indicateurs. L'aspect générique de notre approche est garanti par le fait que la DTD SML peut être générée automatiquement à partir de l'ontologie. En effet, chaque tag SML est égal ou construit à partir d'une relation de l'ontologie et de sa signature. Les autres éléments sont introduits de manière systématique (e.g. contenu, ligneRel, origineVal ...).

L'approche que nous avons proposée est en cours d'expérimentation, dans le domaine du risque alimentaire, par le biais du prototype développé en Java. Pour l'interrogation des documents SML, nous avons écrit dans le language *XQuery* des requêtes qui exploitent certains indicateurs de traitement ajoutés aux documents SML, puis nous les avons testés grâce au moteur d'interrogation MIEL++.

Des approches telles que [Kushmerick2000], [Muslea *et al.*2001] et [Hsu et Dung1998], permettent aussi d'extraire des connaissances en apprenant des règles à partir d'un échantillon de documents annotés manuellement. Notre objectif est différent puisque nous souhaitons que notre approche soit complètement automatique, et uniquement guidée par l'ontologie. De plus, les documents que nous traitons pour alimenter l'entrepôt sont très hétérogènes et, contrairement à des approches telles que [Crescenzi *et al.*2002] et [Arasu et Garcia-Molina2003], nous ne pouvons pas nous fonder sur la découverte d'une structure commune pour découvrir des connaissances à partir d'un ensemble de documents homogènes. Les techniques que nous avons utilisées pour identifier les colonnes du tableau sont fondées en priorité sur les valeurs des colonnes. [Rahm et Bernstein2001] et [Doan *et al.*2003], ont montré que ces techniques donnaient de bons résultats dans le cadre de la recherche de correspondance de schémas pour les bases de données relationnelles ou XML. Dans notre cas, nous ne disposons pas du schéma des tableaux que nous traitons : nous devons l'inférer avant de chercher des correspondances avec les relations de l'ontologie.

Nous pouvons maintenant améliorer nos opérateurs de mapping soit en utilisant des ressources extérieures telles que WordNet, soit en utilisant des mesures de similarité plus élaborées [Robertson et Willett1998]. De plus, nous pouvons envisager l'utilisation d'outils linguistiques permettant de traiter le contenu du tableau (cases, titres) représenté d'une manière plus complexe. Nous souhaitons également tester la généricité de notre approche en traitant ce problème dans le cadre d'un autre domaine d'application.

Summary

This work aims at building automatically a thematic data warehouse composed of heterogeneous XML documents extracted from the Web. We focus on the data tables contained in these documents. This article presents how we enrich semantically those tables by means of tags and values coming from the ontology of the application.

Références

[Arasu et Garcia-Molina2003] Arvind Arasu et Hector Garcia-Molina. Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 337–348. ACM Press, 2003.

[Buche *et al.*2004] Patrice Buche, Juliette Dibie-Barthélemy, Ollivier Haemmerlé, et Mounir Houhou. Towards flexible querying of xml imprecise data in a dataware house opened on the web. In *Flexible Query Answering Systems (FQAS)*. Springer Verlag, june 2004.

[Crescenzi *et al.*2002] Valter Crescenzi, Giansalvatore Mecca, et Paolo Merialdo. Automatic web information extraction in the roadrunner system. In *Revised Papers from the HUMACS, DASWIS, ECOMO, and DAMA on ER 2001 Workshops*, pages 264–277. Springer-Verlag, 2002.

[Doan *et al.*2003] AnHai Doan, Ying Lu, Yoonkyong Lee, et Jiawei Han. Profile-based object matching for information integration. *Intelligent Systems, IEEE*, 18(5) :54–59, September/October 2003.

[e.dot2004] e.dot. Revue intermediare du projet e.dot. Web site, 2004. http ://www-rocq.inria.fr/verso/edot/Revue-29-Juin-04/.

[Hsu et Dung1998] Chun-Nan Hsu et Ming-Tzung Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Inf. Syst.*, 23(9) :521–538, 1998.

[Kushmerick2000] Nicholas Kushmerick. Wrapper induction : efficiency and expressiveness. *Artif. Intell.*, 118(1-2) :15–68, 2000.

[Muslea *et al.*2001] Ion Muslea, Steven Minton, et Craig A. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1-2) :93–114, 2001.

[Rahm et Bernstein2001] Erhard Rahm et Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4) :334–350, 2001.

[Robertson et Willett1998] A.M. Robertson et P. Willett. Applications of n-grams in textual information systems. In *Journal of Documentation*, pages 48–69, 1998.

[sym] http ://www.symprevius.net.