Actes de l'atelier

Fouille du Web

des 6èmes journées francophones

« Extraction et Gestion des Connaissances »

17 Janvier 2006 Lille

Sommaire

Comité de programme

Marie-Aude Aufaure, Supelec
Younnes Bennani, Université Paris-Nord, CNRS (LIPN)
Philippe Bretier, France-Telecom, Division R&D, Lannion, Laboratoire Easy
Valérie Camps, Université Paul Sabatier, Toulouse 3, CNRS (IRIT)
Rose Dieng, INRIA (Sophia-Antipolis)
Mohand Saïd Hacid, Université Claude Bernard, Lyon 1, CNRS (LIRIS)
Patrick Gallinari, LIP6, Université de Paris 6
Daniele Herin, Université de Montpellier, CNRS (LIRMM)
Christian Jacquemin, Université Paris-Sud, CNRS (LIMSI)
Yves Kodratoff, Université Paris-Sud, CNRS(LRI)
Nicolas Labroche, LIP6, Université de Paris 6
Sabine Loudcher, ERIC, Université de Lyon 2
Bénédicte Legrand, LIP6, Université de Paris 6
Denis Maurel, LI, Université François-Rabelais de Tours
Benjamin Nguyen, Université de Versailles, CNRS (PRiSM)
Fabien Picarougne, LINA, Polytech'Nantes
Chantal Reynaud, Université Paris-Sud, CNRS (LRI) & INRIA (Futurs)
Brigitte Safar, Université Paris-Sud, CNRS (LRI) & INRIA (Futurs)
Imad Saleh, PARAGRAPHE, Université de Paris 8
Max Silberztein, LASELDI, Université de Franche-Comté
Gilles Venturini, LI, Université François-Rabelais de Tours

Comité d'organisation

Chantal Reynaud, Université de Paris-Sud, CNRS (LRI) & INRIA (Futurs)
Gilles Venturini et Hanene Azzag, Université François-Rabelais de Tours.

Appel à communication pour un numéro spécial de RNTI :

Le but de cet atelier a été de réunir les chercheurs académiques et industriels autour de la thématique de la fouille du Web. Les articles ont été soumis directement dans leur version définitive, et 8 articles sur 13 ont été acceptés.

A la suite de cet atelier, nous lançons à un appel à communication s'adressant à tous et concernant un numéro spécial Fouille du Web de la revue RNTI, qui sera relayé de manière électronique (voir le site http://www.antsearch.univ-tours.fr/fw-egc06). La date limite de soumission sera fixée à fin avril. Toute personne ayant participée ou non à l'atelier peut soumettre ses travaux.

Classification hiérarchique et visualisation de pages Web

Hanene Azzag*, Christiane Guinot*,**
Gilles Venturini*

*Laboratoire d'Informatique de l'Université de Tours,
École Polytechnique de l'Université de Tours - Département Informatique,
64, Avenue Jean Portalis, 37200 Tours, FRANCE.
hanene.azzag@etu.univ-tours.fr, venturini@univ-tours.fr
http://www.antsearch.univ-tours.fr/webrtic
**CE.R.I.E.S.
20 rue Victor Noir, F-92521 Neuilly-sur-Seine Cedex.
christiane.guinot@ceries-lab.com

Résumé. Nous présentons dans cet article un nouvel algorithme de classification hiérarchique et non supervisée de documents noté AntTree$_{Sans-Seuil}$. Il utilise le principe d'auto-assemblage observé chez des fourmis réelles qui construisent des structures vivantes en se connectant progressivement les unes aux autres. Nous adaptons ces principes pour construire un arbre de documents permettant de générer automatiquement des sites portails. Dans un premier temps, nous avons testé et validé AntTree$_{Sans-Seuil}$ sur des bases de données textuelles, suivie d'une étude comparative avec la méthode CAH. Enfin, dans un second temps, nous introduisons un affichage d'arbre dans un environnement immersif en trois dimensions permettant d'explorer le site portail construit.

1 Introduction

La fouille de textes est l'analogue du domaine de la fouille de données mais pour le traitement de données textuelles Lebart et Salem (1994). En effet, il s'agit d'extraire automatiquement des connaissances à partir d'un ensemble de textes, en utilisant des méthodes de découverte comme les statistiques, l'apprentissage, l'analyse de données. Dans notre étude, nous nous intéressons à une problématique d'apprentissage non supervisé, dans laquelle une classification hiérarchique de textes est nécessaire. Il s'agit de la construction automatique de site portails pour le Web.

En effet un site portail est un moyen de recherche d'information au même titre qu'un moteur de recherche. La différence importante entre les deux vient du fait qu'un site portail regroupe généralement des informations sur un thème donné sous la forme d'une structure arborescente dans laquelle l'utilisateur va pouvoir se déplacer. Une première manière simple (d'un point de vue informatique) d'aborder la problématique de construction de sites portail est d'utiliser une approche manuelle Kumar et al. (2001). Le portail Yahoo ! utilise ce type de construction qui fait intervenir des personnes appelées ontologistes (ou encore *surfeurs* selon l'appellation de Yahoo !). Leur travail consiste à sélectionner les documents fournis par

des développeurs de sites ou par l'intermédiaire de robots d'exploration d'Internet et à les classer au bon emplacement dans une hiérarchie prédéfinie et pouvant contenir une centaine de milliers de thèmes. Ces approches sont souvent jugées comme très pertinentes car elles font appel à un jugement humain (plus efficace qu'un jugement automatique) mais sont fastidieuses et demandent des moyens très importants.

Cet article est consacré à l'étude de solutions originales qui s'inspirent des propriétés du vivant pour proposer des constructions automatiques de hiérarchies de données. Notre modèle utilise le principe d'auto-assemblage observé chez les fourmis réelles qui construisent des structures vivantes en se connectant progressivement à un support fixe puis successivement aux fourmis déjà connectées. Chaque fourmi artificielle représente une donnée à classer. Ces fourmis vont ensuite construire de manière similaire un arbre en appliquant certaines règles comportementales. Les déplacements et les assemblages des fourmis sur cet arbre dépendent de la similarité entre les données.

2 Auto-assemblage chez les fourmis réelles

Dans Anderson et al. (2002), les auteurs ont regroupé plusieurs phénomènes d'auto-assemblage que l'on peut observer chez les animaux, en particulier les insectes sociaux. Le modèle réel d'auto-assemblage qui nous a servi d'inspiration pour développer nos algorithmes de fourmis est basé sur l'étude réalisée sur les fourmis *Linepithema humile* et *Oecophylla Longinoda* par Sauwens (2000) et Lioni (2000) durant leur thèse de doctorat.

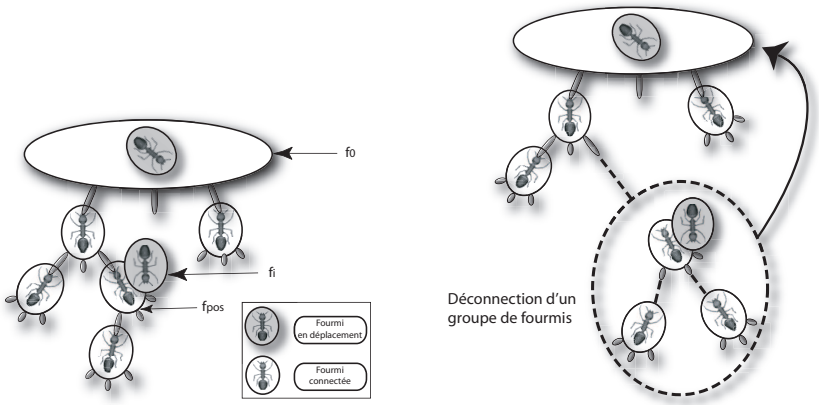
La formation de chaînes et de ponts chez les fourmis fileuses *Oecophylla* permet le franchissement d'obstacles naturels ou sert à la construction de nids. Quant aux fourmis *Linepithema humile* elles construisent des grappes d'ouvrières suspendues dans le vide dont la fonctionnalité est méconnue à ce jour. Ce phénomène de croissance des chaînes ou la construction de grappes fait appel à des comportements de base tout à fait similaires : des ouvrières gagnent la structure collective (chaîne ou grappe), et y séjournent un certain temps au cours duquel elles se déplacent, éventuellement s'immobilisent avant de quitter cette structure. Ce décrochage se traduit par le phénomène de résorption de chaînes observé chez les *Oecophylla longinoda* et la chute de gouttes de fourmis observé chez les *Linepithema humile*.

A partir de ces principes nous avons développé un modèle de règles comportementales pour des fourmis artificielles que nous avons précédemment utilisé pour développer d'autres types d'heuristiques pour la classification non supervisée hiérarchique Azzag et al. (2005a).

3 L'algorithme : AntTree$_{Sans-Seuil}$

Le principe d'AntTree$_{Sans-Seuil}$ est le suivant : chaque fourmi f_i, $i \in [1, N]$ (N est le nombre de données initiales) représente un nœud de l'arbre à assembler, c'est-à-dire une donnée à classer et l'arbre représente la structure que les fourmis vont construire.

Initialement toutes les fourmis sont positionnées sur un support noté f_0 (voir figure 1(a)). A chaque itération, une fourmi f_i est choisie dans la liste des fourmis triée au départ. Cette fourmi cherchera alors à se connecter sur sa position courante sur le support (f_0) ou sur une autre fourmi déjà connectée (noté f_{pos}). Cette opération ne peut aboutir que dans le cas où elle est suffisamment dissimilaire à f_+ (la fourmi connectée à f_0 ou f_{pos} la plus similaire à f_i).



(a) (b)

FIG. 1 – *Construction de l'arbre par des fourmis :les fourmis qui sont en déplacement sont représentées en gris et les fourmis connectées en blanc (a). Décrochage de fourmis dans AntTree$_{Sans-Seuil}$ (b)*

Dans le cas contraire, f_i se déplacera de manière déterministe dans l'arbre suivant le chemin le plus similaire indiqué par f_+. Le seuil permettant de prendre ces décisions ne va dépendre que du voisinage local.

L'algorithme 1 regroupe les règles comportementales suivant que f_i est positionnée sur le support ou sur une fourmi f_{pos} (on considère que $f_{pos} = f_0$).

Notons $T_{Dissim}(f_0)$ (et respectivement $T_{Dissim}(f_{pos})$) la valeur de la similarité minimum observée entre chaque couple de fourmis connectées au support f_0 (respectivement à f_{pos}). $T_{Dissim}(f_i)$ est automatiquement déterminé par la structure de l'arbre lui-même. La valeur de $T_{Dissim}(f_i)$ est comme suit :

$$T_{Dissim}(f_i) = Min\left(Sim_{j,k\in[1,n]}(f_j, f_k)\right) \quad (1)$$

Où $f_1 ... f_n$ représentent l'ensemble des fourmis connectées à f_i.

Ainsi une fourmi f_i se connectera à f_{pos} (respectivement à f_0) si et seulement si cette action diminue la valeur de $T_{Dissim}(f_0)$ (respectivement de $T_{Dissim}(f_{pos})$). En d'autres termes, ceci consiste à comparer f_i à f_+, dans le cas où les deux fourmis sont suffisamment dissimilaires entre elles ($Sim(f_i, f_+) < T_{Dissim}(f_0)$ ou $Sim(f_i, f_+) < T_{Dissim}(f_{pos})$) on connectera f_i à f_{pos} (respectivement à f_0) et dans le cas contraire, on déplacera f_i vers f_+. Dans notre algorithme on peut voir que pour chaque nœud de l'arbre la valeur de T_{Dissim} diminue. En effet à chaque fois qu'une fourmi arrive à se connecter elle baisse par définition la valeur du T_{Dissim} correspondant.

Il est évident que pour calculer la valeur de $T_{Dissim}(f_0)$ (respectivement $T_{Dissim}(f_{pos})$) il faut avoir au minimum deux fourmis connectées à f_0 ou à f_{pos} (suivant la position de f_i).

(1) **si** aucune ou une seule fourmi est connectée à f_{pos} **alors**
(2) connecter f_i à f_{pos} /* une nouvelle classe */
(3) **sinon**
(4) **si** 2 fourmis sont connectées à f_{pos} (uniquement le premier passage) **alors**
(5) **si** $Sim(f_i, f_+) < T_{Dissim}(f_{pos})$ **alors**
(6) /* f_+ la fourmi connectée à f_{pos} la plus similaire à f_i */
(7) /* $T_{Dissim}(f_{pos}) = Sim(f_1, f_2)$ où f_1, f_2 représente les deux fourmis connectées à f_{pos} et $f_+ = f_1$ ou f_2 */
(8) - déconnecter f_+ de f_{pos} (et récursivement toutes les fourmis connectées à f_+)
(9) - replacer toutes ces fourmis déconnectées sur le support f_{pos}
(10) - connecter f_i à f_{pos}
(11) **sinon**
(12) déplacer f_i vers f_+
(13) **finsi**
(14) **sinon**
(15) /* plus de deux fourmis sont connectées au support ou deux fourmis sont connectées au support et on est dans le deuxième passage */
(16) - soit $T_{Dissim}(f_{pos})$ la valeur de la similarité minimum observée entre chaque couple de fourmis connectées au support f_{pos}
(17) **si** $Sim(f_i, f_+) < T_{Dissim}(f_{pos})$ **alors**
(18) connecter f_i à f_{pos} /* f_i dissimilaire à f_+ */
(19) **sinon**
(20) déplacer f_i vers f_+
(21) **finsi**
(22) **finsi**
(23) **finsi**

ALG 1: AntTree$_{Sans-Seuil}$.

Par conséquent dans AntTree$_{Sans-Seuil}$ les deux premières fourmis vont automatiquement se connecter à leur position sans aucun test de validité. Ceci entraîne une connection «abusive» de la seconde fourmi (sans test). De ce fait nous avons décidé qu'une fois une troisième fourmi connectée (pour cette troisième fourmi nous sommes sûrs que le test de dissimilarité a bien réussi), nous déconnecterons une de ces deux premières fourmis. Plus précisément, sera décrochée celle qui est la plus similaire à la nouvelle fourmi connectée (voir point (7) dans l'algorithme 1).

Lorsqu'une fourmi f_i est déconnectée, toutes les fourmis connectées à elle se décrochent également. Toutes ces fourmis seront repositionnées sur le support (voir figure 1 (b)) pour de nouvelles simulations. Ensuite, elles vont chercher à se déplacer pour se connecter en utilisant les mêmes règles comportementales définies dans l'algorithme 1.

3.1 Propriétés de l'algorithme

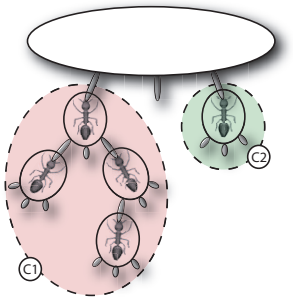


FIG. 2 – *Transformation de l'arbre en classes (partitionnement), ici deux classes sont générées C_1 et C_2.*

La conception d'AntTree$_{Sans-Seuil}$ permet de définir plusieurs propriétés. Tout d'abord, comme cela est représenté sur la figure 2, les sous-arbres placés directement sous le support constituent la classification "plane" trouvée par AntTree$_{Sans-Seuil}$, chaque sous-arbre correspondant à une classe constituée de toutes les données présentes dans ce sous-arbre. Cette classification peut être comparée à d'autres obtenues par des algorithmes non hiérarchiques par exemple comme ceux basés sur les centroïdes.

De plus, AntTree$_{Sans-Seuil}$ répond aux propriétés visées pour une bonne classification de documents représentant un site portail, c'est-à-dire : chaque sous-arbre A représente une catégorie composée de toutes les fourmis de A. Soit f_i la fourmi qui est à la racine d'un sous-arbre A. Nous souhaitons que 1) f_i soit représentative de cette catégorie (les fourmis placées dans A sont les plus similaires possible à f_i), 2) les fourmis filles de f_i qui représentent des sous-catégories soient les plus dissimilaires possible entre elles. Autrement dit, un bon site portail est constitué de catégories homogènes, et pour une catégorie donnée, les sous-catégories sont judicieusement choisies (les plus dissimilaires possible entre elles).

4 Application du modèle de fourmis à la génération de sites portails

4.1 Méthodologie

Nous avons testé notre algorithme sur un ensemble de 6 bases allant de 258 à 4504 textes (voir le tableau 1). La base *CE.R.I.E.S.* contient 258 textes sur la peau humaine saine Guinot et al. (2003). La base *AntSearch* contient des documents sur des sujets scientifiques (73 sur des problèmes d'ordonnancement et de conduite de projets, 84 sur le traitement d'images et la reconnaissances des formes, 81 sur les réseaux et le protocole Tcp-Ip et 94 sur la 3D et les cours de VRML). Les bases *WebAce1* et *WebAce2* contiennent des pages web extraites des catégories de Yahoo !. Elles ont été construites par les auteurs de Han et al. (1998). Enfin, la base

WebKb comporte 4504 documents représentant des pages web de personnes appartenant à plusieurs universités américaines, classées en 6 catégories ("student", "staff", "project", "faculty", "departement", "course").

La classification obtenue est évaluée à la fois en terme de nombre de classes trouvées C_T, de pureté des classes P_R et d'erreur couple E_C (indice de Rand). Pour une classe donnée la pureté représente le pourcentage de pages bien classées ; Ec représente une mesure d'erreur de classification fondée sur les couples de documents de la base. Nous utilisons la mesure de similarité *cosinus*, où chaque document est représenté par un vecteur de poids calculé suivant le schéma *tf-idf* Salton (1971). Pour toutes les bases, nous comparons les résultats obtenus par notre algorithme avec ceux obtenus par la classification ascendante hiérarchique.

Bases	Taille (# de documents)	Taille (Mb)	# de classes
CE.R.I.E.S.	258	3.65	17
AntSearch	332	13.2	4
WebAce1	185	3.89	10
WebAce2	2340	19	6
WebKb	4504	28.9	6

TAB. 1 – *Bases de tests utilisées*

4.2 Résultats

4.2.1 Test de paramètres

AntTree$_{Sans-Seuil}$ ne possède aucun paramètre. Nous testons alors uniquement les différentes stratégies du tri initial des données (tri décroissant, tri croissant et tri aléatoire) en mesurant pour chaque donnée sa similarité moyenne avec les autres données. Les résultats sont présentés dans le tableaux 2.

Au vu des résultats, le tri décroissant est meilleur sur l'erreur en moyenne sur toutes les bases testées. En effet avec un tri croissant les deux premières fourmis à se connecter au support sont celles qui sont le moins similaires à toutes les autres. Une nouvelle fourmi aura donc du mal à se connecter au support, étant donné qu'elle sera rarement suffisamment dissimilaire ($Sim(f_i, f_+) < T_{Dissim}(f_{pos})$: ligne (5) de l'algorithme 1) pour créer une nouvelle classe. Il en résulte qu'avec ce type de tri l'algorithme AntTree$_{Sans-Seuil}$ généra peu de classes à la fin de son exécution.

Par contre avec un tri décroissant les premières fourmis à se connecter au support sont celles qui sont les plus similaires à toutes les autres. Ceci permet aux autres fourmis de se connecter assez facilement. Le tri aléatoire représente un compromis entre un tri croissant et un tri décroissant, qui sont les cas limites de l'algorithme.

4.2.2 Étude comparative

Pour la conception de la CAH (Classification ascendante hiérarchique) Lance et Williams (1967) nous utilisons le critère de *Ward* comme distance d'agrégation. Quant à la coupure du dendogramme elle consiste à couper l'arbre au niveau où la variation de la distance minimum

Tri	Croissant	Décroissant	Aléatoire
E_C moyenne [σ_{Ec}]	0,34	0,30	0,41 [0,07]

TAB. 2 – *Résultats obtenus par AntTree$_{Sans-Seuil}$. E_C représente l'erreur moyenne et l'écart type correspondant sur les 6 bases de test.*

Base de données	C_R	AntTree$_{Sans-Seuil}$ E_C	C_T	P_R	CAH E_C	C_T	P_R
CE.R.I.E.S.	17,00	0,31	6,00	0,23	0,36	3,00	0,29
AntSearch	4,00	0,24	6,00	0,72	0,17	6,00	0,79
WebAce1	10,00	0,29	6,00	0,32	0,28	4,00	0,27
WebAce2	6,00	0,32	7,00	0,77	0,29	3,00	0,79
WebKb	6,00	0,32	9,00	0,40	0,42	3,00	0,39

TAB. 3 – *Résultats obtenus par AntTree$_{Sans-Seuil}$ et CAH sur les bases de test avec un tri décroissant.*

entre les classes est maximale, c'est à dire là où les deux classes que l'on tente de regrouper sont les plus éloignées l'une de l'autre.

Le tableau 3 présente les résultats obtenus par notre algorithme de fourmis et ceux de la CAH (classification ascendante hiérarchique). En examinant le critère de l'erreur moyenne, on constate que les deux algorithmes AntTree$_{Sans-Seuil}$ et CAH sont équivalents et restent relativement proches. En étudiant de plus près ces deux derniers on constate que CAH obtient en moyenne une meilleure pureté et AntTree$_{Sans-Seuil}$ un nombre de classes plus exact.

5 Visualisation de résultats

5.1 Affichage texte

Notre système prend en entrée un ensemble de documents (issus du web ou non) afin de les organiser de manière automatique au sein d'un site portail. Cet ensemble de documents est classé de manière à produire un arbre caractérisant un site portail représenté dans un format XML. A partir de cette représentation XML, nous avons conçu deux types de visualisation du portail 1) par un site PHP généré dynamiquement décrit dans le paragraphe suivant et 2) par un système de visualisation 3D immersif plus complexe décrit dans la section 5.2.

On peut générer de manière automatique le site portail une fois les pages classées en arbre. La hiérarchie de documents ainsi construite est stockée dans une base de données. Les pages HTML du site sont générées de manière dynamique en PHP. La figure 3 représente l'interface du portail obtenu sur la base *AntSearch* (332 documents) où chaque document est représenté par son titre. On peut accéder aux documents via l'arborescence générée et on peut visualiser leur contenu sur l'interface du site portail. Pour pouvoir effectuer des recherches au sein de l'arborescence, nous avons intégré un outil de recherche utilisant un index inversé généré automatiquement dans la base de données.

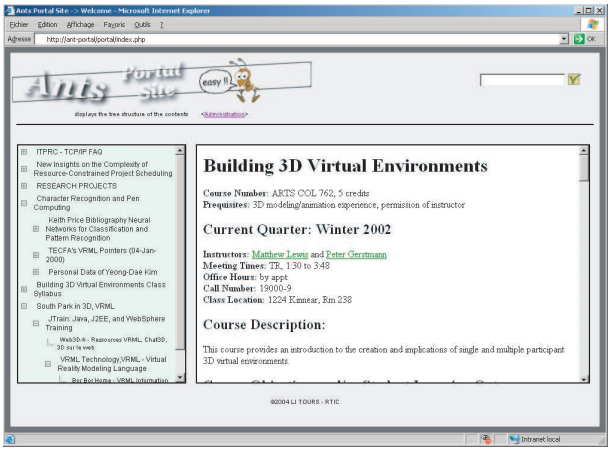


FIG. 3 – *Interface du site portail généré. A gauche, la représentation arborescente des documents. A droite, un document ouvert. En haut, le moteur de recherche à base d'index inversé.*

5.2 Affichage 3D

5.2.1 Système immersif

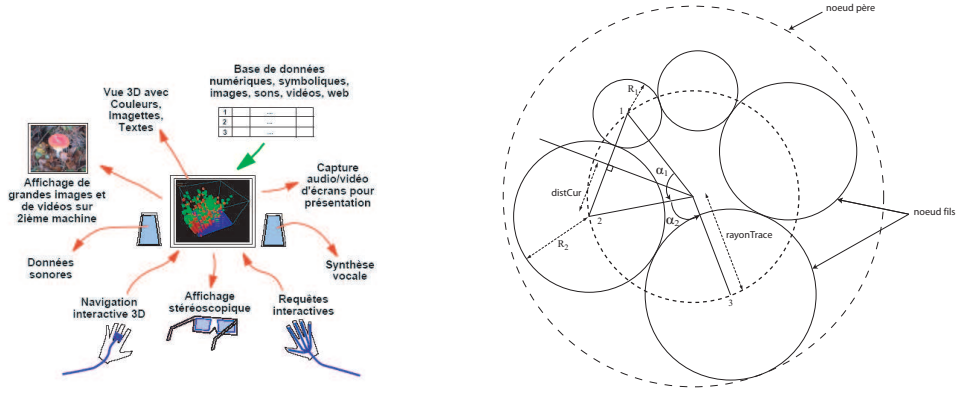
L'affichage de résultats sous forme de texte n'offre pas toujours les opportunités au concepteur pour permettre à l'utilisateur une lecture aisée de la quantité importante d'informations mise à sa disposition. La représentation de données de systèmes d'information sous forme graphique est plus avantageuse car elle fait passer beaucoup d'informations très rapidement à l'utilisateur.

Nous présentons dans cette section une application qui permet de visualiser les résultats obtenus par notre algorithme dans un environnement immersif en trois dimensions. Pour ce faire, l'arbre généré par notre algorithme est représenté par le schéma XML qui sera analysé afin de produire un affichage en 3D le plus compréhensible possible pour un utilisateur (calcul des coordonnées de chaque nœud de l'arbre). Cet affichage peut ensuite être visualisé à l'aide de VRminer Azzag et al. (2005b) (outil immersif de représentation de données) développé dans notre équipe dans le cadre d'une collaboration avec le CE.R.I.E.S (CEntre de Recherches et d'Investigations Épidermiques et Sensorielles de CHANEL).

VRminer est une nouvelle méthode interactive de visualisation 3D de données multimédia (numériques, symboliques, sons, images, vidéos, sites Web) en réalité virtuelle (voir figure 4 (a)). Elle utilise un affichage 3D stéréoscopique permettant ainsi de représenter les données. A cet affichage est ajouté l'apparition de textes contextuels, l'utilisation de la synthèse vocale, la lecture de sons, l'affichage d'imagettes ainsi que l'affichage de grandes images, de vidéos ou de sites web sur une deuxième machine. La navigation au sein des visualisations est effectuée grâce à l'utilisation d'un capteur 3D à six degrés de liberté qui simule une caméra virtuelle. Des requêtes interactives peuvent être posées à la machine par l'utilisation d'un gant de données reconnaissant les gestes.

Tout d'abord, afin d'utiliser VRminer nous avons eu besoin de définir un algorithme d'affichage d'arbre en 3D, les coordonnées de chaque nœud étant ensuite transmises à VRminer

sous un format spécifique interprétable par l'outil. Cet algorithme génère un arbre utilisant le même principe que celui du *Cone tree* à angles variables Robertson et al. (1991).



(a) (b)

FIG. 4 – *Vue globale du système de visualisation interactive (exemple sur des données numériques)(a). Principe de l'algorithme d'affichage d'arbre en 3D (b)*

5.2.2 L'algorithme d'affichage d'arbre en 3D

Afin de pouvoir placer les nœuds dans l'espace 3D, il faut donc déterminer leur position par rapport à leur nœud père. Ainsi on doit résoudre le problème récursif suivant : en partant des nœuds feuilles de l'arbre puis en remontant progressivement jusqu'au nœud racine, déterminer, pour chaque nœud, le rayon du cercle sur lequel seront positionnés les nœuds fils sans chevauchement. Le rayon à calculer à chaque niveau est représenté dans la figure 4(b) par le segment *rayonTrace*.

Si le nœud considéré ne contient aucun sous-arbre, dans ce cas la valeur du rayon R_i de l'objet correspond à la taille nécessaire à la représentation de l'objet symbolisant le nœud. S'il contient uniquement un nœud fils, ce dernier aura les mêmes paramètres que son nœud parent, excepté le niveau de la hauteur qui, lui, sera plus élevé. Dans le cas général, on cherche à calculer une valeur pour le rayon de traçage *rayonTrace* répondant aux critères suivants :

- Le centre du cercle représentatif de chaque sous-arbre doit être inscrit sur un cercle de rayon *rayonTrace*,
- Les cercles représentatifs des sous-arbres positionnés côte à côte doivent être tangents,
- Il ne peut y avoir de recouvrement entre les cercles représentatifs des sous arbres.

Pour satisfaire au conditions précédentes, nous estimons la valeur de *rayonTrace* avec une certaine précision en lançant une recherche dichotomique. L'intervalle de recherche sera initialisé avec les bornes suivantes :

- borne min : on initialisera la valeur minimale de *rayonTrace* à $\frac{R_1+R_2}{2}$ avec R_1 et R_2 représentant les rayons respectifs des deux plus grands sous-arbres du nœud considéré ($R_i = (Max(R_j)\, des\, noeuds\, Fils_j) + rayonTrace$).

– borne max : en prenant *rayonTrace* égale à $Max(R_i) \times NbFils$, on est certain de ne pas avoir de chevauchement.

Une fois que nous obtenons une valeur pour le rayon de traçage *rayonTrace*. Nous cherchons à la valider en examinant la disposition de chacun des éléments en mesurant l'angle compris entre les centres de deux cercles consécutifs (représentant les sous-arbres) et le centre du tracé. Dans le cas d'une disposition idéale, la somme de tous ces angles est égale à 2π. Une valeur supérieure à 2π indiquera un chevauchement tandis qu'une valeur inférieure correspondra à une solution faisable mais non optimale. L'angle est par conséquent calculé comme suit :

$$\alpha_i = 2 \times ArcSin\left(\frac{disCur(i)}{rayonTrace}\right) \tag{2}$$

$$disCur(i) = \frac{R_i + R_{i+1}}{2} \tag{3}$$

A partir de là, tous les paramètres de l'arbre sont connus pour obtenir les coordonnées 3D. Nous présentons dans la section suivante des exemples de résultats graphiques.

5.2.3 Résultats

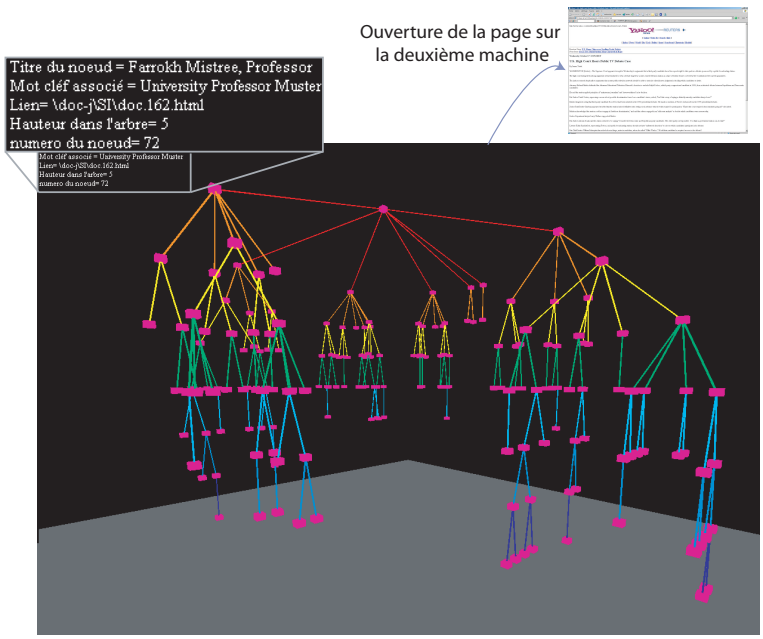


FIG. 5 – *Exemple d'une vue 3D de l'arbre généré sur la base WebAce1 (185 nœuds), avec un zoom sur la partie où sont affichées les caractéristiques du nœud correspondant.*

La figure 5 donne une vue d'ensemble du système présenté. L'arbre est affiché par des vues 3D sur un écran principal, les nœuds peuvent être des sphères ou des cubes selon le choix de

l'utilisateur et à chaque niveau différent est affecté une couleur. Des lunettes LCD permettent à l'utilisateur de percevoir ces données en stéréoscopie, et la navigation au sein des données a lieu grâce à un capteur 3D positionné par exemple sur la main gauche. L'utilisateur donne des commandes et des requêtes interactives via un gant de données placé par exemple sur la main droite (des raccourcis clavier existent aussi pour la navigation 3D). L'utilisateur peut sélectionner des données et obtenir ainsi dynamiquement des informations contextuelles : titre, résumé, mots clés (voir figure 5). Pour générer ces mots clés, nous avons pris pour chaque catégorie les mots les plus significatifs (représentant les poids *tf***idf* les plus élevés) apparaissant dans les documents de la catégorie concernée.

6 Conclusion

Dans cet article, nous avons développé des travaux abordant un certain nombre de thématiques comme un nouveau modèle d'auto-assemblage de fourmis artificielles, la classification hiérarchique de données textuelles, le Web et la construction automatique de sites portails, la visualisation des résultats. Nous avons ainsi introduit une nouvelle méthode construisant un arbre qui se caractérise par le fait qu'elle ne nécessite pas d'information a priori (nombre de classes, partition initiale), et même aucun paramètre.

Plusieurs perspectives peuvent être dégagées. La première consiste à généraliser le type de structures construites. Ainsi, il est simple et naturel d'étendre la construction d'arbres à la construction de graphes. Chaque fourmi serait un nœud du graphe en cours d'assemblage. Étant donné que de nombreux problèmes d'optimisation consistent à obtenir une solution sous forme de graphes, cela permettrait d'étendre le domaine d'application de ces algorithmes d'auto-assemblage.

Nous pourrions aussi appliquer ce principe de construction d'arbres à la découverte d'ontologies dans le web sémantique. Les fourmis pourraient ainsi s'auto-assembler pour définir un arbre de termes.

Références

Anderson, C., G. Theraulaz, et J. Deneubourg (2002). Self-assemblages in insect societies. *Insectes Sociaux 49*, 99–110.

Azzag, H., C. Guinot, A. Oliver, et G. Venturini (2005a). A hierarchical ant based clustering algorithm and its use in three real-world applications. In K. S. Wout Dullaert, Marc Sevaux et J. Springael (Eds.), *European Journal of Operational Research (EJOR)*. Special Issue on Applications of Metaheuristics.

Azzag, H., F. Picarougne, C. Guinot, et G. Venturini (2005b). Vrminer : a tool for multimedia databases mining with virtual reality. In J. Darmont et O. Boussaid (Eds.), *Processing and Managing Complex Data for Decision Support*. to appear.

Guinot, C., D. J.-M. Malvy, F. Morizot, M. Tenenhaus, J. Latreille, S. Lopez, E. Tschachler, et L. Dubertret (2003). Classification of healthy human facial skin. Textbook of Cosmetic Dermatology Third edition (to appear).

Han, E.-H., D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, et J. Moore (1998). Webace : a web agent for document categorization and exploration. In *AGENTS '98 : Proceedings of the second international conference on Autonomous agents*, New York, NY, USA, pp. 408–415. ACM Press.

Kumar, R., P. Raghavan, S. Rajagopalan, et A. Tomkins (2001). On semi-automated web taxonomy construction. In *WebDB*, pp. 91–96.

Lance, G. et W. Williams (1967). A general theory of classificatory sorting strategies : I. hierarchical systems. *Computer journal 9*(4), 373–380.

Lebart, L. et A. Salem (1994). *Statistique textuelle*. Paris : Dunod.

Lioni, A. (2000). *Auto-assemblage et transport collectif chez oecophylla*. Thèse de doctorat, Université libre de bruxelles, Université Paul Sabatier.

Robertson, G. G., J. D. Mackinlay, et S. K. Card (1991). Cone trees : animated 3d visualizations of hierarchical information. In *CHI '91 : Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, pp. 189–194. ACM Press.

Salton, G. (1971). The smart retrieval system. experiment in automatic document processing. *Prentice Hall*.

Sauwens, C. (2000). *Étude de la dynamique d'auto-assemblage chez plusieurs espèces de fourmis*. Thèse de doctorat, Université libre de bruxelles.

Yahoo ! http ://www.yahoo.com.

Summary

The initial goal of this work was to build a new hierarchical clustering algorithm and to apply it to several domains including the automatic construction of portal sites for the Web. A portal site can be viewed as a hierarchical partitioning of a set of documents. We propose in this paper a new approach which builds a tree-structured partitioning of the data. This method simulates a new biological model: the way ants build structures by assembling their bodies together. Ants start from a point and progressively become connected to this point, and recursively to these firstly connected ants. Ants move in a distributed way over the living structure in order to find the best place where to connect. This behavior can be adapted to build a tree from the data to be clustered.

Cartes de pages Web obtenues par un automate cellulaire

H. Azzag[*], D. Ratsimba[***], A. Alarabi[*], C. Guinot[,*], G. Venturini[*]**

[*]*Université François-Rabelais de Tours, Laboratoire d'Informatique (EA 2101),*
64, Avenue Jean Portalis, 37200 Tours, France
{hanene.azzag, venturini}@univ-tours.fr; ali.alarabi@etu.univ-tours.fr

[**]*CE.R.I.E.S., 20, rue Victor Noir, 92521 Neuilly-sur-Seine Cédex, France*
christiane.guinot@ceries-lab.com

[**]*Laboratoire ERIC, Université de Lyon2, Bat. L, 5 avenue Pierre Mendès-France 69676 Bron Cédex*
dratsimb@club-internet.fr

RÉSUMÉ. Nous présentons dans cet article un algorithme de classification visuelle utilisant un automate cellulaire permettant d'afficher une carte de pages Web. Nous commençons par faire des rappels sur les méthodes construisant des cartes de pages Web, sur la classification de données et enfin sur les concepts liés aux automates cellulaires. Nous montrons ensuite comment ces concepts peuvent être appliqués à la classification de pages Web : les cellules réparties sur une grille 2D peuvent soit être vides soit contenir une page. La fonction locale de transition des cellules favorise la constitution de regroupement d'états (pages) similaires pour des cellules voisines. Nous présentons ensuite les résultats visuels de notre méthode sur des données classiques ainsi que sur des documents que l'on souhaite organiser visuellement afin de permettre à un utilisateur de naviguer facilement à travers cet ensemble.

MOTS-CLÉS : Cartes de documents, Automates cellulaires, Classification non supervisée, Méthodes biomimétiques

1 Introduction

La visualisation d'un ensemble de pages Web a pour objectif de présenter d'une manière la plus informative possible des documents à un utilisateur afin de lui permettre de naviguer dans cet ensemble et d'accomplir une tâche donnée, bien souvent la recherche d'informations (voir un survol dans (Mokaddem et al. 2006)). Cela peut correspondre à plusieurs problématiques liées au Web : l'affichage des résultats d'un moteur de recherche, mais aussi la visualisation d'un graphe de pages comme un hypertexte ou comme la navigation d'un utilisateur.

En plus de cette visualisation, le regroupement thématique des pages apporte des informations importantes à l'utilisateur et lui permet d'explorer ces pages en s'appuyant sur des similarités automatiquement calculées (Zamir et Etzioni 1999). Ces similarités doivent tenir compte du contenu textuel des pages. Un certain nombre de systèmes ont donc pour double objectif de construire des groupes de pages et de visualiser ces groupes. Si l'on considère le modèle visuel représenté par les cartes (« topic maps ») (Kohonen 1988) (Wise 1999) par opposition aux systèmes utilisant des graphes (Kartoo, Mapstan, TouchGraph GoogleBrowser), des nuages de points et autres représentations (Cugini 2000), on peut dire que les avantages des cartes sont de pouvoir représenter un grand ensemble de documents et les

regroupements que l'on peut y effectuer, de donner une vue globale de cet ensemble qu'il est possible de zoomer, et surtout d'utiliser une représentation cartographique familière à l'utilisateur et nécessitant donc peu d'apprentissage.

Nous allons donc considérer dans ce travail un ensemble de n pages Web que nous souhaitons représenter sous la forme d'une carte servant de base à la navigation de l'utilisateur. De ce point de vue, le problème que nous cherchons à résoudre est proche de la construction automatique d'hypertextes (un ensemble de pages que l'on souhaite publier sur le Web) ou de la présentation visuelle des résultats d'un moteur de recherche, ou encore de la construction automatique du plan d'un site Web. Cette carte doit représenter toutes les pages en faisant apparaître des regroupements thématiques. Pour établir une telle carte, il faut être capable de mesurer la similarité entre les pages, de détecter des groupes au sein des pages, et de visualiser ces groupes en faisant apparaître les relations de voisinages entre pages (visualiser des pages similaires à proximité les unes des autres). En atteignant cet objectif, l'utilisateur va pouvoir dépasser les limites des interfaces textuelles : dans le cas des moteurs de recherche, un plus grand ensemble de résultats va pouvoir être exploré, dans le cas de la construction d'un hypertexte, les relations de voisinage trouvées vont jouer naturellement le rôle d'hyperliens entre les pages.

La section 2 décrit les principes des méthodes classifiant visuellement des pages Web, ainsi que les propriétés des automates cellulaires. La section 3 présente notre algorithme et les différents choix que nous avons du effectuer. La section 4 présente des résultats expérimentaux sur des jeux de données classiques et sur des pages Web. Nous concluons ensuite en présentant notamment les limites et les perspectives liées à ce travail.

2 Principes des cartes de documents et des automates cellulaires

2.1 Principes de la classification visuelle de documents sous forme de cartes

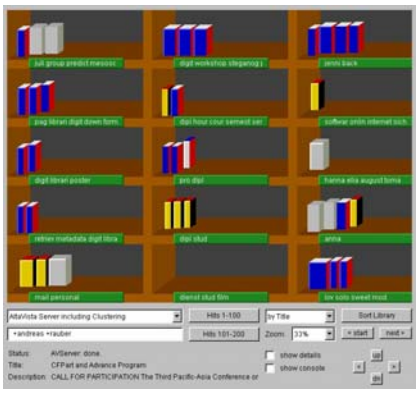


Figure 1 Exemple d'une carte de documents utilisant une métaphore de bibliothèque : LibViewer du système SOMLib (Rauber et Bina 1999) (image fournie gracieusement par A. Rauber)

Pour résoudre le problème posé nous devons nous intéresser aux méthodes de classification de documents donnant un résultat visuel sous la forme d'une carte (ou alors aux méthodes de classification dont les résultats seront visualisables à l'aide d'un autre outil) (Kohonen 1988) (Wise 1999). Les cartes de Kohonen sont les exemples les plus connus (Roussinov 1999) (Chen et al., 1996) (Rauber et Bina 1999, voir figure 1). D'autres méthodes moins connues ont été plus récemment appliquées à la construction d'une carte de documents : il s'agit d'une approche biomimétique utilisant les fourmis artificielles (Handl et Meyer 2002) dans laquelle des fourmis placent des documents sur une grille 2D.

Les caractéristiques de ces représentations sont les suivantes : la carte est construite à partir de l'ensemble de documents en appliquant un algorithme de classification dont le résultat est directement interprétable visuellement. Ce résultat est organisé dans un plan 2D. Des documents proches thématiquement se retrouvent également à proximité sur la carte. Des annotations sont ajoutées de manière à aider l'utilisateur à comprendre les regroupements effectués sur la carte. Ces annotations peuvent être les titres des documents, des mots-clés extraits des documents, mais également des couleurs indiquant la densité des documents dans une région donnée. On peut donc donner les caractéristiques que doivent vérifier ces cartes : faire apparaître des groupes (avec des relations de voisinage entre les documents du type « documents proches sur la carte ⇔ documents proches thématiquement »), faire apparaître la taille des groupes (effectifs), donner des détails sur la thématique d'un groupe (affichage de mots clés en commun), donner des détails sur un ou plusieurs documents (titre du document, possibilité de l'ouvrir).

2.2 Principes des automates cellulaires pour la classification visuelle de données

Parmi toutes les méthodes et problématiques liées au domaine de la classification (Jain et al. 1999), des chercheurs s'intéressent plus spécialement aux méthodes inspirées de systèmes ou de phénomènes biologiques (voir survol dans (Azzag et al. 2004)). A notre connaissance, aucun algorithme de classification utilisant les automates cellulaires n'a été défini à ce jour. Pourtant le modèle des automates cellulaires est connu depuis longtemps (Von Neumann 1966) et possède de nombreuses propriétés intéressantes comme celles que l'on retrouve notamment dans le célèbre "jeu de la vie" (Gardner 1970) : l'émergence de comportements complexes à partir de règles locales plus simples. Nous allons donc montrer dans la suite de cet article que ce modèle, utilisé dans de nombreux domaines (Ganguly et al. 2003), peut apporter sa contribution au problème de la classification visuelle de documents et dans la construction d'une carte de pages Web.

Nous rappelons quelques principes des automates cellulaires (AC par la suite) que nous avons utilisé pour définir notre algorithme. Un AC est défini par la donnée d'un quadruplet (C, S, V, δ). $C=\{c_1, ..., c_{NCell}\}$ représente un ensemble de cellules où $NCell$ est constant au cours du temps. $S=\{s_1, ...,s_k\}$ est l'ensemble fini d'états que va pouvoir prendre chaque cellule. L'état de la cellule c_i est noté $c_i(t)$. V représente le voisinage entre cellules qui va structurer l'ensemble des cellules. Pour chaque cellule c_i, on définit $V(c_i)$ comme l'ensemble des cellules voisines de c_i. Nous allons nous intéresser dans ce travail à une structuration en 2D des cellules qui sont placées sur une matrice ou grille de dimension $N \times N$ (le nombre de cellules vaut donc $NCell = N^2$). Chaque cellule possède un voisinage carré de coté v centré sur elle-même. Ce voisinage est tel que la grille est toroïdale (le haut est relié au bas, le coté droit au coté gauche). Une cellule a donc toujours un voisinage de v^2 cellules. La fonction de transition locale δ détermine le nouvel état d'une cellule en fonction des états perçus. Enfin, on appelle configuration de l'AC à l'instant t le vecteur d'états $AC(t) = (c_1(t), ..., c_{NCell}(t))$. Un AC évolue de $AC(t)$ à $AC(t + 1)$ en appliquant δ à chacune des cellules, soit de manière synchrone (mode parallèle), soit de manière asynchrone (mode parallèle ou séquentiel).

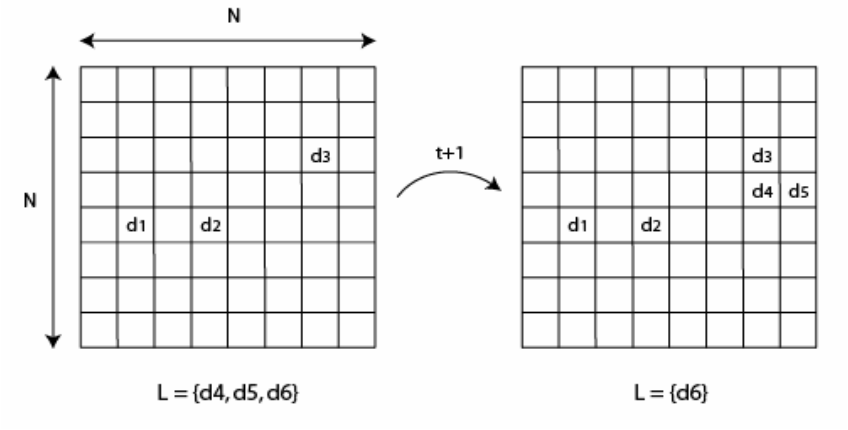


Figure 2 – Représentation de notre automate cellulaire 2D et de la liste *L* d'états

Le principe de notre méthode est de considérer que les états de notre AC vont pouvoir contenir des documents. Ensuite, nous allons adapter la fonction locale de transition de manière à ce que des états se forment en groupe lorsqu'ils représentent des documents similaires. Ainsi, une fois établie une configuration stable dans laquelle des documents thématiquement proches se retrouvent dans des cases voisines sur la grille, il est immédiat d'obtenir une carte à partir de cette grille, et nous verrons dans la section 4.3 que cette forme de résultat vérifie les propriétés recherchées mentionnées dans la section précédente.

3 Modélisation pour la classification visuelle de documents et description de l'algorithme

Nous notons dans la suite par $d_1, \ldots, d_n$ les n documents/données à classer et par $Sim(i, j) \in [0, 1]$ la mesure de similarité entre deux documents d_i et d_j. Nous avons considéré un automate 2D où les *NCell* cellules sont réparties sur une grille carrée (voir figure 2).

L'ensemble des états des cellules est $S = \{vide, d_1, \ldots, d_n\}$. Autrement dit, chaque cellule sera vide ou bien contiendra une (et une seule) donnée. À chaque itération de l'algorithme, les états de toutes les cellules vont être (éventuellement) modifiés selon des règles locales qui vont tendre à faire apparaître des états (documents) similaires pour des cellules voisines sur la grille.

La taille de la grille est fixée empiriquement (Lumer et Faeita 1994) en fonction de n avec la formule $N = E(\sqrt{3n}) + 1$ afin de laisser de la place (N^2 cellules au lieu de n) pour l'organisation spatiale des classes. La taille v du carré définissant le voisinage a été fixée également empiriquement à $v = E(N/10.) + 1$.

Nous allons utiliser les notations/définitions suivantes : une cellule est isolée si son voisinage immédiat comporte moins de 3 cellules non vides. Nous avons choisi d'obtenir des classifications non recouvrantes : un état d_i donné ne pourra apparaître qu'une seule fois dans la grille. Nous utilisons donc une liste L qui représente la liste des documents qui n'apparaissent pas sur la grille (et qui restent à placer). Initialement, L contient tous les documents et les états de toutes les cellules sont à "*vide*".

Les règles locales de changement d'état sont les suivantes, pour une cellule C_{ij} vide :

- R_1 : Si C_{ij} est isolée, Alors $C_{ij}(t+1) \leftarrow d_k$ où d_k est un document choisi aléatoirement dans L

- R_2 : Si C_{ij} est non isolée, Alors $C_{ij}\ (t+1) \leftarrow d_k$ où d_k est soit un document choisi aléatoirement dans L (probabilité $P = 0.032$), soit le document de L le plus similaire à ceux du voisinage de C_{ij} (probabilité $1 - P = 0.968$)

Pour une cellule C_{ij} contenant un document d_k (i.e. $C_{ij}\ (t) = d_k$) :

- R_3 : Si $\overline{Sim}_{d_{k'} \in V(C_{ij})}(d_k, d_{k'}) < Seuil(t)$, alors $C_{ij}(t + 1) \leftarrow vide$ et d_k est remise dans L
- R_4 : Si C_{ij} est isolée Alors $C_{ij}(t+1) \leftarrow vide$ avec une probabilité $P' = 0,75$ (d_k est remise dans L).

Dans les autres cas, la cellule reste inchangée ($C_{ij}(t + 1) \leftarrow C_{ij}(t)$).

Pour appliquer ces règles sur les cellules et éviter les conflits d'affectation des données présentes dans L, nous avons testé plusieurs ordres de parcours de la grille afin de décider quelles cellules accèdent à la liste en premier. L'ordre que nous avons sélectionné est de parcourir aléatoirement les cellules (une permutation des N^2 cellules est générée aléatoirement au début de l'algorithme).

La valeur de *Seuil(t)* est initialisée à la similarité maximum entre les données, puis va décroître progressivement. Initialement, les documents placés côte à côte seront donc très similaires. À chaque itération de l'algorithme, ce seuil est décrémenté d'un pas constant (égal à un 200ième de l'écart type observé dans les similarités). La diminution de ce seuil fait que l'algorithme va converger puisque les documents une fois mis en place ne bougeront plus lorsque *Seuil (t)* sera faible.

Bases	# de données	# de classes réelles
Art1	400	4
Art2	1000	2
Art3	1100	4
Art4	200	2
Art5	900	9
Art6	400	4
Ceries	259	6
Glass	214	7
Iris	150	4
Pima	768	2
Soybean	47	4
Thyroïd	215	3
Wine	165	3

Tableau 1 – Bases de données classiques testées (Machine Learning Repository)

4 Résultats

4.1 Données numériques et symboliques classiques

Afin de valider les capacités de classification de notre algorithme, nous l'avons tout d'abord appliqué sur des bases de données classiques issues du *Machine Learning Repository* (Blake et Merz 1998) (voir tableau 1). La mesure de similarité que nous utilisons est calculée grâce à une distance Euclidienne (données numériques) ou de Hamming (données symboliques).

Nous avons utilisé le même jeu de paramètre pour toutes les bases (voir section précédente) qui a été déterminé à l'issue de nombreux tests statistiques. Nos premières expériences ont donc été orientées dans le but de tester tous les paramètres de l'algorithme afin de trouver un ou plusieurs jeux de paramètres satisfaisants. Nous avons lancé une série de tests sur les 13 bases de données du tableau 1. Chaque jeu de test a été testé 20 fois sur chaque base de données pour en tirer des statistiques correctes. Au total, environ 1500 jeux de paramètres différents auront été testés sur les treize bases en deux sessions de tests. Au final, nous avons pu mettre en évidence la corrélation de certains paramètres et trouver des valeurs permettant de construire un classifieur ayant des performances intéressantes en terme de taux d'erreur de classification (indice de Rand), taux de pureté des classes formées, mais aussi par rapport au taux d'erreur du nombre de classes trouvées par rapport au nombre de classes théoriques.

Les résultats visuels présentés sur la figure 2 illustrent les classifications trouvées pour certaines de ces données (voir tableau 2). Nous remarquons que la disposition des classes correspond aux propriétés connues des bases, comme par exemple pour les bases Iris et Wine. Les temps d'exécutions ont été relevés sur un PC AMD Athlon64 792MHz avec 500Mo de RAM pour une implémentation en Applet Java (exécution nettement plus lente qu'en langage C par exemple).

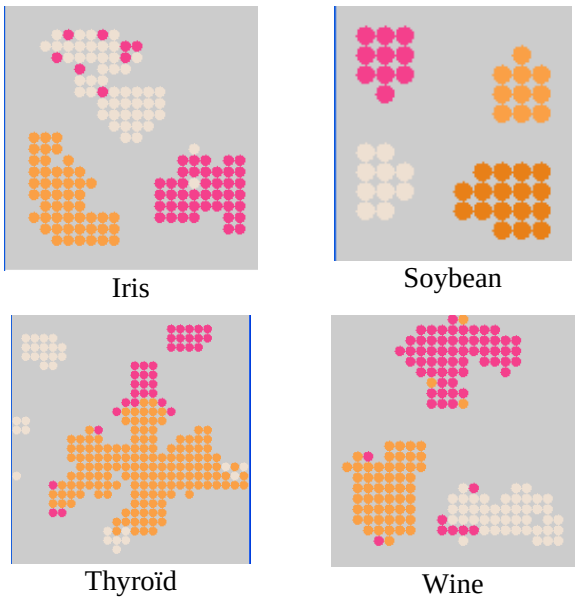


Figure 3 – Résultats visuels obtenus (les couleurs indiquent les classes réelles)

	Automate cellulaire				Classification ascendante hiérarchique			
Bases	# de classes trouvées	Pureté	Erreur	Temps moyen	# de classes trouvées	Pureté	Erreur	Temps moyen
Art1	6.25	87%	15%	4.3s	5	84%	15%	0.92s
Art2	6.35	97%	20%	28s	3	98%	14%	23.73s
Art3	7.70	92%	20%	59s	3	89%	16%	32.34s
Art4	6.25	100%	29%	1.2s	3	100%	13%	0.11s
Art5	8.65	63%	14%	7.5s	10	78%	8%	14.25s
Art6	4.75	100%	2%	8.0s	5	100%	3%	0.91s
CE.R.I.E.S.	5.80	75%	18%	3.5s	3	56%	24%	0.25s
Glass	5.50	56%	36%	3.7s	3	49%	43%	0.14s
Iris	4.75	95%	15%	0.8s	3	88%	13%	0.05s
Pima	5.50	67%	47%	70s	3	48%	48%	9.52s
Soybean	3.95	96%	2 %	0.4s	6	100%	8%	0.00s
Thyroïd	4.80	85%	33%	2.2s	5	84%	35%	0.14s
Wine	4.70	90%	13%	2.6s	6	84%	20%	0.06s

Tableau 2 – Résultats sur des bases de données classiques

Une analyse des résultats en terme de pureté et de nombre de classes a été réalisée (voir tableau 2). Ces performances sont plutôt correcte par rapport à la classification ascendante hiérarchique qui est un algorithme de classification largement utilisé.

4.2 Données textuelles

Bases	# de classes réelles	# de documents	Volume (Mo)
WebAce 1	10	185	3,89
WebAce2	6	2340	19
AntSearch	4	319	13,2
CERIES	17	258	3,65

Tableau 3 – Bases de données textuelles testées

Nous avons appliqué notre algorithme en considérant cette fois que la base de données est un ensemble de documents. Nous avons donc calculé la matrice de similarité en utilisant des méthodes spécifiques aux données textuelles.

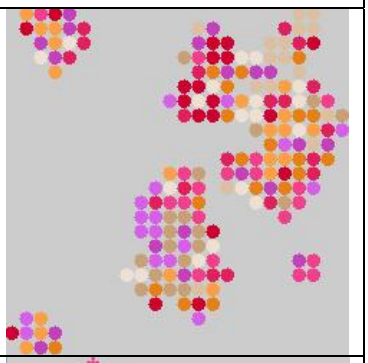
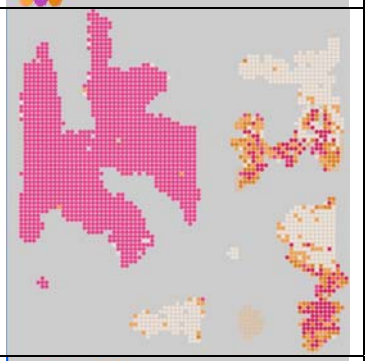
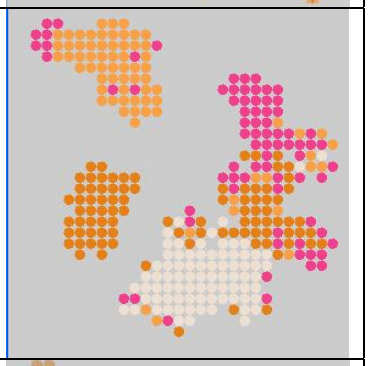
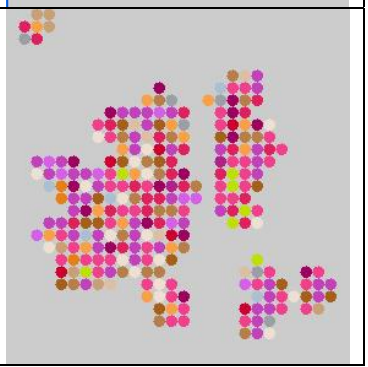
Bases	Classes trouvées	% erreur	Temps de calcul	Aperçu de la carte obtenue
Webace1	4	65%	4 s	
Webace2	7	10%	1800 s	
AntSearch	4	20%	22 s	
CERIES	4	47%	9 s	

Tableau 4 – Résultats sur les bases de données textuelles

Nous avons utilisé un ensemble de bases allant de 258 à 2340 textes (voir le tableau 3). La base *CE.R.I.E.S.* contient 258 textes sur la peau humaine saine (Guinot et al. 2003). La base *AntSearch* contient des documents sur des sujets scientifiques (73 sur des problèmes d'ordonnancement et de conduite de projets, 84 sur le traitement d'images et la reconnaissance des formes, 81 sur les réseaux et le protocole Tcp-Ip et 94 sur la 3D et les cours de VRML). Nous avons choisi d'extraire nous mêmes ces textes afin d'avoir une base réelle pour nos tests. Les bases *WebAce* contiennent des pages web extraites des catégories de Yahoo!. Elles ont été construites par les auteurs de (Han et al. 1998).

Nous utilisons la mesure *cosinus* pour calculer la similarité entre les documents à analyser. Chaque document est ainsi représenté par un vecteur de poids calculé suivant le schéma *tf-idf* [SAL 1975]. La classification obtenue est évaluée de la même manière que dans le cas des données numériques.

Nous présentons dans le tableau 4 les résultats obtenus en terme de classification. Nous savons par avance que les bases Webace1 et CERIES sont difficiles, pour de nombreux algorithmes de classification, car la similarité statistique calculée n'est pas assez précise. On peut donc dire que les mauvais résultats obtenus ne sont pas surprenants. Cependant, dans un deuxième, nous pensons vérifier la carte qui est générée avec ces données pour voir si les regroupements ont tout de même un sens (ils ont actuellement un sens vis-à-vis de la similarité calculée mais celle-ce ne reflète pas le classement des experts humains).

Pour les bases Webace2 et Antsearch, les cartes obtenues sont cette fois nettement meilleures et l'on peut clairement distinguer que les classes d'origine se retrouvent bien sur la grille.

4.3 Génération d'une carte

A partir de la grille précédente, nous avons généré une carte « navigable » de la manière suivante : les positions 2D des documents sont respectées et transformées en un tableau HTML. Chaque case du tableau contenant un document est annotée en utilisant le début du titre du document. Ensuite, à l'aide de commandes en JavaScript nous pouvons ajouter des interactions. Un clic sur une case ouvre le document correspondant. Un passage de la souris affiche le titre du document. Avec le navigateur utilisé (FireFox), il est possible de zoomer de manière très fluide avec la molette de la souris. On dispose donc d'une carte représentant les proximités thématiques entre les documents avec la possibilité d'avoir un aperçu du titre du document, ainsi que la possibilité d'ouvrir le document en entier. Il est possible d'évaluer visuellement la taille des groupes et de naviguer d'un document voisin à un autre « par le contenu ».

Nous présentons une carte complète pour la base Antsearch sur la figure 4 ainsi qu'un zoom sur la figure 5. La génération de mots-clés dans la carte est relativement simpliste (et rapide) mais elle fournit facilement une explication sur la classification thématique des documents. Les débuts de titres se complètent assez bien pour donner une idée du thème abordé dans telle région de la carte. Cependant, en observant les titres, on s'aperçoit que les documents à proximité les uns des autres ont des titres contenant beaucoup de mots-clés significatifs en commun. Il est certain qu'une première extension très simple de ce travail consiste à représenter sur la carte les mots en commun dans les titres des 9 documents centrés sur chaque case (la case elle-même et les 8 documents voisins sur la grille) plutôt que d'utiliser la méthode simpliste actuelle.

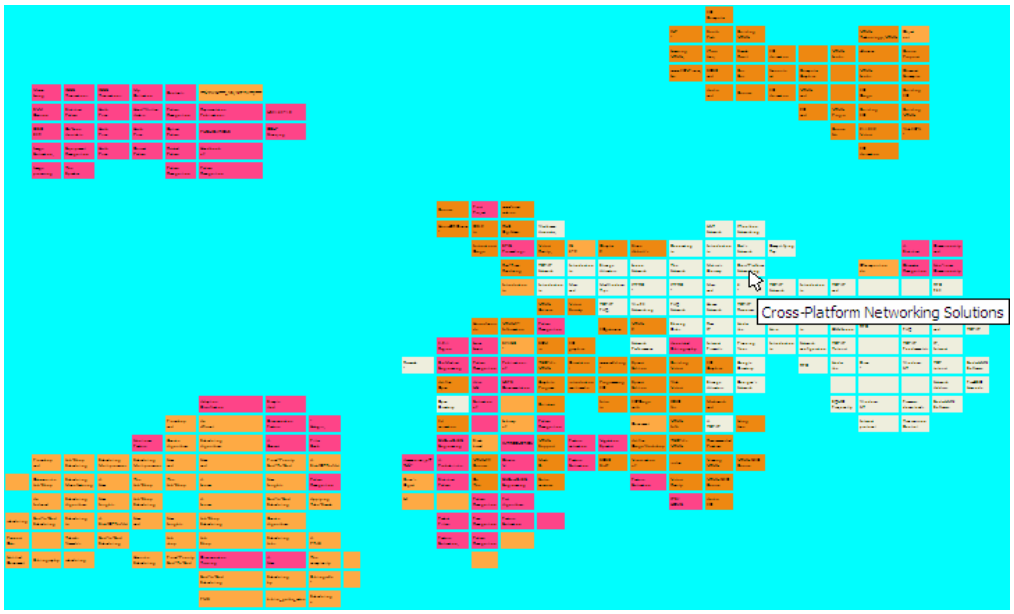


Figure 4 – Exemple d'une carte générée sur la base Antsearch (319 documents) avec annotations

		UNT Network	ITtoolbox Networking				
Steve Abbott's	Connecting to	Introduction to	Bart's Network	Demystifying Tcp			
Linux Network	The Network	Matisse's Glossary	Cross-Platform Networking				D?composi du
ITPRC -	ITPRC -	Macs and	6 -	TCP/IP Network	Introduction to	TCP/IP and	
Win95 Networking	FAQ: Network	Home Network	TCP/IP Resources		Internet Protocols	Sniffing (network	Sangoma -
VRML 2	Sharing Disks	Raw IP	Under the	Newbie: Unix	Introduction to	Debian GNU/Linux	RFC
Network Performance	Annotated Bibliography	Internet Firewalls:	Planning Your	Introduction to	Network configuration	TCP/IP Tutorial	
Special Edition	Building Virtual	3D Graphics	Google Directory		RFC	Under the	Cisco -
Special Edition	Web Virtual	Strange Attractors	Dunigan's Network				
3D-Design with	ICSD for	Meshwork and				SQUID Frequently	Windows NT
Document	VRML Info:	A TCP/IP	Using Java				Internet protocol
Artifice DesignWorkshop	TECFA's VRML	Recommended Practices					

Figure 4 – Zoom sur une partie de la carte Antsearch

5 Conclusion et perspectives

Nous avons proposé un algorithme de classification visuelle utilisant les automates cellulaires. Nous avons montré expérimentalement que cet algorithme est capable de regrouper de manière pertinente des données de bases classiques. Il est de plus capable de produire une visualisation des résultats et peut contribuer ainsi à la problématique de la génération de cartes de documents permettant une exploration intuitive.

Les limites de notre méthode portent actuellement sur l'extraction des mots clés annotant la carte, la perte du contexte dans de grandes cartes (plusieurs milliers de documents), des erreurs de classification qui subsistent localement. Pour l'extraction des mots clés, nous avons mentionné dans la section précédente une méthode qui consisterait à extraire les mots les plus fréquents dans les 9 documents centrés autour d'une case donnée. En ce qui concerne la perte du contexte, nous pensons pouvoir résoudre ce problème de plusieurs manières. D'une part l'utilisation d'une « mini-carte » permettrait à l'utilisateur de se repérer globalement dans la visualisation. Nous envisageons également d'utiliser un zoom « sémantique » en établissant plusieurs niveaux hiérarchiques dans la carte. A partir de la grille initiale, on peut facilement regrouper les cellules (par carré de 3x3 par exemple) et obtenir ainsi plusieurs niveaux dans la carte. D'un niveau à l'autre, la taille de la carte serait ainsi divisée par 9, ce qui assure un nombre de niveaux limité. Les annotations faites sur un niveau supérieur seraient obtenues par extraction des mots clés des regroupements de cellules effectués au niveau inférieur. Ce zoom sémantique permettrait donc de passer d'un niveau à l'autre et donnerait à l'utilisateur la possibilité de garder le contexte global de la carte. Finalement, en ce qui concerne la visualisation, nous pouvons également envisager la possibilité de représenter chaque document par des signes visuels plus informatifs qu'une simple case colorée : on pourrait par exemple utiliser des vues réduites des documents, ou bien des indices visuels donnant d'autre informations.

Du point de vue de la classification, nous pensons pouvoir encore améliorer les performances de l'algorithme, à la fois du point de vue des erreurs de classification mais également de sa complexité. Dans le premier cas, nous allons utilisé un seuil local à chaque donnée, ce qui évitera que des données se place rapidement à la fin de l'algorithme. Pour la complexité, nous comptons exploiter le fait que des états ne changent plus au bout d'un certain nombre d'itérations, et qu'il devient donc inutile de faire évoluer ces cellules. Le nombre de cellule à traiter pourrait donc diminuer au cours du temps.

Références

Azzag H., Picarougne F., Guinot C., Venturini G., *Un survol des algorithmes biomimétiques pour la classification*. Classification et Fouille de Données, pages 13-24, RNTI-C-1, Cépaduès. 2004.

Blake C.L., Merz C.J., *UCI Repository of machine learning databases*. http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science. 1998.

Chen, H.; Schuffels, C.; Orwig, R.: Internet categorization and search: a self-organizing approach. In: Journal of visual communication and image representation, 7 (1996) 1, p. 88-102.

Cugini, J.: Presenting Search Results: Design, Visualization and Evaluation. In: Workshop: Information Doors - Where Information Search and Hypertext Link (held in conjunction with the ACM Hypertext and Digital Libraries conferences) Conference: San Antonio, TX, May 30 2000.

Ganguly N., Sikdar B. K., Deutsch A., Canright G., Chaudhuri P. P., *A Survey on Cellular Automata*. Technical Report Centre for High Performance Computing, Dresden University of Technology, December 2003.

Gardner M., *Mathematical Games: The fantastic combinations of John Conway's new solitaire game 'life'*. Scientific American, pages. 120-123, Octobre 1970.

Guinot C., Malvy D. J.-M., Morizot F., Tenenhaus M., Latreille J., Lopez S., Tschachler E., et Dubertret L.. *Classification of healthy human facial skin*. Textbook of Cosmetic Dermatology Third edition (to appear), 2003.

Han Eui-Hong., Boley Daniel., Gini Maria., Gross Robert., Hastings Kyle., Karypis George., Kumar ipin., Mobasher Bamshad., et Moore J. *Webace : a web agent for document categorization and exploration*. In AGENTS '98 : Proceedings of the second international conference on Autonomous agents, pages 408–415, New York, NY, USA, 1998. ACM Press.

Handl Julia, Bernd Meyer: Improved Ant-Based Clustering and Sorting. Parallel Problem Solving from Nature - PPSN VII, LNCS 2439, pp 913-923

Jain A. K., Murty M. N., Flynn P. J., *Data clustering: a review*, ACM Computing Surveys, 31(3), pages. 264-323, 1999.

Kohonen, T: Self-organization of very large document collections: State of the art. In: Niklasson, Lars; Bodén, Mikael; Ziemke, Tom (Eds.): Proceedings of ICANN98, the 8th International Conference on Artificial Neural Networks. Conference: Skövde, Sweden, 1998 September 2-4 Springer (London) 1998. p. 65-74, vol. 1. (lien de la démonstration http://websom.hut.fi/websom/stt/doc/eng/)

Lumer E., Faieta B., *Diversity and adaption in populations of clustering ants.* In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 3, pages 501-508. MIT Press, Cambridge, MA, 1994.

Mokaddem F., F. Picarougne, H. Azzag, C. Guinot, G. Venturini, Techniques visuelles de recherche d'informations sur le Web, à paraître dans *Revue des Nouvelles Technologies de l'Information*, numéro spécial Visualisation en Extraction des Connaissances, Pascale Kuntz et François Poulet rédacteurs invités, Cépaduès.

Von Neumann J., *Theory of Self Reproducing Automata.*, University of Illinois Press, Urbana Champaign, Illinois, 1966.

Roussinov, D., Tolle, K., Ramsey M., McQuaid, M., and Chen, H., "Visualizing Internet Search Results with Adaptive Self-Organizing Maps," Proceedings of ACM SIGIR, August 15- 19, 1999, Berkeley, CA.

Salton G., Yang C. S., et Yu C. T., A theory of term importance in automatic text analysis. Journal of the American Society for Information Scienc, 26(1):33–44, 1975.

Wise, J. A.: The Ecological Approach to Text Visualization. In: Journal of the American Society for Information Science (JASIS), 50 (1999) 13, p. 1224-1233

Zamir O, O. Etzioni. Grouper : a dynamic clustering interface to Web search results. Computer Networks (Amsterdam, Netherlands : 1999), 31(11–16) :1361–1374, 1999.

Classification et visualisation des données d'usages d'Internet

Khalid Benabdeslem*, Younès Bennani**

*IBCP-CNRS, 7 Passage du vercors, 69367 Lyon Cedex07, France
kbenabde@ibcp.fr
**LIPN-CNRS, 99 Avenue J-B. Clément, 93430 Villetaneuse, France
younes.bennani@lipn.univ-paris13.fr

Résumé. Dans ce papier, nous présentons une nouvelle méthode de classification des données de navigation dans les sites Web. Nous présentons plus particulièrement, le développement d'une approche d'apprentissage non supervisé pour ce type de données stockées sous forme de traces dans des fichiers Log.
La première partie de cette étude concerne le problème du codage des données qui seront ensuite utilisées pour l'analyse. En effet, actuellement, les site web sont dynamiques et les pages ne peuvent pas être caractérisées par des variables fixes comme : la hiérarchie des adresses URL, le contenu, etc. Elles sont représentées par des adresses numériques n'ayant aucun « sens » et qui servent à récupérer des informations dans des bases de données pour remplir les contenus des pages. Pour cette raison, nous proposons une nouvelle méthode de codage de sessions à partir du fichier Log. Cette technique consiste à caractériser une page donnée par un vecteur de poids d'importance de passage, i.e. par ses poids de précédence et de succession relatives à toutes les autres pages qui apparaissent dans le fichier Log.
Dans la deuxième partie de ce travail, nous analysons les propriétés des cartes topologiques de Kohonen et nous proposons une version adaptée aux données comportementales. Cette étape nous permet (1) de construire une cartographie du site web tel qu'il est aperçu par les clients (2) de regrouper les pages pour un objectif de codage de sessions (3) et de projeter les interactions des clients du fichier Log sur la cartographie sous forme de trajectoires symbolisant leurs comportements.

1 Introduction

Les sites Web représentent actuellement une véritable source de production de grands volume d'informations. Cependant, ce gisement d'information ne représente pas une évidence de compréhension des utilisateurs qui se retrouvent généralement perdus devant de telles quantités d'informations Zeboulon et al. (2001). Une technique de E-Mining est donc nécessaire pour comprendre les interactions des utilisateurs pour répondre aux mieux à leurs besoins prioritaires. Le E-Mining est une chaîne complète de fouille de données qui permet d'analyser des formes basées sur des interactions pour extraire des connaissances sur les comportements des utilisateurs dans les sites. Dans ce contexte, nous définissons une session d'utilisateur comme une séquence temporelle des pages qui l'ont intéressé durant son parcours dans le site.

Par ailleurs, comprendre le comportement des utilisateurs dans les sites est devenu un souci majeur pour les propriétaires de ces sites. Cette compréhension se traduit par

l'adaptation de leurs sites selon les comportements et les besoins des utilisateurs. Pour se faire, l'exploitation de toute sources d'informations sur les utilisateurs est nécessaire pour bien les comprendre. Cependant, pour des raisons de disponibilités et de coût, nous nous contentons dans cette étude d'une source d'informations enregistrées dans un fichier fournis par le serveur du site. Ce fichier appelé Fichier Log, affiche plusieurs types de caractéristiques (temps, adresse IP, Url des pages visitées, etc). Ce fichier permet de reconstituer des sessions des utilisateurs ayant visité le site associé.

Le travail présenté dans ce papier a deux objectifs : d'une part, de trouver une méthode de codage pour les données basées sur les séquences en prenant en compte la dépendance entre les composantes de ces séquences ; et d'autre part de trouver une méthode de classification incrémentale capable de traiter de nouveaux fichiers Log tout en préservant le modèle initial.

2 Données de navigation et codage

Une des importantes sources d'informations sur la navigation dans un site Web est le fichier Log. Connu aussi sous le nom du « Accès Log » où sont enregistrées toutes les transactions entre le serveur et le navigateur.

Il existe plusieurs formats de représentation d'un fichier Log. La méthode standard par laquelle les serveurs enregistrent les accès par les navigateurs est intitulée : « *Common Log Applications* » créée par NCSA (National *Center for Supercomputing Applications*).

Le fichier Log regroupe plusieurs sessions de plusieurs utilisateurs d'un site. Une session d'utilisateur est définie comme une séquence temporelle d'accès aux pages du site par un seul utilisateur. L'identificateur de cet utilisateur est rarement fourni par le serveur. Pour cette raison, nous définissons une session utilisateur comme un accès de la même adresse IP, pourvu que le temps entre deux pages consécutives ne dépasse un seuil de visualisation.
Chaque URL dans le site est assignée par un index $j \in \{1, \ldots\ldots, N\}$
Où $\boldsymbol{N}$ représente le nombre total de pages.
Globalement, une session peut être codée de la manière binaire suivante :

$$S_j^{(i)} = \begin{cases} 1 & \text{si l'utilisateur demande la } j^{\text{ème}} \text{ URL durant la } i^{\text{ème}} \text{ session} \\ 0 & \text{sinon} \end{cases}$$

Ce codage ne traite pas les mouvements dans le site et n'explicite pas l'intérêt de l'utilisateur devant les pages. Il existe d'autres méthodes de codage Trousse (2000) qui consistent à pondérer des mesures associées aux adresses hiérarchiques des pages et à leurs contenus. Cependant, ces méthodes ne sont pas adéquat à la plupart des sites conçus aujourd'hui à cause de leurs structures aléatoires dues à l'aspect dynamique (accès aux base de données, adressage aléatoire, contenu personnalisé, etc.). Pour cette raison, nous avons développé une autre méthode de codage à partir du fichier Log, qui consiste à caractériser une page par son importance de passage. En d'autres termes, par son poids de précédence et de succession par rapport aux autres pages du site apparues dans le fichier Log.

Le principe de cette méthode consiste à calculer pour chaque page sa fréquence de précédence et de succession par rapport à toutes les autres pages et de regrouper ces fréquences

dans un seul tableau de données de taille égale au *Nombre de page* $\times$ *(Nombre de pages précédentes + Nombre de pages suivantes)*.

Nous avons remarqué que ce codage tel qu'il est d écrit cause un problème d'effectif. En effet, nous ne disposons pas d'assez d'URLs significatives dans le site, nous ne pouvons pas avoir un tableau avec un nombre de données suffisantes pour l'apprentissage.

Pour remédier à ce problème, nous avons proposé de glisser la matrice sur le mois. En d'autres termes, nous avons calculé une matrice de codage par jour, voire même par semaine ou généralement par pourcentage de partitionnement sur la base. Cette méthode nous permet de multiplier le nombre d'échantillons et d'avoir plusieurs exemples pour chaque URL.

URLs	Jour	E	URL_1.... URL_j.... URL_N	URL_1.... URL_k.... URL_N	S
URL_1					
URL_2					
.....					
URL_i					
.....					
URL_N					
URL_1					
URL_2					
.....					
URL_i	*k*	*a*	*b*	*c*	*d*
.....					
URL_N					
….					
URL_1					
URL_2					
.....					
URL_i					
.....					
URL_N					

TAB. 1 – *Codage des URLs par rapport aux sessions dans le fichier Log.*

Dans le tableau ci-dessus, E représente une variable caractérisant le nombre de fois qu'une URL donnée apparaît en entrée (respectivement, S en sortie).

A titre d'exemple, *Dans le* $k^{iéme}$ *jour du fichier l'*URL_i *est apparue a* fois comme page d'entée, précédée *b* fois par la page URL_j, suivie *c* fois par la page URL_k et apparue *d* fois comme page de sortie.

Par ce codage, Non seulement, on caractérise les pages par des variables comportementales i.e. tel que les internautes perçoivent le site. Mais on multiplie aussi les informations sur les pages en codant leurs intérêts sur les pages jour par jour.

3 Classification incrémentale et visualisation

Les techniques numériques de reconnaissance de formes sollicitent une méthode de classification améliorée pour comprendre, interpréter et simplifier les grandes quantités de données multidimensionnelles. La plupart du temps, K-Means et d'autres méthodes de partionement sont utilisées dans les applications industrielles Ribert et al. (1999). Cependant, elles présentent l'inconvénient majeur de déterminer une solution finale indépendante des conditions initiales, spécialement concernant le nombre des classes. Cette connaissance préalable est rarement disponible pour les utilisateurs. En effet, s'ils utilisent une technique de classification c'est parce qu'ils ignorent la structure de leurs données. Par conséquent, cette contrainte est généralement intraitable. Par ailleurs, les techniques de classification considèrent que les bases de données sont complètement représentatives aux problèmes. Or c'est rarement le cas, quand il s'agit de traiter des problèmes complexes. En d'autres termes, d'un point de vue pratique, un problème complexe est souvent incrémental. Par conséquent, il devient très important, de concevoir des systèmes de classification incrémentale capables de prendre en compte les formes qui ne sont pas disponibles au moment de la constitution du modèle initial par les données initiales. Introduire l'aspect d'incrémentalité n'est pas nouveau. Il existe plusieurs méthodes qui mettent à jour dynamiquement des modèles de classification. Cependant, ces méthodes ont quelques limites. Par exemple dans IGG (Incremental grid growing) Blackmore (1995), bien que le processus soit incrémental, il utilise toujours la même base à chaque adaptation de la grille courante. L'aspect incrémental concerne donc, la topologie sans s'occuper de l'arrivée dynamique des données. La méthode GNG (Growing Neural Gas) dans Fritz (1994) contruit quant à elle, le modèle en tenant compte des nouvelles formes. Cependant cette méthode souffre d'un problème de lissage. Cela veut dire que, après sa création, la carte est représentée par des sous-ensembles de neurones indépendants. En effet, entre chaque paire de sous-ensembles, des neurones ne possédant pas de connexions sont définitivement supprimés. Cela dit, ces neurones n'auront jamais la possibilité d'être activés par un éventuel ensemble de données qui risquent d'arriver ultérieurement.

3.1 Un bref rappel sur SOM

L'algorithme SOM de Kohonen représente un véritable outil de visualisation des données multidimensionnelles. Il permet de convertir les relations statistiques complexes et non linéaires en relations géométriques simples dans une carte bidimensionnelle. De plus cet algorithme permet de compresser l'information tout en gardant les relations topologiques et métriques les plus pertinentes dans l'espace de données réel.

L'apprentissage dans les cartes topologiques se fait avec une fonction de voisinage. Il procède en 3 étapes :

Initialisation : il s'agit de l'initialisation des poids.

Compétition : à chaque entrée d'un exemple au réseau, un calcul de distance est effectué pour activer un neurone dit gagnant, c'est celui dont le potentiel d'activation est le plus fort en fonction de l'entrée.

Adaptation : le choix d'un nœud particulier permet alors d'ajuster les poids localement, en minimisant la différence qui existe encore entre les poids et le vecteur d'entrée. Cet ajustement se fait suivant une forme de voisinage qui peut être carrée, ronde ou hexagonale.

Le processus est donc constitué de ces trois phases qui sont itérées jusqu'à la minimisation d'une erreur globale calculée sur l'ensemble du réseau ou sur un nombre de cycles d'apprentissage fixé empiriquement.

3.2 Une version incrémentale de SOM : e-SOM

Dans cette section, nous proposons une nouvelle version des cartes topologiques. Cette version possède un aspect incrémental « complet ». En effet, il s'agit de créer la topologie en fonction de l'arrivée des données. Nous appelons cette version e-SOM.

Notre problématique est due à l'arrivée de plusieurs masses de données après la création du modèle initial. Au lieu de refaire tout le modèle, nous proposons de garder l'existant est de l'incrémenter (le mettre à jour), en fonction des nouvelles vagues de données qui arrivent.

3.2.1 Création et gestion de nouveaux neurones

Nous proposons tout d'abord d'étudier la distribution des nouvelles données sur la carte initiale. Pour cela nous appliquons la procédure de *compétition* de SOM (i.e. le classement des données dans la carte). Nous trouvons, par exemple, 10% de données sur le premier neurone, 24% sur le $i^{émé}$ neurones…etc

De manière statistique, nous marquons les neurones qui appartiennent au périmètre de la grille et qui possèdent une densité (effectif) supérieur ou égal à N/C. Tels que C représente le nombre des neurones de la carte et N le nombre d'exemples dans la nouvelle base en cours.

Intuitivement, nous ne nous intéressons qu'aux neurones du périmètre, car nous estimons que les neurones situés à l'intérieur de la carte, sont déjà enfermés par leur voisinage, alors que ceux des frontières possèdent une possibilité de voisinage en dehors de la carte. Par exemple, le neurone en bas à gauche, possède deux possibilités de voisinage (un au dessous et un autre à gauche)

De plus, l'ajout des neurones dépend du voisinage du neurone marqué (un neurone est dit marqué, s'il est actif et autorisé à un voisinage).

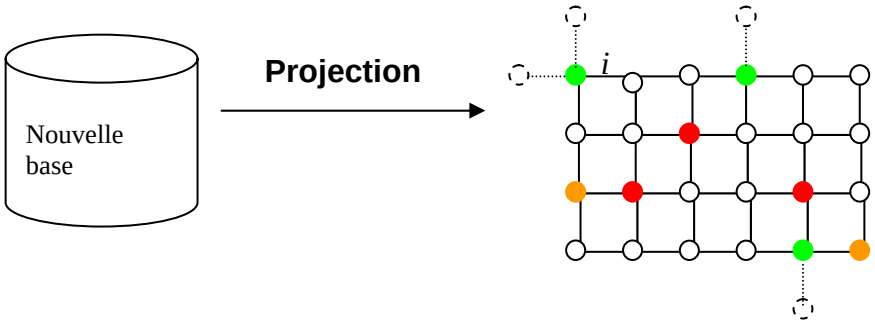


FIG. 1 – *Gestion des nouveaux neurones crées pour l'apprentissage incrémental*

- Neurone actif autorisé à avoir de nouveaux voisins
- Neurone actif non marqué, car sa densité n'est pas importante
- Neurone actif mais non autorisé à un nouveau voisinage à cause de sa position dans la carte
- Nouveau neurone « provisoirement » créé.

Les nouveaux neurones crées sont ainsi initialisées par la formule suivante :

$$W^*_j(k) = \frac{1}{2}\left(\frac{1}{N'}\sum_{i=1}^{N'} V_i(k) + W_j(k)\right) + \varepsilon$$

N_c : est le nombre de neurones de la carte initiale
$W^*_j(k)$: est le $k^{ème}$ élément du vecteur poids du nouveau neurone créé au voisinage du neurone *j(j est le neurone marqué)*
N' : est l'effectif de la nouvelle base
$Vi(k)$: est le $k^{ème}$ élément du $i^{ème}$ vecteur de la nouvelle base.
$W_j(k)$: est le $k^{ème}$ élément du vecteur poids du neurone *j* après l'apprentissage de la première carte.
ε : est un nombre aléatoire tel que : $0 \leq \varepsilon \leq 1$.

3.2.2 Mise à jour de la topologie

Une fois le nouveau voisinage organisé, il ne reste qu'à entraîner la carte avec la nouvelle base. Nous répétons donc, le processus SOM sur la nouvelle base pour l'adaptation générale sur les anciens et les nouveaux neurones. Cela fait référence à un apprentissage par morceau (de neurones de la carte) en fonction des données disponibles.

Le processus incrémental est itératif à chaque arrivée de nouvelles bases de données, ce qui explique la dynamique de la restructuration de la carte.

3.2.3 Optimisation

Une fois l'apprentissage incrémental fini à chaque arrivée d'une base, on réutilise la procédure de compétition de SOM pour déterminer l'activation des neurones marqués selon le principe de la section 3.2.1. (FIG. 1)

Si après cette compétition, le neurone créé est inactif, il est directement supprimé sinon il est gardé , et il devient donc « réel ».

Nous présentons la formulation algorithmique qui interprète cette nouvelle version :

-Application de SOM (Création de la première carte: C_0) sur la première base de données (BDD_0)
-Initialisation des poids
-Compétition des formes par rapport au poids
-Adaptation des poids

-Mise à jour de la topologie (i=1)
-$\forall z \in BDD_i$, $\forall j \in C_0$, Compétition (z, j).

-Si j ∈ périmètre (C_0) Alors
-Si proportion (j) ≥ (100×C/N) % Alors
Création (j', voisinage(j))
Finsi
Finsi

-Initialisation des nouveaux neurones

$$W^*_{j'}(k) = \frac{1}{2}\left(\frac{1}{N'} \sum_{i=1}^{N'} V_i(k) + W_j(k) \right) + \varepsilon$$

-Adaptation
Adaptation selon SOM de C_i , ∀j
// $C_i = C_{i-1} \cup \{j\}$//

-Remise à jour
Éliminer tout j' , tel que Compétition (j', C_i) = ∅
i = i+1, si i < M Alors Retour à 2 sinon Arrêt
M : représente le nombre de sous – bases.

3.3 Post-classification de e-SOM

Comme l'algorithme classique de SOM, e-SOM fournit une topologie de neurones représentés par les vecteurs de poids (souvent appelés référents). Chaque vecteur est comparé par rapport à tous les exemples de la base pour un objectif de groupement (Clustering). Cependant, on peut trouver quelques neurones inactifs à l'intérieur de la carte à cause du nombre fixé a priori à cause de la topologie initiale faite par SOM (première étape de e-SOM). Ceci étant, pour optimiser ce nombre, nous proposons de une classification des référents par la méthode de la classification ascendante hiérarchique (CAH) Bouroche et al. (1994) (FIG. 2). Dans ce cas, si deux exemples X_1, X_2 activent respectivement deux neurones N_1, N_2 et N_1, N_2 appartiennent à la même classe formée par CAH, nous considérons que X_1 et X_2 sont de la même classe dans la cartographie.

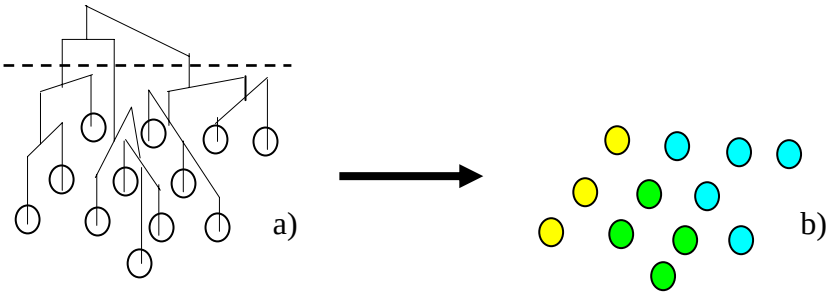


FIG. 2. *Post – classification de e-SOM par CAH. a) Carte faite par e-SOM (12 neurones) .b) Carte optimisée par regroupement de ces neurones (3 classes).*

4 Résultats

4.1 Les données

Pour notre application, nous utilisons un site commercial nommé : www.123credit.com. Il s'agit d'un site qui commercialise un certain nombre de services pour des clients désirant avoir des crédits (Auto, Maison, terrain, etc.). Le fichier Log a été fourni dans son état brut en 4 temps. Ce fichier dans son intégralité, décrit des transactions de navigation pendant 1 mois. Son volume avoisine les 700 Méga octets.

Après prétraitement et codage selon le principe décrit dans la section 2, nous extrayons un nombre de sessions égal à 37174 dont le nombre d'Urls est égal à 40. Le fichier commence donc à perdre de sa taille. Une session est une succession d'enregistrements ayant la même adresse IP et ne dépassant pas un seuil de visualisation paramétrable (30 secondes pour notre expérience) [Ref, ma these].

Le fichier ayant des interactions de 1 mois (soit 30 jours), notre un tableau de données représente (40 × 30) exemples, caractérisés par ((40×2)+2) Urls. Soit donc, un tableau de 1200 individus sur 82 variables.

Il est bien évident que ce dernier tableau contient des entiers discrets. Nous normalisons le tableau en colonnes pour centrer et réduire les données. Nous proposons aussi, de le normaliser en lignes pour éliminer toute dépendance entre les variables.

4.2 Classification et visualisation

Le fichier étant fourni en 4 morceaux, nous avons commencé avec un premier codage d'une première base qui était initialement disponible et nous l'avons incrémenté au fur et à mesure de l'arrivée des 3 autres (en moyenne 300 exemples obtenus après l'arrivée de chaque nouvelle base). Ces bases sont ainsi présentées de manière itérative à e-SOM pour former des classes de pages selon les interactions des utilisateurs dans le site. Ensuite, l'algorithme CAH est appliqué sur les référents pour optimiser le nombre de neurones de la carte. Nous obtenons donc des classes de neurones. Cependant, après la projection des exemples dans la carte obtenue, chaque Url peut activer plus qu'un seul neurone (selon sa parution pendant les jours du mois). Intuitivement, si la topologie est bien optimisée, les neurones activés par la même Url devraient être voisins dans la carte. Pratiquement, pour étiqueter la carte nous procédons par vote majoritaire.

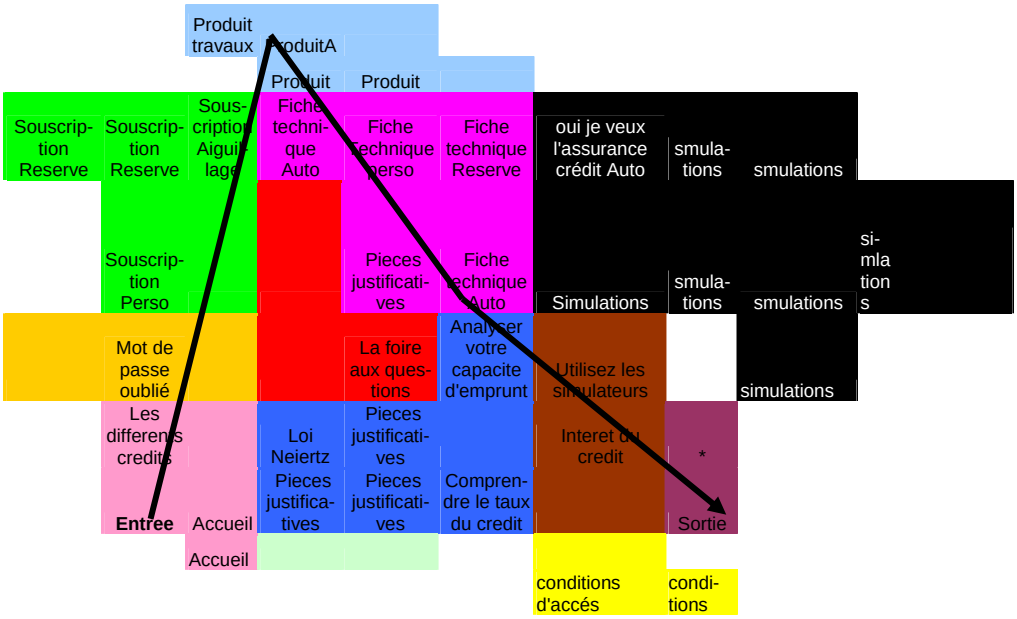


FIG. 3. *Visualisation du site www.123credit.com tels que les internautes le perçoivent*

FIG3 montre l'application de e-SOM sur les données de navigation enregistrées dans le fichier Log du site. Chaque couleur est spécifique à une classe de neurone regroupant les pages susceptibles d'intéresser un profil donné d'utilisateurs. Nous pouvons aussi remarquer que la carte n'est pas uniforme (rectangulaire ou hexagonale comme dans SOM) à cause des nouveaux neurones sui sont créés dynamiquement selon la disponibilité des bases.

Nous avons aussi comparé e-SOM avec l'algorithme classique de SOM ainsi que d'autres version incrémentales de SOM comme IGG et GNG (TAB. 2). Cette comparaison est faite sur quatre critères :

Méthode	N	Lissage	TE	GDR
SOM	960	Oui	0.3	0.46
IGG	850	Oui	0.2	0.38
GNG	230	Non	0.15	0.20
e-SOM	240	Oui	0.10	10.23

TAB. 2 – *Comparaison entre différentes versions incrémentale de SOM*

- N : représente le nombre de neurones dans la carte. Pour SOM, ce nombre est calculer selon l'heuristique de Kohonen ($N= 5 \times (\text{Nombre d'exemples})^{0.54321}$). Pour les versions incrémentales, ce nombre est calculé dynamiquement chacune selon son algorithme.

- TE : représente l'erreur topologique pour représenter la qualité de la carte. Cette mesure représente le pourcentage que des données pour lesquelles le premier neurone activé et le deuxième ne sont pas adjacents dans la carte.
- GDR= $\frac{E_i - E_f}{E_i}$: la décroissance de l'erreur quadratique moyenne sur la carte. E_i et E_f représentent respectivement l'erreur initiale et l'erreur finale de l'apprentissage.

De plus, FIG3 fournit un outil de visualisation des différentes interactions des utilisateurs du site. Ces interactions sont représentées par la projection des sessions du fichier Log vers la carte obtenue. Par exemple, le chemin illustrées dans la figure représente 40% des utilisateurs qui (1) entrent dans le site par la page d'accueil (Cela montre que le site est assez bien connu) (2) visitent les produits de la catégorie A (crédits pour voitures) (3) demandent plus d'informations sur ces produits (4) sortent du site sans l'achat d'aucun service. Par ce cheminement, nous pouvons constater que les différentes informations ou les hyper liens dans les pages associés à ces produits doivent être mis en question et donc modifiés car elles ne sont pas attractives pour les utilisateurs. Ce genre d'analyse de profils est ainsi réalisé facilement sur l'ensemble de tous les utilisateurs du site par une simple projection de leurs interactions sur la cartographie obtenue par e-SOM.
Nous pouvons remarquer dans TAB. 2 que e-SOM donne généralement de meilleurs résultats que ses concurrentes. En effet, le nombre final de neurones obtenus est optimal, l'erreur topologique et l'erreur quadratique sont considérablement réduites et la propriété du lissage est respectée grâce à la topologie initiale faite par SOM sur la base initiale. Cette propriété est importante, notamment pour de nouvelles données susceptibles d'activer les neurones à l'intérieur de la carte, est absente dans la méthode GNG qui est relativement l'autre meilleure approche particulièrement sur N et GDR.

5 Conclusion

Ce travail nous a permis d'analyser les comportements des internautes faces à des sites Web. Nous avons tout d'abord, proposé une nouvelle méthode de codage basée les interactions enregistrées dans le fichier Log. Ensuite, Nous avons développé une idée qui consiste à traiter les nouvelles données qui arrivent, en utilisant le même modèle issu des données antérieures. En d'autres termes, nous avons rendu dynamique la création du modèle de la classification (la carte). Il s'agit donc, d'une carte qui change de figure dans le temps.

L'algorithme que nous avons proposé, respecte les mêmes règles de la création des cartes topologiques SOM (initialisation, compétition et adaptation) et permet de rendre la création des neurones de la carte de manière dynamique et dépendante de la nature des informations qui arrivent dans le temps.

Nous avons donc, rendu évolutive, la découverte de l'espace topologique : i.e. le nombre de regroupement homogènes (clusters) change dans le temps en fonction des informations complémentaires qui arrivent. Une procédure tout à fait intéressante pour une mise à jour intelligente de la cartographie.

Références

K. Benabdeslem, Y. Bennani and E. Janvier. Connectionnist approach for Website visitors behaviors minin. it In Proceedings of ACS/IEEE International Conference on Computer Systemes and Applications, Lebanon.2001.

K. Benabdeslem, Y. Bennani and E. Janvier. Visualization and analysis of web navigation data in LNCS2415 (Springer), pp 486-491, Madrid, 2002.

J-M. Bouroche and G. Saporta, L'analyse des donnees, Presse universitaire de France, 1994.

Y. Bennani. Multi-expert and hybrid connectionnist approach for pattern recognition: speaker identification task. International Journal of Neural Systems, Vol.5, No. 3, 207-216.1994.

J. Blackmore. Visualizing high dimensional structure with the incremental grid growing neural network. technical report, departement of computer sciences of the university of Texas, Austin, 1995.

I. Cadez, D. Heckerman, C. Meek, P. Smyth, S.White. Visualization of Navigation Patterns on a Web Site Using Model Based Clustering. it In proceedings of the KDD, 2000.

M. Deshpande, G. Karypis. Selective Markov Models for predicting Web-page accesses. 1st SIAM conference on Data Mining, Chicago, Illinois, 2001.

E. Forgy, "Cluster analysis of multivariate data : efficiency versus interpretability of classifications", Biometrics, Vol. 21, 768, 1965.

C. Fraley and A. Raftery. How many clusters ? Which clustering method ? Answers via model-based cluster anlysis. Computer Journal, 41, 578-588.1998.

B. Fritz. Growing cell structures- A self organizing network for unsupervized learning. Neural Networks, Vol 7, N9, pp 1441,1460, 1994.

K. Hattori and Y.Torii. "Effective algorithms for the nearestneighbor method in the clustering problem", Pattern Recognition, Vol. 26, N°5, pp. 741-746, 1993.

T. Kohonen. Self-Organizing Maps. Series in Information Sciences, Vol. 30. Springer, Heidelberg. 1995.

T. Martinez and K. Schulten. A neural gas network learns topologies. Artificial Neural Networks, Elsevier Science Publishers B.V, pp 397,402, 1991.

G. McLachlan and K. Basford. Mixture Models : Inference and Applications to Clustering. Marcel Dekker, 1988.

M. Perkowitz and O. Etzioni. Towards adaptative web sites : conceptual framework and case study. Artificial Intelligence Journal, 118 ,1-2.2000.

A. Ribert, A. Ennaji and Y. Lecourtier. An Incremental Hierarchical Clustering. Vision Interface'99, Trois riviÃ¨res, Canada, 1999.

B. Trousse - Evaluation of the Prediction Capability of a User behaviour Mining Approach for Adapative Web Sites. In RIAO 2000, 6th Conference on "Content-Based Multimedia Information Access", College de France, Paris, France, April pp12-14, 2000.

A. Zeboulon. Reconnaissance et classification de séquences. DEA127's internship report, University of Paris 9, 2001.

A. Zeboulon Y. Bennani and K. Benabdeslem. Hybrid connextionnist approach for knowledg discovery frow web navigation pattern. In the proceedings of ACS/IEEE International Conference on Computer Systems and Applications, Tunisia, July 2003.

Extraction des connaissances à partir des fichiers Logs

Malika Charrad*, Mohamed Ben Ahmed*, Yves Lechevallier**

*Ecole Nationale des Sciences de l'Informatique, Laboratoire RIADI
Université de la Manouba, 1010 La Manouba
{malika.charrad, mohamed.benahmed}@riadi.rnu.tn
** INRIA – Institut National de Recherche en Informatique et en Automatique
Domaine de Voluceau-Rocquencourt, B.P.105, 78153 Le Chesnay Cedex, France
yves.lechevallier@inria.fr

Résumé. L'approche que nous proposons de caractériser les utilisateurs d'un site Web en se basant sur leurs motifs de navigation sur le site comporte trois phases : prétraitement des fichiers Logs, classification des pages et classification des internautes. Dans la phase de prétraitement, les requêtes sont organisées en visites. Dans la phase de classification des pages, des paramètres introduits à partir des statistiques sur les accès aux pages sont utilisés pour la catégorisation des pages Web en pages auxiliaires et pages de contenu. Les requêtes aux pages de contenu servent à la découverte des motifs de navigation. Pour construire des groupes d'utilisateurs, deux méthodes hybrides de classification automatique basées sur l'analyse en composantes principales, l'analyse des correspondances multiples et les cartes de Kohonen sont appliquées aux visites. Une expérience effectuée sur des données réelles prouve l'efficacité de cette méthodologie.

1 Introduction

Au cours de ces dernières années, la croissance exponentielle du nombre des documents en ligne a entraîné une croissance rapide de l'activité sur le Web, et une explosion des données résultant de cette activité. En effet, le nombre des utilisateurs d'Internet dans le monde a atteint 972.8 millions au mois de Novembre 2005[1], et le nombre de sites Web a atteint 74.4 millions au mois d'Octobre 2005[2]. Ces données, en particulier celles relatives à l'usage du Web, sont traitées dans le Web Usage Mining (WUM). Dans cet article, nous nous intéressons à l'analyse des fichiers Logs afin de comprendre le comportement des internautes sur un site Web. L'apport de ce travail réside principalement dans trois points :

- Utiliser plusieurs heuristiques pour l'identification des robots Web et l'identification des images dans la phase du prétraitement des fichiers Logs.
- Associer la classification des pages à la classification des usagers du site Web. En d'autres termes, exploiter les résultats de la classification des pages dans la classification des internautes.

[1] www. Internetworldstats.com
[2] www.netcraft.com

- Intégrer et combiner deux types de méthodes de fouille des données, deux méthodes factorielles et une méthode neuronale pour la classification des utilisateurs.

Cet article est organisé en trois sections distinctes. La première section présente les différentes étapes du prétraitement des fichiers Logs ainsi que les résultats de leur application sur des données réelles. La deuxième section présente une méthodologie de classification des pages Web. La dernière est consacrée à la classification des utilisateurs du site Web étudié.

2 Approche proposée

Récemment, de nombreux travaux en Web Usage Mining ont été menés. Certains de ces travaux se sont intéressés à la phase du prétraitement des données du Web tels que les travaux de Tanasa (2003) et Srivastava (2000) ; d'autres travaux sont concentrés sur la détermination des modèles comportementaux des internautes fréquentant les sites Web. Ce second axe est traité dans les travaux de Pierrakos (2003) et Mobasehr (2002). Notre approche consiste à intégrer la classification des pages dans la classification des utilisateurs. En d'autres termes, exploiter les résultats de la classification des pages dans la classification des utilisateurs.

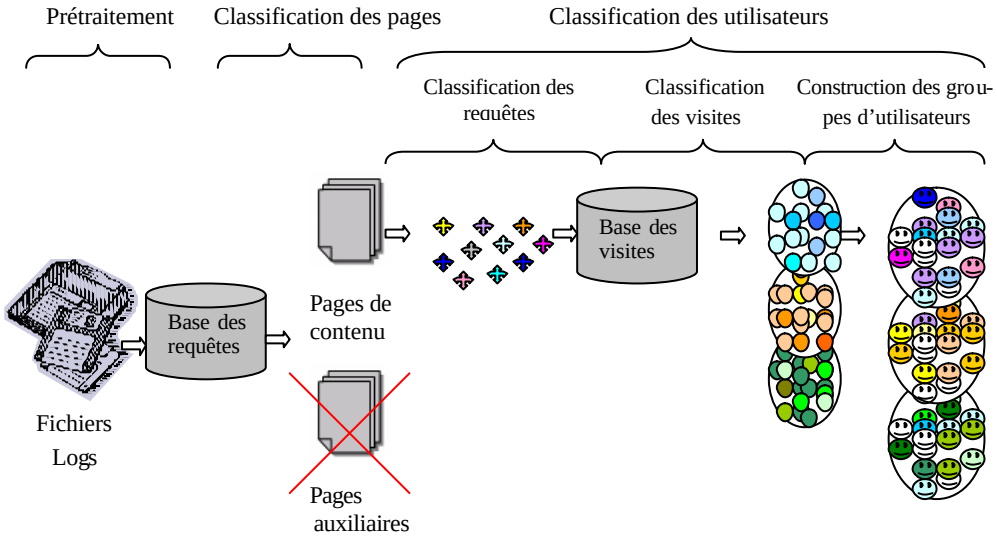


FIG. 1 - *Etapes de classification des utilisateurs.*

2.1 Prétraitement des données

2.1.1 Nettoyage des données

Le nettoyage des données consiste à supprimer les requêtes inutiles des fichiers logs à savoir :

- Les requêtes non valides. Ce sont les requêtes dont le statut est inférieur à 200 ou supérieur à 399 ;

- Les requêtes provenant des robots Web. Il est presque impossible aujourd'hui d'identifier tous les robots Web puisque chaque jour apparaissent des nouveaux. Pour identifier les requêtes provenant des robots, nous avons utilisé cinq heuristiques, dont les trois premières ont été utilisées par Tanasa et al. (2003), en considérant qu'il suffit de vérifier l'une de ces heuristiques pour considérer la requête correspondante comme étant générée par un robot Web :
  - Identifier les adresses IP et les « User-Agents » connus comme étant des robots Web. Ces informations sont fournies généralement par les moteurs de recherche[3],
  - Identifier les adresses IP ayant fait une requête à la page « \robots.txt »,
  - Utiliser un seuil pour la vitesse de navigation BS « Browsing Speed » égale au nombre de pages visitées par seconde. Le calcul du Browsing Speed n'est possible qu'après détermination des sessions et des visites.
  - Identifier les « User-Agents » comportant l'un des mots clés suivants : « crawler », « spider » ou encore « bot »,
  - Identifier les requêtes effectuées par des aspirateurs de sites Web (HTTrack par exemple), ou par des modules de certains navigateurs permettant la consultation de pages hors ligne tels que DigExt d'Internet Explorer. L'identité de ces aspirateurs ou de ces modules est trahie par la mention de leurs noms au niveau de leurs User-Agents. Pour les aspirateurs qui cachent leurs User-Agents, leur identification est effectuée en se basant sur la durée de leurs requêtes généralement nulle.
- Les requêtes aux images. Cette étape de nettoyage consiste à supprimer les fichiers dont les extensions sont : .jpg, .gif, .png, etc... et les fichiers multimédia dont l'extension est : .wav, .wma, .wmv, etc… Deux méthodes ont été utilisées pour supprimer les requêtes aux images. La première consiste à utiliser la carte du site afin d'identifier les URLs des images nécessitant de cliquer sur un lien pour être affichées. Les images inclues dans les fichiers HTML sont supprimées car elles ne reflètent pas le comportement de l'internaute. Cependant, ce n'est pas toujours possible d'identifier toutes les images inintéressantes quand le site est volumineux. Dans ce cas, nous proposons une autre méthode dont l'application nécessite tout d'abord l'identification des sessions. Cette méthode consiste à supposer qu'un utilisateur ne peut cliquer à la fois (au même instant) sur plusieurs images pour les visualiser ; Tenant compte de cette hypothèse, nous déterminons pour chaque utilisateur l'ensemble des requêtes effectuées au même instant. Les requêtes correspondantes à des fichiers images sont éliminées ;
- Les requêtes dont la méthode est différente de « GET » ;
- Les scripts. Généralement, le téléchargement d'une page demandée par un utilisateur est accompagné par le téléchargement automatique des scripts tels que les scripts Java (fichiers .js), des feuilles de style (fichiers .css), des animations flash (fichier .swf),…etc. Ces éléments doivent être supprimés du fichier Log étant donné que leur apparition ne reflète pas le comportement de l'internaute ;

[3] Informations obtenues au site http://www.searchturtle.com/search/Computers_Internet/Robots/ , http://www.robotstxt.org/wc/active/html/index.html et http://www.iplists.com/

- Les requêtes spécifiques à l'activité sur le site. Ce sont les requêtes relatives au trafic sur le site objet de l'analyse. Cette étape montre que la méthodologie d'analyse du comportement des internautes sur le Web n'est pas unique et qu'elle dépend de plusieurs facteurs, en particulier du site analysé. Par exemple, en considérant le site du CCK, cette étape consiste à supprimer :
  - Les requêtes pour les pages « proxy.pac »,
  - Les requêtes pour les annonces (les popups). En effet, les annonces apparaissent toutes seules dès que l'utilisateur se connecte sur le site du CCK. De ce fait, les requêtes correspondantes ne reflètent pas son comportement.

2.1.2 Transformation des fichiers logs

Identification des utilisateurs et des sessions : Une session est composée de l'ensemble de pages visitées par le même utilisateur durant la période d'analyse. Afin d'identifier les sessions, nous considérons que deux requêtes provenant de la même adresse IP mais de deux user-agents différents appartiennent à deux sessions différentes donc elles sont effectuées par deux utilisateurs différents. Ainsi, le couple (IP, User-Agent) représente un identifiant des utilisateurs. Toutefois, nous ne pouvons nier la limite inhérente à cette méthode. En effet, une confusion entre deux utilisateurs différents utilisant la même adresse IP et le même User-Agent est toujours possible surtout en cas d'utilisation d'un serveur Proxy ou d'un firewall.

```
Tant qu'il y'a des enregistrements dans la base faire
   Lire l'enregistrement i
   Récupérer l'adresse IP_i et le User Agent UA_i
      Si le couple (IP_i, UA_i) = (IP_(i-1), UA_(i-1))
         Alors ajouter l'enregistrement i à la session S_(i-1)
         Sinon recommencer une nouvelle session S_i
      Fin Si
Fin Tant Que
```

FIG. 2 – *Algorithme d'identification des utilisateurs et des sessions.*

Identification des visites : Une visite est composée d'une suite de requêtes séquentiellement ordonnées, effectuées pendant la même session et ne présentant pas de rupture de séquence de plus de 30 minutes (d'après les critères empiriques de Kimball (2000)). L'identification des visites sur le site, est effectuée en suivant cette démarche :

1. Déterminer la durée de consultation des pages. La durée de consultation d'une page est le temps séparant deux requêtes successives. Si la durée de consultation d'une page dépasse 30 minutes alors la page suivante dans la même session est attribuée à une nouvelle visite.
2. Une fois les visites identifiées, la durée de consultation de la dernière page de chaque visite est obtenue à partir de la moyenne des temps de consultation des pages précédentes appartenant à la même visite.

2.1.3 Résultats de l'analyse des fichiers Log du CCK

Le tableau suivant présente les résultats du prétraitement des fichiers Logs du site du Centre de Calcul elKhawarizmi[4] collectées pendant la période allant du 17 Septembre au 14 Octobre 2004.

	Nombre de requêtes	**Pourcentage**
Total de requêtes	**279879**	**100 %**
Requêtes non valides	*13028*	4.6 %
Requêtes provenant des WRs	*7187*	2.7 %
Identification par IP ou UA	2651	
Requêtes à « /robots.txt »	0	
Identification par mots clés	3689	
Requêtes effectuées par des aspirateurs	847	
Identification par BS	0	
Requêtes aux images et fichiers multimédia	*144025*	55.4 %
Requêtes dont méthode<>GET	*158*	0.13 %
Scripts et feuilles de style	*7426*	6.4 %
Requêtes spécifiques au site du CCK	*83163*	76.9 %
Requêtes à « /proxy.pac »	*78439*	
Annonces	4724	
Total	**254987**	**91.1 %**
Nombre de requêtes après nettoyage et retraitement	35353	3.7 %
Nombre des sessions	1770	
Nombre des visites destinées à l'analyse	2700	

TAB. 1 - *Tableau récapitulatif des résultats.*

2.1.4 Création de nouvelles variables

A partir des variables préexistantes, des nouvelles variables sont crées pour faciliter l'analyse envisagée. D'autres variables peuvent être créées suivant la nature de l'analyse envisagée.

Variable Originale	**Variables créées**	**Type**	**Valeurs variables**
Time	Période-journée	Discrète	Matin, Midi, Après midi, Soir, Nuit
Statut	Statut-200	Discrète	1, 0 (1 si le statut est 200, 0 sinon)
URL	Extension	Discrète	PDF, html, rtf, asp, doc, dot, jpg, ppt,
	Niveau 1	Discrète	
	Niveau 2	Discrète	
	⋮		
User-Agent	Navigateur	Discrète	MSIE, Netscape, Autres navigateurs
	Plateforme	Discrète	Windows, Unix/Linux, MacOS

TAB. 2 - *Création de nouvelles variables.*

[4] www.cck.rnu.tn

URL	Extension	Niveau 1	Niveau 2
/français/espace_chercheur.htm	htm	français	espace_chercheur

TAB. 3 - *Transformation de la variable URL.*

2.2 Classification des pages

La classification des pages a pour objectif de distinguer les pages de contenu présentant l'information recherchée par l'internaute des pages de navigation utilisée pour faciliter la navigation de l'utilisateur sur le site de manière à ne garder dans la base que les requêtes aux pages présentant un contenu intéressant aux visiteurs. Notre approche consiste à définir des variables servant à la caractérisation des pages et les utiliser pour la classification des pages.

2.2.1 Collecte des informations sur les accès

Afin de caractériser les pages visitées par les internautes, les variables suivantes sont définies pour chaque page :

- Nombre de Visites (NV) effectuées à chaque page ;
- Nombre des Inlinks (NI) : nombre d'hyperliens qui mènent à la page en question à partir des autres pages ;
- Nombre des Outlinks (NO) : nombre d'hyperliens dans la page qui mènent vers d'autres pages ;
- Durée Moyenne par page : temps moyen de visite de chaque page (DM) ;
- Taille du Fichier (TF) ;
- Type du Fichier (.html, .doc, .pdf, .rtf, …etc) (TYF).

Ainsi, chaque page peut être représentée par un vecteur :

Page = {NV, NI, NO, DM, TF, TYF}

Les variables TF et DM sont obtenues directement à partir de la base des requêtes après nettoyage et transformation. Pour la variable TYF, nous supposons que les pages dont l'extension est «.doc, .pdf, .rtf…» sont des pages de contenu. Par conséquent, nous ne considérons dans la suite que les pages «.asp» et «.html» auxquelles nous appliquons l'ACP.

La détermination des variables NV, NI et NO nécessite tout d'abord l'indexation des pages du site Web pour faciliter leur manipulation et la construction de deux matrices : matrice d'hyperliens et matrice d'accès.

URL	Referrer
/arabe/index.htm	/haut.htm
/arabe/main.htm	/arabe/index.htm
/arabe/espace_chercheur/cher_ar.htm	/arabe/main.htm
/arabe/services_en_ligne/serv_ligne_ar.htm	/arabe/espace_chercheur/cher_ar.htm
/arabe/haut.htm	/arabe/index.htm
/arabe/services_internet/serv_internet_ar.htm	/arabe/haut.htm

TAB. 4 - *Exemple de visite.*

URL	Index
/haut.htm	A
/arabe/index.htm	B
/arabe/main.htm	C
/arabe/haut.htm	D
/arabe/espace_chercheur/cher_ar.htm	E
/arabe/services_en_ligne/serv_ligne_ar.htm	F
/arabe/services_internet/serv_internet_ar.htm	G

TAB. 5 - *Indexation des pages de la visite.*

Matrice d'accès. Cette matrice est utilisée pour déterminer le nombre de visites effectuées par les internautes à chaque page. Chaque entrée (i,j) de la matrice représente le nombre de visites effectuées de la page i à la page j. Si cette entrée est égale à zéro alors la page j n'a jamais été visitée à partir de la page i.

	A	B	C	D	E	F	G	**Total**
A	0	0	0	0	0	0	0	**0**
B	5	0	0	0	0	0	0	**5**
C	0	24	0	0	0	0	0	**24**
D	0	9	0	0	0	0	0	**9**
E	0	0	2	0	0	0	0	**2**
F	0	0	0	0	1	0	0	**1**
G	0	0	0	1	0	0	0	1

TAB. 6 - *Matrice d'accès.*

Matrice d'hyperliens. Cette matrice est utilisée pour calculer le nombre d'inlinks et le nombre d'outlinks. En effet, le nombre d'inlinks est le total sur les lignes alors que le nombre d'outlinks est le total sur les colonnes.

	A	B	C	D	E	F	G	**Inlinks**
A	0	0	0	0	0	0	0	**0**
B	1	0	0	0	0	0	0	**1**
C	0	1	0	0	0	0	0	**1**
D	0	1	0	0	0	0	0	**1**
E	0	0	1	0	0	0	0	**1**
F	0	0	0	0	1	0	0	**1**
G	0	0	0	1	0	0	0	**1**
Outlinks	**1**	**2**	**1**	**1**	**1**	**0**	**0**	**6**

TAB. 7 - *Matrice d'hyperliens.*

Chaque ligne de la matrice correspond à une page du site. Il en est de même pour chaque colonne. Ainsi, s'il existe N pages différentes visitées par les internautes, la matrice d'hyperliens sera de dimension (N, N). Chaque entrée (i,j) de la matrice prend la valeur 1 si l'utilisateur a visité la page j à partir de la page i (présence d'un lien direct entre les deux pages) et la valeur 0 sinon. Toutefois, il ne faut pas oublier que certaines pages du site ne sont pas visitées par les internautes et que certains liens dans les pages visitées ne sont pas utilisés. Ces pages et hyperliens ne sont pas considérés dans cette représentation matricielle qui ne prend que les accès enregistrés dans les fichiers Logs.

2.2.2 Application de l'analyse en composantes principales

En considérant les variables présentées ci-dessus, nous avons appliqué l'ACP au tableau (pages × variables).

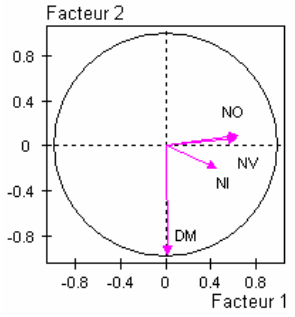


FIG. 4 - *Projection des variables sur les axes factoriels.*

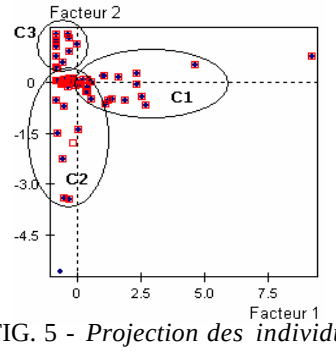


FIG. 5 - *Projection des individus sur les axes factoriels.*

D'après Fig.10, le premier axe factoriel est expliqué par les trois variables NV, NI et NO. Il oppose les pages les plus fréquentées et ayant un nombre important d'inlinks et d'outlinks aux pages les moins fréquentées et caractérisées par un faible nombre d'inlinks et d'outlinks. Le second axe factoriel est celui de la durée moyenne de consultation de pages. Les pages projetées définissent quatre classes.

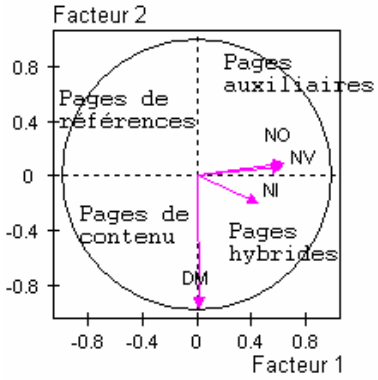


FIG. 6 - *Classification des pages.*

Type de page	NV	NI	NO	DM
Page de contenu	-	-	-	+
Page auxiliaire	+	+	+	-
Page hybride	+	+	+	+
Page de référence	-	-	-	-

TAB. 8 - *Caractérisation des pages.*

La première classe (C1) est composée de pages visitées fréquemment et caractérisées par un nombre important d'inlinks et d'outlinks. Elle correspond à la classe de pages auxiliaires ou de navigation. La deuxième classe (C2) est celle de pages de contenu caractérisées par une durée de consultation assez élevée. L'intersection de ces deux classes est composée de pages présentant à la fois les caractéristiques des pages de contenu et des pages auxiliaires. C'est la classe de pages hybrides (C4). La dernière classe (C3) est celle des pages visitées rarement, qui ne pointent nulle part et vers lesquelles pointent peu de pages. La durée moyenne de consultation de ces pages est faible. Nous considérons que ces pages correspondent à ce que Rao (1996) appelle « pages de références » utilisées pour définir un concept ou expliquer des acronymes. Cependant, nous considérons que ces pages sont, dans une certaine mesure, des pages de contenu.

2.3 Classification des utilisateurs

Cette phase est réalisée en trois étapes. La première consiste à classifier les requêtes effectuées par les internautes afin de découvrir des motifs de navigation. Les résultats de cette première classification sont injectés dans la base des visites. La clusterisation des visites permet de construire des groupes d'utilisateurs.

2.3.1 Découverte de motifs de navigation

La découverte de motifs de navigation est effectuée à deux niveaux en combinant deux méthodes de classification. La première est l'analyse des correspondances multiples appliquées aux variables présentées dans TAB. 1. La seconde est la carte topologique de Kohonen utilisée pour déterminer des groupes de requêtes. Les axes factoriels résultant de l'application de l'ACM servent de variables d'entrée (inputs) pour la carte de Kohonen.

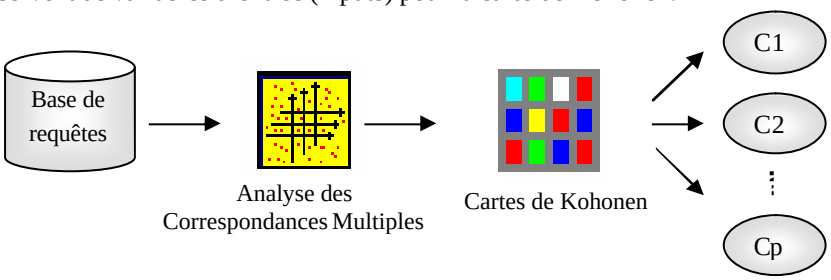


FIG. 7 - *Etapes de la classification des requêtes.*

La caractérisation des classes obtenues par la projection des variables « niveau 1» et « niveau 2 » déterminées à partir de la variable « URL » a donné le résultat présenté par la figure 14. En examinant les éléments caractérisant chaque classe, il est possible d'attribuer un label à chaque classe. La classe 1-1 par exemple comporte des requêtes aux institutions universitaires, la classe 3-4 comporte des requêtes dont l'objectif est le téléchargement des cours. la carte de Kohonen, après division en aires logiques et labellisation, met en évidence cinq aires logiques correspondant à cinq motifs de navigation : visites aux institutions universitaires, activités de recherche, demandes des informations sur les congrès, services CCK et téléchargement des cours.

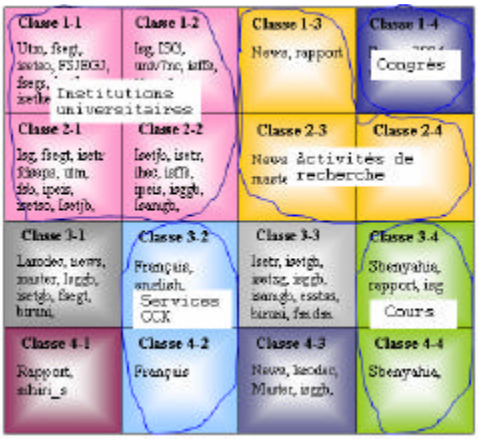


FIG. 9 - *Carte de Kohonen après division en aires logiques et labellisation.*

2.3.2 Construction de groupes d'utilisateurs

Pour construire des groupes d'utilisateurs, il faut tout d'abord attribuer à chaque visite un ou plusieurs motifs de navigation, caractériser les visites par un ensemble de variables et les regrouper en classes en suivant le schéma suivant. La seconde étape consiste à construire des groupes d'utilisateurs et les caractériser.

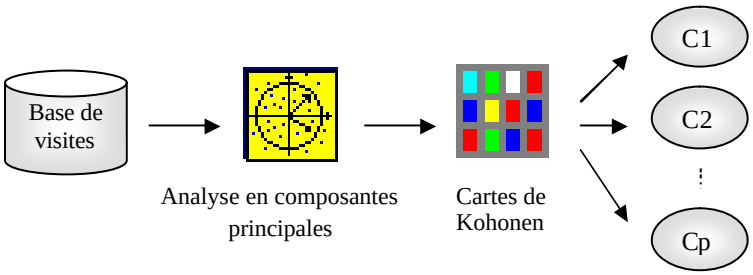


FIG. 10 - *Etapes de la clusterisation des visites*

Analyse en composantes principales : L'analyse en composantes principales s'applique à un tableau (individus x variables). Dans notre cas, les individus sont les visites et les variables considérées sont présentées dans Tab.5.

Nom de la variable	Description de la variable
Durée_visite	La durée de chaque visite
DuréeMoyPage	Durée moyenne de consultation des pages dans chaque visite
NbReqVisite	Nombre de requêtes dans chaque visite
PourcReqDiff	Pourcentage de requêtes différentes dans chaque visite
PourcReqOk	Pourcentage de requêtes réussies dans chaque visite

TAB.9 - *Variables utilisées dans l'ACP.*

Cartes de Kohonen : L'application des cartes de Kohonen met en évidence trois classes de visites. La première classe est composée de visites dont la durée moyenne, le nombre moyen de requêtes par visite et la durée moyenne de consultation des pages sont assez élevées en comparaison avec les deux autres classes. Ceci s'explique par le fait que ces visites sont effectuées principalement dans le but de télécharger des cours ou visiter des institutions universitaires. La deuxième classe est caractérisée par le pourcentage le plus élevé de requêtes réussies (95%) et de requêtes différentes (98%). Ces visites sont effectuées afin de profiter des services fournis par le CCK tels que les services Internet, les services de calcul et le compte Internet. La dernière classe comporte des visites dont l'objectif est d'avoir des informations sur les congrès, les colloques, ..etc.

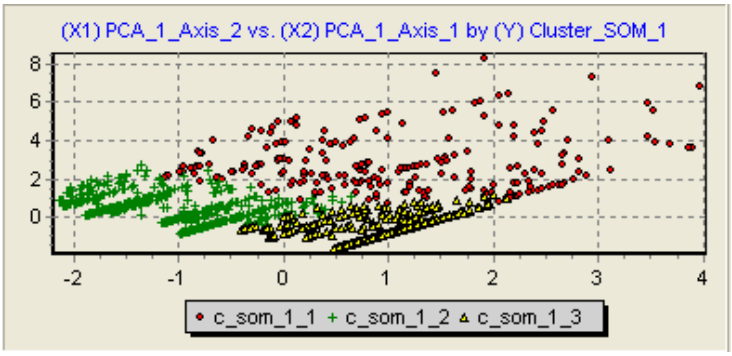


FIG. 11 - *Résultat de la classification des visites.*

Pour chaque groupe de visites, un groupe d'utilisateurs est construit contenant tous les utilisateurs possédant au moins une visite dans ce groupe de visites. Ainsi, nous obtenons des groupes d'utilisateurs ayant le même motif de navigation. Le premier groupe est celui des universitaires dont l'objectif de la navigation sur le site et le téléchargement des cours, l'inscription dans les établissements universitaires et la visite des bibliothèques universitaires. Le deuxième groupe est celui des chercheurs qui demandent des informations sur les congrès, les colloques, les mastères et les thèses et visitent les laboratoires de recherche. Le troisième groupe est celui des visiteurs du site du CCK afin de profiter des services qu'il fournit. Un dernier groupe est déjà défini lors du prétraitement des fichiers Logs et dont les requêtes ont été supprimées pendant la phase du prétraitement. Il s'agit des agents et robots utilisés par les moteurs de recherche pour mettre à jour leurs index de recherche.

3 Conclusion

Dans ce travail, nous avons développé une méthodologie de prétraitement des fichiers Logs permettant de transformer l'ensemble de requêtes enregistrées dans les fichiers Logs à des données structurées et exploitables. L'hybridation des méthodes de classification nous a permis de surmonter l'obstacle de la quantité des données et de tirer profit du pouvoir classificatoire de certaines d'entre elles, à savoir les cartes topologiques de Kohonen.

Références

Charrad, M. (2005) *Techniques d'extraction des connaissances appliquées aux données du Web*. Mémoire de Mastère présenté en vue de l'obtention du diplôme de Mastère en Informatique, Ecole Nationale des Sciences de l'Informatique de Tunis, Laboratoire RIADI.

Charrad, M., M. Ben Ahmed et Y. Lechevallier (2005). Web Usage Mining : WWW pages classification from log files. *In Proceeding of International Conference on Machine Intelligence*, Tozeur, Tunisia, 5-7 Novembre.

Cooley, R., B. Mobasher, et J. Srivastava (1999). Data Preparation for Mining World Wide Web Browsing Patterns. *Journal of Knowledge and Information Systems*.

Kimball, R. et R. Merz (2000). *Le data webhouse. Analyser des comportements clients sur le Web*. Editions Eyrolles, Paris.

Lechevallier, Y., D. Tonasa, B. Trousse, R. Verde (2003). Classification automatique : Applications au Web Mining. *In Proceeding of SFC2003, Neuchatel*, 10-12 Septembre.

Mobasher, B., H. Dai, T. Lou, et M. Nakagawa (2002). Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6: 61-82.

Pierrakos, D., G. Paliouras, C. Papatheodorou, et C.D. Spyropoulos (2003). Web Usage Mining as a tool for personalization: A survey. *User Modeling and User-Adapted Interaction*, 13:311-372.

Rao, R., P. Pirolli, J.Pitkow (1996). Silk from a sow's ear : Extracting usable structures from the web. *In proc. ACM Conf. Human Factors in Computing Systems, CHI.*

Srivastava, J., R. Cooley, M. Deshpande et P.-N. Tan (2000). Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKKD Explorations.*

Tanasa, D. et B. Trousse (2003), Le prétraitement des fichiers Logs Web dans le Web Usage Mining Multi-sites. *In Journées Francophones de la Toile*.

Summary

The approach we proposed to characterize users of a Web site consists of three steps: preprocessing of log files, Web site pages classification and users clustering. In Preprocessing, requests are processed to be organized into sessions. In page classification, parameters are introduced from pages access statistics to help classify pages into auxiliary pages and content pages. Requests to content pages are used to discover browsing patterns. Two hybrid clustering methods based on Principle Component Analysis, Multiple Correspondences Analysis and Self Organizing maps are applied to sessions to construct users' groups. An experiment on real log files shows that the approach is efficient and practical.

Fouille du Web pour la collecte de données linguistiques : avantages et inconvénients d'un corpus hors-normes

Florence Duclaye* , Olivier Collin*, Emmanuelle Pétrier**

*France Télécom R&D
2, avenue Pierre Marzin
22307 Lannion Cedex
{florence.duclaye,olivier.collin}@francetelecom.com
**Teamlog
rue Fulgence Bienvenue
22304 Lannion
epe@teamlog.com

Résumé. De plus en plus utilisé pour l'acquisition de ressources linguistiques, le Web ne répond pas aux critères habituels des corpus "traditionnels". L'abondance, la redondance des informations qu'il contient et son multilinguisme intrinsèques sont des atouts utiles à des fins d'acquisition de données linguistiques. En revanche, son hétérogénéité et le manque de stabilité et de fiabilité des informations rendent les données collectées parfois fortement bruitées. Cet article propose une analyse du Web et de ses caractéristiques sous l'angle de l'acquisition de ressources linguistiques. Les avantages et limites liés à la fouille du Web sont présentés puis illustrés par quelques expériences.

1 Introduction

En parallèle des travaux qui exploitent le Web pour l'acquisition de données linguistiques, de nombreuses études reposent sur des corpus fermés, constitués pour répondre à des besoins spécifiques. Ces corpus peuvent être généralistes ou spécialisés, bruts ou pré-traités, mono- ou multi-lingues, et de taille très variable. Construire un corpus qui réponde à des besoins précis en matière d'apprentissage est une étape certes déterminante pour la réussite de l'apprentissage, mais très longue et consommatrice de ressources humaines, et qui aboutit à des corpus de taille limitée. C'est pourquoi le Web est devenu une ressource privilégiée, utilisée depuis une dizaine d'années pour en extraire, avec un succès parfois très variable, tous types de contenus. Néanmoins, pour que sa fouille soit efficace, le Web doit être considéré en fonction de l'usage que l'on souhaite en faire et de ses caractéristiques. En effet, s'il ne répond pas aux définitions standard des corpus (Rundell (2000)), il ne répond pas non plus exactement aux caractéristiques habituelles des corpus. Ensemble de données brut, ouvert et multilingue, le Web est à la fois d'un corpus généraliste et spécialisé, dans la mesure où l'on y trouve tous types de documents. Simple regroupement de textes, il n'a pas été organisé en fonction de critères précis et est devenu un « entrepôt chaotique » de productions de documents du monde entier, comme le montre Lynch (1997). Dans cet article, nous analysons le Web et ses caractéristiques sous

l'angle de la collecte de données linguistiques. Dans les parties 2 et 3, nous argumentons sur les intérêts et limites liés à l'utilisation du Web comme support d'apprentissage puis comme support linguistique. Nous présentons ensuite, en partie 4, les travaux de fouille de Web que nous avons menés et qui exploitent les caractéristiques ainsi mises en évidence.

2 Le Web comme corpus d'apprentissage

2.1 La démesure du Web

Estimer la taille du Web est une tâche d'autant plus difficile que le Web se décompose en deux sous-ensembles, le Web visible, et le Web invisible (encore appelé Web caché ou Web profond), inaccessible aux moteurs de recherche classiques. Le web invisible comprend des bases, des banques de données ou encore des bibliothèques en ligne. Il est quasiment impossible d'estimer raisonnablement son étendue. Les études se concentrent majoritairement sur la taille du Web visible, faisant elle-même l'objet d'estimations parfois très divergentes. Début 2005, une étude menée par Gulli et Signorini (2005) estimait le nombre de pages sur le Web indexable à 11.5 milliards. En octobre 2005, Netcraft estimait le nombre de sites à un peu moins de 75 millions [1]. Ces chiffres, probablement inexacts, permettent néanmoins d'en apprécier la démesure. Des corpus comme le Brown Corpus ou le British National Corpus, pourtant de taille très importante (1 million de mots pour le premier, 100 millions pour le second) ne peuvent égaler la couverture du Web. Cette démesure constitue un atout primodial pour l'apprentissage statistique de données linguistiques, dont la qualité est fonction de la quantité d'informations à disposition. Nos travaux en classification automatique et en validation d'attachements syntaxiques exploitent cette propriété du Web (voir partie 4.1).

2.2 Redondance des informations

La redondance des données sur le Web prend plusieurs formes. D'une part, Zhu et Rosenfeld (2001) notent que, dans les résultats proposés par les moteurs de recherche, de multiples occurrences d'un même document peuvent apparaître. Un même document peut en effet parfois avoir plusieurs URLs différentes. Cette redondance ne rend pas compte du poids de l'information mais biaise au contraire les résultats obtenus sur les données acquises à partir de méthodes statistiques.

Au-delà de ces difficultés, la redondance concerne aussi les informations elles-mêmes, déclinées sous des formes linguistiques diverses et variées. On trouvera sur le Web quantité d'expressions de même sens, utilisant des unités lexicales et des constructions syntaxiques différentes. Les informations journalistiques, par exemple, se propagent très rapidement, en raison de la concurrence des sites qui les fournissent. Les travaux en apprentissage statistique de données à partir du Web exploitent cette redondance. Plus l'on dispose de données, meilleures sont les données apprises. Le Web constitue à la fois une masse gigantesque de données qui, elle-mêmes, y sont redondantes. Cela en fait un très intéressant support d'apprentissage. Les travaux que nous avons menés en acquisition de concepts bilingues en sont un exemple d'illustration (voir partie 4.4).

[1] `http://news.netcraft.com/archives/web_server_survey.html`

2.3 Un corpus hétérogène

En plus de pouvoir accéder au Web par des services de recherche d'informations comme les moteurs de recherche, on peut y réserver un train, acheter des livres, chercher une adresse... Le Web apparaît comme une immense base textuelle et multimédia où l'on trouve tous types de données, sous tous types de formes et de formats, appartenant à tous les domaines de la connaissance. Dans cette masse hétérogène de documents, les données sont dispersées. Brin (1998) indique par exemple qu'un type de données particulier comme les listes de restaurants peut être éparpillé dans des milliers de sources d'information indépendantes. Les informations revêtent également diverses formes linguistiques, dans une multitude de langues et de styles langagiers (voir partie 3.1). Les données se trouvent sous des formes structurées très diverses, mais aussi des formes semi-structurées ou non structurées (XML, texte libre...). Face à une telle hétérogénéité de contenus et de formes, les traitements automatiques effectués sur le Web perdent en fiabilité. Habert (2000) confirme ce risque de perte de qualité des études s'appuyant sur le Web : « plusieurs études convergent [...] pour rendre plausible l'hypothèse selon laquelle la fiabilité des traitements automatiques dépendrait de l'homogénéité des données en cause ». Lors de l'évaluation d'étiqueteurs morphosyntaxiques GRACE, Illouz (1999) met en évidence la corrélation entre la plus ou moins grande précision des étiqueteurs et la nature des textes à catégoriser (romans, essais ou mémoires). Ce constat s'applique donc a fortiori sur le Web. L'hétérogénéité du Web peut pénaliser l'apprentissage.

2.4 Un corpus en évolution permanente

En permanence sur le Web, des pages sont créées, certaines modifiées ou déplacées, d'autres supprimées. Il s'agit d'un corpus en perpétuel changement. Sundaresan et Yi (2000) expliquent que "le Web est un ensemble de pages en croissance permanente, dont les auteurs ont des cultures, des intérêts et des niveaux d'éducation très différents [...]". Quoique difficilement quantifiable, la vitesse de croissance du Web est exponentielle. La croissance des débuts fut fulgurante : 623 sites Web étaient dénombrés en fin 1993, 10.022 en fin 1994, 100.000 en fin 1995, 650.000 en décembre 1996... En octobre 2005, Netcraft comptabilisait environ 75 millions de sites (voir partie 2.1) et estimait que 2005 est l'année de la plus forte croissance du Web depuis ses débuts (17 millions de nouveaux sites, contre 16 millions en 2000). Cette dynamicité permet d'accéder à des données constamment mises à jour.

Les moteurs de recherche ne peuvent suivre cette évolution sans un certain délai. En plus de renvoyer des documents qui n'existent parfois plus, ils renvoient quelquefois des résultats différents à quelques heures d'intervalle seulement. La dynamicité du Web peut rendre les évaluations statistiques sur le Web très inexactes. Certains travaux visent à aspirer le Web pour en obtenir une image valable à un moment donné précis. Mais le Web est trop grand pour pouvoir être aspiré dans un intervalle de temps acceptable (Kahle (1997)). Une stratégie possible pour pallier à cette dynamicité consiste à aspirer uniquement les sites sur lesquels porte une étude, lorsque cela est possible, comme le font Beaudouin et al. (2001).

Langue	Nb de locuteurs*	Dont langue maternelle*	langue officielle : nb d'états	% des internautes**	% de pages Web***
Chinois	1.300	885	3	10.8	3.9
Anglais	1.000	341	45	36.5	68.4
Espagnol	450	332	20	7.2	2.4
Bengali	250	207	nd	nd	nd
Arabe	250	200	25	0.9	0.04
Hindi/Ourdou	900	190	2	nd	nd
Portugais	200	176	7	3	1.4
Russe	320	170	3	2.9	1.9
Malais/Indonésien	160	150	4	1.3	nd
Japonais	130	125	1	9.7	5.9
Allemand	125	100	5	6.6	5.8
Français	125	77	30	3.5	3

TAB. 1 – *Le poids des langues sur le Web en 2003 (chiffrages en millions)*
nd : chiffre non disponible - * millions - ** sur total mondial d'internautes
** sur total mondial de pages

3 Le Web comme corpus linguistique

3.1 Richesse et diversité linguistique

La prédominance de l'anglais sur toutes les autres langues est incontestable sur le Web : en 2000, Rundell (2000) indiquait qu'environ 70% des sites étaient en anglais. Les 30% de sites restants couvrent à eux-seuls 14 autres langues. Nunberg (1998) tire les observations suivantes de ce phénomène : les pays où Internet a peu pénétré et dont la langue est peu parlée en dehors de leurs frontières utilisent majoritairement l'anglais sur leurs sites Web (ex : la Bulgarie : 86% des sites en anglais, la Chine : 82%, l'Égypte : 95%, la Grèce : 81%). Inversement, l'anglais est minoritaire sur les sites des pays non-anglophones où Internet est déjà bien implanté. C'est le cas de la France, du Japon ou de l'Allemagne par exemple. Le tableau 1 indique les pourcentages de sites par langue sur le Web en 2003, et les compare à la quantité de locuteurs de la langue et à celle d'internautes dans le monde (d'après Aït Hamadouche (2003)).

Au sein d'une même langue, on trouve des variantes géographiques, également largement disponibles sur le Web. Les pages en français non-hexagonal sont nombreuses : français du Québec, de Suisse, de Belgique, d'Afrique ou du Moyen-Orient. Alors que les québécois ne représentent que 5% de tous les locuteurs du français, 30% des pages en français étaient canadiennes en 1997 (Oudet (1997)).

Le Web couvre donc un large éventail de langues et de variantes linguistiques. Un corpus de référence comme le corpus PAROLE [2], riche d'environ 30 millions de mots, est loin de pouvoir offrir tant de diversité linguistique. Les travaux présentés en partie 4.4 tirent profit du multilinguisme du Web pour apprendre automatiquement des expressions sémantiquement équivalentes en français et en anglais.

[2] `http://www.elda.fr/catalogue/fr/text/doc/parole.html`

À la diversité sur le plan des langues s'ajoute la diversité sur celui des styles langagiers présents sur le Web. Tous les styles y sont présents (soutenu, familier...), et l'évolution des langues y est aussi reflétée, ce qui laisse le champ libre à l'étude des équivalences entre expressions de styles différents (ex. « acheter » et « avaler » dans un contexte d'achat d'une entreprise par une autre). On assiste même à la naissance d'un style propre au Web, au travers du chat et des forums de discussion (Beaudouin et al. (2001)). Alors que le Web contient essentiellement des données de la langue écrite, on trouve sur les forums et les blogs un style qui se rapproche de la langue orale. Cette diversité contribue en grande partie à faire du Web un corpus d'une grande richesse.Les genres sont également nombreux sur le Web (ex : français littéraire, journalistique, scientifique...). Malheureusement, les moteurs de recherche actuels ne permettent pas de faire des requêtes linguistiques « évoluées ». De nouveaux projets démarrent pour tenter d'exploiter la diversité linguistique présente sur le Web. Kilgarriff (2003) expose par exemple un projet de moteur de recherche linguistique sur le Web.

3.2 Contextualité de l'information linguistique

La richesse et la diversité des documents présents sur le Web permet l'étude des termes et expressions dans leur contexte d'utilisation, que J. Weil appelle « co-texte » dans Weil (2001) : « even if we have only a very rough idea of what is in the whole collection, any individual example (phrase, collocation...) can be studied in its full co-text » [3]. Dans un contexte boursier, la nourriture ne sera pas mangée, mais plutôt échangée, vendue, achetée, etc. Les termes apparaissent dans différents contextes en fonction du domaine en question. Les concordanciers se multiplient depuis quelques années sur le Web, fournissant des fenêtres contextuelles de plusieurs termes à partir des documents. Ces fenêtres contextuelles sont recueillies sur les documents renvoyés par des moteurs de recherche. Citons le projet GoogleLing dont l'objectif est la conception d'un logiciel de recherche linguistique dans les bases documentaires d'un moteur de recherche (en l'occurrence GOOGLE) (Smarr et Grow (2002)). L'utilisation du Web apporte une grande souplesse aux études contextuelles sur les mots. Non seulement il est possible d'y rechercher les fenêtres contextuelles de n'importe quel terme ou expression, mais il est aussi possible d'étendre la notion de contexte aux phrases entières, aux paragraphes et même aux documents entiers où apparaissent ces termes. Les travaux en classification de noms propres présentés en partie 4.3 exploitent la contextualité des informations du Web.

3.3 Fiabilité linguistique des informations

En partie 2.3, nous évoquions l'impossibilité technique de caractériser les pages Web sur le plan du genre, de la validité du contenu informationnel, etc. Dès l'automatisation du traitement de documents inconnus, on se heurte au problème de la fiabilité des documents et informations exploités sur le Web. Tanguy et Hathout (2002) parlent de corpus « incontrôlable », où chacun peut publier ce qu'il veut. La question de la fiabilité se pose non seulement sur les information factuelles, qui divergent parfois sur un même sujet, mais surtout sur les données linguistiques. De qualité linguistique très variable, les pages Web ne sont pas toujours rédigées par des natifs de la langue. Toutes sortes d'erreurs de syntaxe et de vocabulaire surviennent. À cela s'ajoutent

[3] Même si nous n'avons qu'une vague idée de ce qui se trouve dans la base entière, il est possible d'étudier tout exemple isolé (proposition, collocation...) dans son « co-texte » complet.

les erreurs d'analyse d'origines diverses (fautes de frappe et d'orthographe, segments de textes en langue étrangère, noms propres, adresses de courrier électronique, segments d'URLs, noms de variables ou de fonctions des langages de scripts, noms de fichiers, extraits de code informatique...).De plus, la frontière entre expression correcte et expression incorrecte est plus floue sur le Web. Hathout et Tanguy (2002) mettent en évidence le phénomène linguistique des "web-néologismes" et note que les nouveaux mots sont d'une grande variété, en particulier du point de vue des domaines et des sous-lexiques auxquels ils appartiennent : « nouveaux concepts passés dans le langage courant (pacsage, surencadrement), termes techniques (hémagglutination, antialiassage) ou créations idiolectales (capellitractage, choucroutage) ». On trouve également des formes ayant des préfixes nouveaux comme les suffixes euro- (europétition), cyber- (cyberespace, cyberpublication), vapo- (vapogazéification) ou web- (webpromotion). Le fait que le Web suive l'évolution des langues est un avantage incontestable sur les dictionnaires et les bases documentaires fermées. Certainement plus fiables du point de vue lexical et grammatical, le British National Corpus et le Penn Treebank ne pourraient rendre compte de la créativité linguistique du Web. La fiabilité des informations linguistiques traitées sur le Web reste néanmoins invérifiable. Nos travaux sur la validation d'attachements syntaxiques considèrent néanmoins le Web comme une référence en matière d'emploi linguistique 4.1. D'autres travaux tels que ceux de Léon et Millon (2005) illustrent de manière analogue l'utilisation du Web pour l'acquisition de traductions de groupes nominaux.

4 Exploitation du Web et de ses caractéristiques pour l'acquisition de données linguistiques : résultats de quelques expériences

Les travaux décrits dans cette partie exploitent une base SQL de stockage d'un grand nombre de cas d'usage de termes sur le Web, issus d'opérations d'acquisition de phrases sur le Web, de filtrage syntaxique, de comptage et de classification. Ces données permettent de produire, pour chaque terme, un ensemble d'attributs linguistiques et d'attributs contextuels qualitatifs et quantitatifs. Les expériences décrites illustrent et exploitent les caractéristiques du Web mises en évidence précédemment, pour l'acquisition de différents types de ressources linguistiques.

4.1 Décision d'attachement de couples de noms communs

Le problème consiste à prendre une décision d'attachement syntaxique entre deux noms communs successifs. Le corpus est un ensemble de phrases anglaises décrivant des images en style télégraphique. Ces descriptions comportent souvent des ambiguïtés d'attachement que notre analyseur syntaxique ne peut résoudre seul, sans un apport de connaissance supplémentaire. Dans le syntagme "bird on a tree branch", il y a attachement, tandis que dans "bird on a tree lake in background", l'attachement syntaxique ne serait pas sémantiquement plausible.

Nous utilisons le Web comme source de connaissance. La quantité de données très importante nous laisse supposer que des noms communs successifs liés syntaxiquement seront fréquents, dans le cas contraire ils seront très peu fréquents. La fréquence d'occurrence des couples de noms communs sur le Web peut donc être une connaissance permettant de décider

Séquences	Occurrences
a police car	1 020 000
a tree trunk	367 000
a church sun	10
a parent mountain	12

TAB. 2 – *Fréquences brutes de couples de noms communs sur le Web*

de l'attachement ou non de ces couples. Pour valider cette hypothèse, nous avons constitué une liste de 340 couples de noms communs issus de notre corpus. Un expert a jugé que 85 d'entre eux pouvaient s'attacher et que les 255 restant ne pouvaient pas s'attacher. Nous avons donc récupéré la fréquence d'apparition de ces couples sur le WEB contraints par un déterminant : "a tree branch", " a tree lake". Le tableau 2 montre des exemples de fréquences brutes obtenues, très élevées en cas d'attachement possible (police car, tree trunk), très faible dans le cas contraire.

Nous avons réalisé un classifieur binaire (attachement oui/non) en fixant arbitrairement, en fonction des fréquences observées, un seuil de 100 occcurences. Ce classifieur "primaire" nous a permis d'obtenir globalement 82% de couples bien classés sur notre corpus, soit globalement 18% d'erreurs.Ces résultats sont très intéressants car ils nous ont permis d'apporter une connaissance supplémentaire à notre analyseur syntaxique en un temps très court. Ils ont aussi remis en cause la connaissance de notre expert : certains attachements ont finalement été validés alors que l'expert les avait jugés non valides dans un premier temps. Des améliorations peuvent être réalisées sur au moins deux points. La fréquence "utilisable" des couples et le classifieur. Certains couples qui ne devraient pas s'attacher peuvent avoir une fréquence relativement élevée, le moteur ne tenant pas compte de la ponctuation : "a parent. Mountain". Notre modèle de classification "primaire" peut être amplement amélioré, mais il exploite d'ors et déjà avec succès la grande quantité de données disponibles sur le Web.

4.2 Classification du genre des prénoms

Nous nous proposons de tester l'étiquetage automatique du genre de 622 prénoms, majoritairement français, comportant 347 prénoms masculins et 275 prénoms féminins. L'hypothèse de départ est que les titres associés à un prénom tels que "Monsieur, M, Mr, Madame, Mademoiselle, Melle" sont des variables pertinentes pour discriminer le genre. Ces variables sont utilisées pour représenter les prénoms et les classer en deux classes : M, F. Le choix des variables n'a pas été validé, nous espérons une validation a posteriori de ce choix. Comme précédemment les fréquences brutes associées à des requêtes particulières sont récupérées sur le Web. Nous supposons que ("Madame Adrien" et "Mademoiselle Adrien" et "Mme Adrien") est beaucoup plus fréquent que ("Monsieur Adrien" et "Mr Adrien" et "M Adrien"). Pour chaque prénom nous réalisons six requêtes, récupérons leur fréquence et en faisons la somme par genre $N(F)$ et $N(M)$. Notre modèle de classification consiste à faire le rapport $N(F)/N(M)$. Le seuil de décision est fixé simplement à 1 : si $N(F)/N(M) > 1$, le prénom est féminin, sinon il est masculin. Le tableau 3 montre une partie des résultats obtenus avec cette méthode. Nous avons ainsi pu réaliser très simplement et rapidement un classifieur du genre des prénoms en 2 classes (masculin, féminin). Nous obtenons, sur notre corpus, 2% d'erreurs de classification.

Prénom	Adrien	Agnès	Barbara	Benoit
N(M)=N(Madame+Mademoiselle +Mme)	152	12477	8103	1015
N(F)= (Monsieur+Mr+m)	12222	362	540	56340
R= N (F) / N(M)	0.012	34.37	14.97	0.018
genre R > 1 : F sinon M	M	F	F	M

TAB. 3 – *Exemples de résultats de classification du genre des prénoms*

La répartition des erreurs n'est pas symétrique, avec environ 4% d'erreurs sur les prénoms féminins et aucune erreur sur les prénoms masculins. Les voies d'amélioration possible sont les suivantes. D'une part, il est nécessaire de diminuer les erreurs de comptage : nous avons observé que Madame peut introduire un nom d'homme (Madame Jacques Chirac) ou que M peut être l'abréviation d'un prénom féminin (M Elizabeth pour Marie Elizabeth). D'autre part, l'utilisation de variables différentes ou complémentaires pourrait améliorer les résultats. Enfin, la prise en compte des prénoms mixtes associé à un autre modèle de classification basé sur une distribution des rapports d'occurrences suivant trois gaussiennes est une piste envisageable.

4.3 Classification d'entités nommées

Dans cette expérimentation nous traitons un problème standard de classification contextuelle des noms propres en trois classes (HUMAIN, LIEU, SOCIETE) par une technique de co-classification, inspirée de celle de Collins et Singer (1999), exploitant la contextualité des informations présentes sur le Web et utilisant peu d'exemples manuellement étiquetés. Nous montrons une utilisation plus complète des différents modules : récupération de phrases centrées sur des termes à partir du Web, extraction de contextes syntaxiquement contraints, comptage et classification probabiliste.

L'initialisation consiste à choisir 20 noms propres de chaque classe. Pour chaque nom propre nous effectuons ensuite une requête sur le Web et nous récupérons en moyenne 1000 phrases contenant chaque nom propre. Nous supposons alors que chacune de ces phrases représente bien l'emploi du nom propre pour sa classe initiale dans ses contextes d'usage (phrases). Nous avons choisi d'extraire les noms communs (NC) immédiatement à gauche des noms propres (NP). Nous conservons les couples NC-NP de fréquence supérieure à 4 et constituons une matrice dont les lignes sont les NC, les colonnes les NP, et les éléments la fréquence d'un couple NC-NP. Connaissant les classes initiales des noms propres d'amorçage, nous calculons les probabilités conditionnelles des noms communs extraits pour chacune des classes telles que $p(humain/president)$, $p(lieu/president)$, $p(societe/president)$. Connaissant la classe du NP prédit par un NC, nous pouvons donc affecter au nom propre le suivant immédiatement à droite une probabilité pour chacune des classes. On peut ensuite conserver la classe prédite la plus probable. Les résultats de la première phase sont encourageants : nous obtenons en un temps court et avec peu de supervision des "prédicteurs" pertinents des classes visées de noms propres. Le tableau 4 affiche (par ordre d'acquisition) la classe la plus probable du nom propre (immédiatement à droite de chacun des noms communs). En regardant les phrases associées on peut constater que "municipalité" est souvent associée à des noms de lieux ou que "client" et "partenaire" sont souvent syntaxiquement liés à des sociétés. Outre une évaluation plus précise, nous pensons améliorer ce travail sur plusieurs points. Tout d'abord distinguer

	Nom commun (NC)	Classe	$p(Classe/NC)$
1	municipalité	L	0.75
2	offre	S	0.9
4	gouvernement	H	0.99
5	équipe	L	0.77
6	livre	H	0.86
7	français	S	0.84
9	voyage	L	0.96
10	parti	H	0.9
14	client	S	0.92
15	partenaire	S	0.90

TAB. 4 – *Prédiction des classes par 10 premiers noms communs (sur 396)*
H : humain, S : société, L : Lieu

les schémas "NC NP" et "NC préposition NP" qui possèdent des pouvoirs prédictifs différents. Puis réaliser une fusion de classifieurs basés sur des variables complémentaires : morphologie du NP, NC postposés , verbe en dépendance avec le NP.

Nous n'avons présenté ici que la première phase du processus puisque nous pouvons ensuite sélectionner les NC possédant une probabilité conditionnelle supérieure à 0.95 pour réitérer l'ensemble du processus et extraire des noms propres. Nous co-classifions ainsi des classes de NP et des classes de NC en exploitant les contextes proches des termes. La quantité de données utilisable sur le Web pour les noms propres initiaux est sans commune mesure avec la quantité utilisable en moyenne sur des corpus standards (Le Monde par exemple). Cette quantité nous permet de suivre une démarche de classification peu supervisée initialisée par un étiquetage des noms propres, démarche naturelle lorsque l'on vise à les classer.

4.4 Acquisition de concepts bilingues français/anglais

Ce travail reprend les travaux de Duclaye et al. (2003) pour en réaliser une extension à des classes de termes bilingues : nous cherchons à identifier des classes de termes français et anglais (synonymes bilingues) qui sont substituables dans des contextes d'emplois très contraints. Les termes visés sont des verbes ou des déverbaux prenant en arguments des couples de noms propres. Le passage du français à l'anglais se fait par l'intermédiaire des noms propres qui possèdent la même forme dans les deux langues, ce qui constitue un type d'alignement "faible". Nous utilisons des patrons syntaxiques, bâtis au-dessus d'un étiqueteur, permettant d'extraire les verbes ou noms à partir des couples de noms propres et inversement des couples de noms propres à partir des verbes ou noms. Le tableau 5 résume les différents cas de figure, en fonction de la catégorie des termes à extraire et de leur position relative par rapport aux noms propres (avant, entre, après). Cet exemple d'utilisation de notre boîte à outils complexifie encore la tâche à réaliser puisqu'il effectue des traitements couplés sur deux langues. Pour ne pas rencontrer simultanément tous les problèmes, nous avons décidé de superviser les différentes étapes d'acquisition, ce qui ne nuit pas à l'intérêt des résultats.

Le processus, visant à acquérir des équivalences du verbe acheter, se déroule de la manière suivante. Un amorçage est réalisé à partir des différentes formes du lemme acheter qui

Extractions verbales	Extractions nominales
AOL achète | acheter Netscape	le rachat de | rachat AOL par Netscape
Après avoir acheté | acheter Netscape, AOL est devenu déficitaire	AOL propriétaire de | propriétaire Netscape
AOL et Netscape négocient | négocier	AOL/Netscape, la rachat | rachat

TAB. 5 – *Six catégories d'extractions syntaxiques utilisées*

sont récupérées dans notre base "experte" et lancées sur le Web. Les patrons syntaxiques permettent d'extraire N couples de noms propres (en général des sociétés) à partir des phrases renvoyées. Ces couples sont ordonnés par leur fréquence. Les dix couples les plus fréquents (validés manuellement) sont conservés et constituent l'amorce. Ces couples permettent à leur tour de récupérer chacun M phrases à partir du Web francophone et du Web anglophone. Des verbes et des noms sont ensuite extraits dans les deux langues au moyen des patrons syntaxiques contraints par les noms propres. Ces termes sont ensuite regroupés par lemmes et ordonnés par leur fréquence. Les plus fréquents sont validés manuellement avant d'être relancés à leur tour sur le Web afin d'extraire de nouveaux couples de noms propres. La figure 1 illustre la première phase de ce processus. Les nouveaux lemmes acquis et sélectionnés suite à

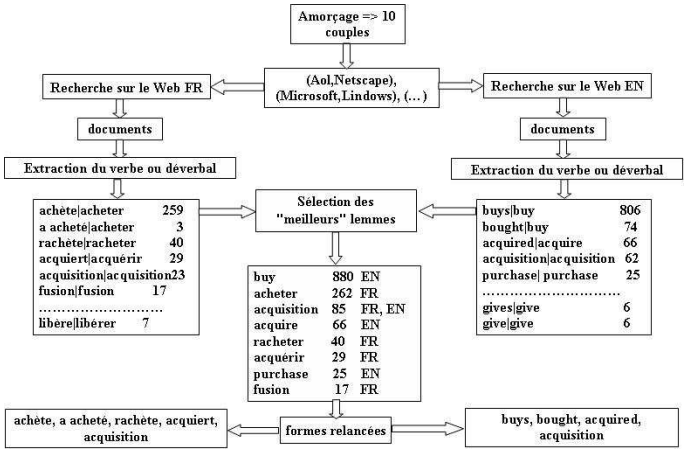


FIG. 1 – *Processus global d'acquisition et de classification de noms propres*

ce processus sont : buy , acquisition, acquire, racheter, acquérir, purchase, fusion.

Les équivalences récupérées sur cette relation exploitent la redondance des informations présentes sur le Web et sont de bonne qualité. Ces résultats sont encourageants par les données obtenues qui permettent d'alimenter nos ressources linguistiques et par le temps de travail nécessaire pour les obtenir : la boîte à outils étant réalisée, l'essentiel du traitement est automatique et réalisé en back office.Nous améliorons actuellement ce processus pour y intégrer un nouveau modèle de classification peu supervisée (voir Clérot et al. (2004)). Nous le testons

sur d'autres relations (possession, mariage, invention) et comptons utiliser plusieurs stratégies en fonction des familles de relations et de la quantité de données récupérées à chaque étape.

5 Conclusion

Par une analyse du Web et de ses caractéristiques, cet article met en lumière les avantages et inconvénients liés à sa fouille pour la collecte de données linguistiques. Corpus hors-normes en de multiples points, le Web est de plus en plus utilisé pour la modélisation de ressources linguistiques. Le traitement de grosses masses de données hétérogènes, les techniques d'apprentissage peu supervisé ainsi qu'une gestion incrémentale du couplage expertise/apprentissage constituent un terrain favorable à cette modélisation. Les expériences d'acquisition de connaissances à partir de corpus décrites dans cet article nous orientent vers une meilleure gestion de nos ressources. Les premières utilisations de notre base de stockage de cas d'usage de termes nous ont permis de mettre rapidement en oeuvre des idées utilisant intensivement le Web comme source de connaissance. Nous avons ainsi montré quelques possibilités d'exploitation des atouts du Web dans un processus global d'apprentissage statistique partiellement supervisé de ressources linguistiques.

Références

Aït Hamadouche, L. (2003). Politiques de la langue - Guerres des langues, enjeux de pouvoir. *Courrier International*, 27–28. Hors-Série "À la découverte des 6700 langues de la planète".

Beaudouin, V., S. Fleury, B. Habert, G. Illouz, C. Licoppe, et M. Pasquier (2001). TyPWeb : décrire la Toile pour mieux comprendre les parcours. In *Actes du Colloque International sur les Usages et les Services de Télécommunications (CIUST'01)*, Paris, France, pp. 492–503.

Brin, S. (1998). Extracting patterns and relations from the world wide web. In *Proceedings of WebDB Workshop at the 6th International Conference on Extending Database Technology*, Valencia, Spain, pp. 172–183.

Clérot, F., O. Collin, O. Cappé, et E. Moulines (2004). Le modèle "monomaniaque" : un modèle simple pour l'analyse exploratoire d'un corpus de textes. In *Actes de la conférence CFTI*.

Collins, M. et Y. Singer (1999). Unsupervised models for named entity classification. In *Proceedings of the EMNLP conference*.

Duclaye, F., F. Yvon, et O. Collin (2003). Unsupervised incremental acquisition of a thematic corpus from the web. In *Proceedings of the NLP-KE conference*.

Gulli, A. et A. Signorini (2005). The Indexable Web is More than 11.5 Billion Pages. *Proceedings of the WWW Conference*.

Habert, B. (2000). *Actes des Journées Scientifiques 1999 "L'imparfait - Philologie électronique et assistance à l'interprétation des textes"*, Chapter Création de dictionnaires sémantiques et typologie des textes, pp. 171–188. Reims, France : Presses Universitaires de Reims.

Hathout, N. et L. Tanguy (2002). Vers un autodétection des webnéologismes. In *TALN, Corpus et Web (TCW'02)*, Paris, France. Résumés des interventions orales.

Illouz, G. (1999). Méta-étiqueteur adaptatif : vers une utilisation pragmatique des ressources linguistiques. In P. Amsili (Ed.), *Actes de la conférence TALN'99*, Cargèse, France, pp. 15–24.

Kahle, B. (1997). Preserving the Internet. *Scientific American 276*(3), 72–73. Special Issue on the Internet.

Kilgarriff, A. (2003). Linguistic search engine. In *Proceedings of the Shallow Processing of Large Corpora Workshop (SproLaC'03) of the Corpus Linguistics Conference*, Lancaster University, UK, pp. 53–58.

Lynch, C. (1997). Searching the Internet. *Scientific American 276*(3), 44–48. Special Issue on the Internet.

Léon, S. et C. Millon (2005). Acquisition semi-automatique de relations lexicales bilingues (français-anglais) à partir du web. In *Actes des Rencontres des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL)*.

Nunberg, G. (1998). Languages in the Wired World. In *Proceedings of the conference on "The Politics of Language and the Building of Modern Nations"*, Paris, France.

Oudet, B. (1997). Multilingualism. *Scientific American 276*(3), 67–68. Special Issue on the Internet.

Rundell, M. (2000). The biggest corpus of all. *Humanising Language Teaching Magazine*.

Smarr, J. et T. Grow (2002). Googleling : The web as a linguistic corpus. Technical report, Stanford University.

Sundaresan, N. et J. Yi (2000). Mining the Web for Relations. In *Proceedings of the 9th International World Wide Web Conference*, Amsterdam, Netherlands, pp. 699–711.

Tanguy, L. et N. Hathout (2002). Webaffix : un outil d'acquisition morphologique dérivationnelle à partir du Web. In *Actes de la conférence TALN*, Nancy, France, pp. 245–254.

Weil, J. (2001). The World Wide Web as a Corpus - How to make use of conventional search engines for linguistic purposes. Homework paper at the University of Berlin - `http://www.user.tu-berlin.de/jawebada/Web_corpus.html`.

Zhu, X. et R. Rosenfeld (2001). Improving Trigram Language Modeling with the World Wide Web. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'01)*, Salt Lake City, USA, pp. 533–536.

Summary

Mining the Web for linguistic resource acquisition is becoming more and more frequent despite its non-standard characteristics compared with "traditional" corpora. The information contained on the Web is intrinsically abundant, redundant and multilingual, which constitutes useful assets for linguistic data acquisition. However, the collected data may sometimes contain much noise due to its heterogeneity, its dynamicity and the irregular reliability of the information. This paper proposes an analysis of the Web and of its characteristics on the point of view of linguistic resource acquisition. The advantages and limitations of mining the Web are argued and then illustrated with some examples of experiments.

Découverte de relations candidates à l'enrichissement d'un entrepôt thématique de données

Hélène Gagliardi*, Ollivier Haemmerlé**
Nathalie Pernelle*, Fatiha Sais*

*LRI (UMR CNRS 8623 - Université Paris-Sud) /INRIA (Futurs),
batiment 490, 91405 Orsay cedex
{prenom.nom@lri.fr}
** Département de Mathématiques-Informatique UFR Sciences, Espaces et Sociétés
Université Toulouse le Mirail
5, Allées Antonio Machado 31058 Toulouse Cedex 1
ollivier.haemmerle@univ-tlse2.fr

Résumé. Ce travail a pour objectif d'enrichir automatiquement un entrepôt thématique de données à partir de documents de format divers provenant du Web et comportant des tableaux. Cet enrichissement est guidé par une ontologie comportant un ensemble de termes et de relations. Le caractère automatique de notre approche nous a conduit à utiliser un format flexible permettant de représenter les données (termes ou relations) dont l'identification est partielle, incertaine ou multiple. Cet article se focalise sur la découverte dans les tableaux de relations non identifiées ainsi que sur la découverte d'attributs qui peuvent enrichir ou modifier l'interprétation de relations identifiées.
Mots-clés : extraction de connaissances, entrepôt, ontologie, XML, Web.

1 Introduction

Notre travail concerne la construction automatique d'entrepôts thématiques. Il s'agit plus précisément d'enrichir un entrepôt constitué de différentes bases de données à l'aide de données trouvées sur le Web.

Ce travail a été réalisé dans le cadre du projet e.dot (e.dot (2004)) (Entrepot de Données Ouvert sur la Toile [1]). Le domaine d'application choisi est le risque microbiologique dans les aliments, domaine qui présente des enjeux majeurs. En effet, les lois sur la sécurité alimentaire étant de plus en plus strictes, la quantité d'analyses effectuées tous les jours par l'industrie alimentaire entraîne des dépenses importantes qui peuvent être limitées si des outils sont développés pour mieux connaître le comportement des microorganismes dans les aliments. Il sera alors possible de prévenir le risque de contamination au lieu de constater les épisodes de crises. C'est l'objectif du projet Sym'Previus, lancé par des institutions gouvernementales, et, c'est dans le cadre de ce projet, que le système MIEL++ (Buche et al. (2004)) a été développé. Ce système permet d'interroger deux bases de données locales contenant des résultats expérimentaux et industriels sur le comportement de germes pathogènes dans des aliments en fonction de

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

différents facteurs tels que le pH ou la température. Ces données sont incomplètes : il y a en effet une infinité d'expériences pouvant être menées. Il est donc important de pouvoir alimenter cet entrepôt avec des données trouvées sur le Web.

Nous nous sommes focalisés sur les données présentées sous forme de tableaux. En effet, il s'agit d'un mode de représentation habituel et synthétique de résultats expérimentaux. Ces données sont interrogées à l'aide d'une architecture médiateur fondée sur une ontologie du domaine, ontologie développée lors du projet Sym'Previus. Pour utiliser cette ontologie lors de l'interrogation, nous devons représenter les données à l'aide du vocabulaire présent dans l'ontologie. Plus précisément, notre approche permet de découvrir dans les tableaux des instances de relations décrites dans l'ontologie. Nous avons défini une représentation XML - appelée SML (Semantic Markup Language) - permettant de stocker ces relations dans l'entrepôt avec un maximum de flexibilité (Gagliardi et al. (2005)). En effet, la transformation des données du tableau en une représentation utilisant le vocabulaire (termes et relations) de l'ontologie est complètement automatique.

En particulier, et c'est sur ce point que nous nous focalisons dans ce papier, il nous semble utile de pouvoir enrichir l'entrepôt avec des instances de relations existantes complétées d'informations imparfaitement identifiées ou même de garder, en cas d'échec lors de l'identification des relations représentées dans le tableau, une relation dite générique qui permettra de conserver l'existence d'un lien (même si il n'est pas identifié) entre certaines données. Ces relations sont stockées et peuvent être exploitées lors de l'interrogation. Elles peuvent également servir de base à un enrichissement de l'ontologie si un expert les identifie par la suite.

Ce papier est structuré de la manière suivante. Nous présentons en section 2 le format d'entrée des tableaux ainsi que le format SML dans lequel ces tableaux sont transformés. En section 3 nous présentons l'alimentation de l'entrepôt avec des relations enrichies et des relations génériques candidates. Nous présentons ensuite comment ces relations peuvent être exploitées lors d'une interrogation, avant de terminer par une étude des premiers résultats obtenus.

2 Notions préliminaires

Dans le projet e.dot, les données sont acquises par un crawler combiné à un outil de filtrage qui ne sélectionne que les documents html ou pdf qui contiennent des tableaux présentant des mots clef de l'ontologie Sym'Previus (http ://www.symprevius.net). Ces tableaux sont tout d'abord transformés dans un format XML utilisant des tags syntaxiques indépendants du domaine (format XTab). Ils sont donc représentés par une liste de lignes, chacune comportant une liste de cellules (Fig. 1). L'ontologie, développée par des experts lors du projet Sym'Previus, concerne le domaine du risque microbiologique. Elle comporte une taxonomie de 428 termes ainsi que 25 relations sémantiques qui sont caractérisées par leur signature. Par exemple, l'ontologie comporte la relation *foodFactorMicroorganism* qui a pour signature (*food, factor, microorganism*).

Notre système utilise cette ontologie pour transformer le document XTab en document SML - Semantic Markup Language - où les lignes du tableau ne sont plus représentées par des cellules mais par un ensemble de relations.

Dans l'exemple des figures 1 et 2, la relation *foodPh* qui lie un aliment à sa valeur de pH a été reconnue. Elle est représentée et instanciée par les valeurs apparaissant dans les cellules du tableau présentes dans $< originalVal >$. L'utilisation d'opérations de mapping nous permet

Products	pH values
Cultivated mushroom	5.00
Crab	6.60

```
< ?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<table> <title> <table-title>
approximative pH of some food products</table-title>
<column-title>Products</column-title>
<column-title>pH values</column-title> </title>
<nb-col>2</nb-col>
<content>
<line>
<cell>cultivated mushroom</cell>
<cell>5.00</cell>
</line>
<line>
<cell>crab</cell>
<cell>6.60</cell>
</line>
</content> </table>
```

FIG. 1 – *Un tableau et sa représentation XTab*

```
<table> <title> <table-title>
approximative pH of some food products </table-title>
<column-title> Products </column-title>
<column-title>pH values</column-title>...
</title> <content>
<rowRel>
<foodPH>
<food> <ontoVal>mushroom</ontoVal>
<originalVal> cultivated mushroom </originalVal>
</food>
<ph> <ontoVal/>
<originalVal>5.00</originalVal> </ph> </foodPH>
</rowRel>
<rowRel>
<foodPH>
<food> <ontoVal>crab</ontoVal>
<originalVal>crab</originalVal> </food>
<ph> <ontoVal/> <originalVal>6.60</originalVal> </ph>
</foodPH> </rowRel>
</content> </table>
```

FIG. 2 – *Une représentation SML simplifiée de la représentation XTab de Fig. 1*

d'associer à chacune de ces valeurs un ou plusieurs termes de l'ontologie et ces termes peuvent être utilisés lors de l'interrogation du document, interrogation guidée par l'ontologie. La valeur originale, lorsqu'elle diffère du terme de l'ontologie, peut alors être visualisée sur demande de l'utilisateur.

La reconnaissance des relations s'effectue en deux étapes : dans un premier temps le système associe, lorsque cela est possible, un terme de l'ontologie à chaque colonne du tableau, ce qui permet d'obtenir le schéma du tableau. Ensuite, le système représente l'ensemble des relations sémantiques dont la signature est compatible avec un sous-ensemble du schéma du tableau. La transformation étant complètement automatique, le système est également flexible lors du processus de reconnaissance de relations de l'ontologie dans le tableau.

Il est en effet possible de reconnaître des relations même si tous leurs attributs ne sont pas identifiés dans le tableau ; tel est le cas si, par exemple, l'un des attributs est une constante présente dans le texte qui environne le tableau. Il s'agit alors de relations dites partielles. Nous avons pu montrer dans Gagliardi et al. (2005) que la possibilité de reconnaître des relations

partielles permettait d'augmenter significativement le rappel sans trop faire chuter la précision.

3 Découverte de relations candidates

La représentation de relations partielles permet d'alimenter l'entrepôt avec des informations qui semblent être incomplètes. Il s'agit maintenant de savoir ce que le système doit faire des informations contenues dans les tableaux et qu'il n'a pas pu représenter dans une instance de relation de l'ontologie. Nous avons distingué deux cas de figures : celui où nous utilisons les informations pour compléter une relation reconnue et celui où aucune relation n'a pu être identifiée dans le tableau.

3.1 Capture de l'information non reconnue dans des attributs supplémentaires

A l'étape de l'identification des colonnes du tableau, il arrive souvent que certaines colonnes ne soient associées à aucun attribut ou qu'elles soient identifiées mais ne soient associées à aucune relation reconnue dans le tableau. Nous considérons alors que ces attributs sont des informations additionnelles qui peuvent venir préciser ou modifier l'interprétation d'une relation reconnue. Les informations de ces colonnes sont alors représentées dans le document SML par des attributs supplémentaires à l'aide du tag générique <attribut>. Ce dernier est ajouté à toute relation reconnue dans le tableau.

Ces informations additionnelles peuvent jouer plusieurs rôles :

- recueillir un attribut non identifié dans une relation partielle (ex : attribut numérique) ;
- recueillir un argument supplémentaire de la relation qui n'est pas défini dans l'ontologie (ex : forme ou quantité du produit) afin de préciser l'interprétation de la relation ;
- recueillir une relation supplémentaire qui partage des attributs avec la relation reconnue (ex : le tableau comportant les colonnes food, lipid, ph, microorganism).

Le tableau de la figure 3 représente une étude de l'influence de trois facteurs expérimentaux sur la croissance de la *listeria* dans les *aliments fumés*. Dans cet exemple de tableau, le document SML correspondant comporte deux attributs supplémentaires permettant de représenter deux attributs *Factor* représentés dans les 4ème et 5ème colonnes. Le système n'a en effet pas reconnu que les colonnes ayant pour titre *growth rate* et *egrc at 5○c (log10 cfu/day)* représentaient des attributs *Factor*.

microorganism	food	temperature	growth rate	egrc at 5○c (log10 cfu/day)
listeria	coppa (smoked pork)	4○c	2.1 logs in 28 days	0.107
listeria	cold-smoked salmon	8○c	5.4 logs in 21 days	0.116

FIG. 3 – *Microorganismes et aliments fumés*

Dans ce deuxième exemple de tableau **??**, l'attribut additionnel, que l'on peut voir dans le document SML de la figure 7, permet de compléter la relation reconnue par une information qui n'a pas été prévue dans l'ontologie. Ainsi, dans le domaine du risque alimentaire, on trouve souvent des tableaux qui précisent des références d'articles pour chaque résultat présenté dans

une ligne. Ce paramètre n'a pas été prévu dans la signature de la relation mais il est recueilli dans un attribut additionnel et peut être proposé par la suite à l'utilisateur.

microbe	vehicle/source	reference
Campylobacter spp.	poultry	Friedman et al 2000 [1]
Clostridium perfringens	meat	Todd 1997 ; Olsen, S. J. et al. 2000 [1]

FIG. 4 – *Aliments et Microorganismes*

```
<table>
</table-title >
<column-title> microbe </column-title>
<column-title>vehicle/source</column-title>
<column-title>reference</column-title>
<column-nb> 3 </column-nb>
<content>
<rowRel additionalAttr="yes">
...
<foodMicroorganism relType ="completeRel">
<food><finalVal>...</food>
<microorganism> <ontoVal>campylobacter</ontoVal>
<originalVal>campylobacter spp. </originalVal> </microorganism>
< attribute indMatch="attribut"> </ontoVal> <originalVal> Friedman et al 2000 </originalVal>
</attribute>
</foodMicroorganism>
...

</rowRel> ...
</content> </table>
```

FIG. 5 – *Représentation SML du tableau de la fig.4 - Attribut supplémentaire*

3.2 Représentation de tableaux par une relation générique

Nous allons maintenant montrer le rôle des relations génériques dans la représentation des informations des tableaux et leur intérêt lors de l'interrogation. Une relation générique permet de représenter les données d'un tableau dans lequel aucune relation de l'ontologie n'a été reconnue. L'échec de la reconnaissance des relations sémantiques est dû à plusieurs facteurs :

- les relations du domaine représentées dans le tableau ne sont pas décrites dans l'ontologie du domaine ;
- des tableaux peuvent représenter des informations concernant deux thématiques à la fois. Par exemple, certains des tableaux sur l'épidémiologie représentent des informations concernant le risque alimentaire. Il s'agit en fait d'une thématique connexe ;
- des attributs de relations sémantiques non reconnus dans le tableau empêchent la reconnaissance des relations. C'est en particulier le cas pour des attributs qui représentent des valeurs numériques et pour lesquels le processus de reconnaissance se fonde entièrement sur le titre des colonnes. Dans ce cas, la relation générique permet de représenter les attributs reconnus et non reconnus.

Les relations génériques permettent de conserver ces informations. La figure 7 représente, par exemple, la dose à laquelle on peut dire qu'un germe a infecté un support. Il s'agit d'une relation MicroorganismeFacteur mais la colonne numérique qui a pour titre infective dose et qui correspond au facteur n'a pu être reconnue. Ce tableau a pour représentation SML le document présenté en figure 8, document dans lequel les valeurs apparaissant dans les colonnes sont conservées et liées par une relation générique.

Pathogen	**Infective dose**	**reference**
Campylobacter spp.	500-800	Robinson 1981 ; Black et al 1988
Clostridium perfringens	10^7	Bermith 1988

FIG. 6 – *Doses infectieuses*

```
<table>
</table-title >
<column-title> pathogen </column-title>
<column-title>infective dose</column-title>
<column-title>reference</column-title>
<column-nb> 3 </column-nb>
<content>
<rowRel additionalAttr="no">
...
<relation relType ="generic">
<attribute indMatch="microorganisme"><finalVal>...</attribute>
<attribute indMatch="attribut"> <ontoVal>campylobacter</ontoVal>
<originalVal>campylobacter spp. </originalVal> </microorganism>
< attribute indMatch="attribut"> </ontoVal> <originalVal> Robinson 1981 ; Black et al 1988
</originalVal>
</attribute>
</relation>
...

</rowRel> ...
</content> </table>
```

FIG. 7 – *Représentation SML du tableau de la fig.6 - Relation générique*

4 Interrogation des données imparfaitement identifiées

Dans le cadre du projet e.dot, le moteur d'interrogation MIEL++ est capable d'interroger les deux bases existantes, relationnelle et graphe conceptuels, en combinaison avec l'interrogation de la base SML. Il est donc en mesure de traduire une requête MIEL en une requête sur la base des documents SML. Le vocabulaire utilisé pour exprimer les requêtes et les relations sémantiques à interroger est représenté dans l'ontologie Sym'Previus.

Ainsi, l'interrogation de l'entrepôt SML et des bases pré-existantes se fait de manière uniforme et transparente pour l'utilisateur par le biais d'une même interface graphique qui permet à l'utilisateur de sélectionner dans l'ontologie ses attributs de projection et de sélection.

Une extension de ce moteur d'interrogation peut permettre d'interroger les données additionnelles non identifiées ou imparfaitement identifiées qui apparaissent en tant qu'attributs supplémentaires et relations génériques.

4.1 Exploitation des attributs supplémentaires

Les attributs supplémentaires peuvent être exploités pour toute requête dont les réponses proviennent de relations avec attributs supplémentaires. Pour permettre à l'utilisateur d'avoir plus d'informations concernant le contexte des réponses à sa requête, le contenu et les titres des colonnes correspondant aux attributs supplémentaires peuvent être affichés.

Voici un exemple de requête Q1 où l'utilisateur cherche les aliments pouvant véhiculer le microorganisme "*Campylobacter spp.*". La figure 8 représente le résultat de l'évaluation de la requête Q1 sur l'entrepôt de données SML contenant des documents représentant des informations sur "*Campylobacter spp.*". Parmi ces document se trouve le document SML de la figure 7.

Microorganisme	**Aliment**	**Attributs Supp.**
campylobacter	poultry	Référence : Friedman et al 2000
campylobacter	turkey	

FIG. 8 – *La réponse à la requête Q1*

Cet exemple montre l'intérêt de présenter le contenu des attributs supplémentaires. Ici, la complétion de la réponse par la référence bibliographique intéressera vraisemblablement l'utilisateur. La présentation du titre de la colonne devant la valeur de cet attribut facilite l'interprétation de cette nouvelle information.

4.2 Exploitation des relations génériques

L'exploitation des relations génériques par l'utilisateur consiste entre autres en l'interrogation de la base de données SML par des requêtes par mots clés. Une requête Q2 cherchant toute l'information que la base de données contient au sujet du microorganisme *"Campylobacter"* est un exemple d'une telle requête par mots clés. L'évaluation de cette requête contiendra toutes les relations identifiées et toutes les relations génériques dont une des valeurs du tag *ontoVal* est "*Campylobacter*" ou une spécialisation de "*Campylobacter*".

Attribut1	**Attribut2**	**Attribut3**
Pathogen : campylobacter	Infective dose : 500-800	reference : Robinson 1981 ; Black et al 1988
Organism : campylobacter	Temp : 5◦C	

FIG. 9 – *Réponse à la requête Q2*

Dans la figure 9 est présenté un extrait de la réponse à la requête Q2 qui correspond à une ligne de la relation générique représentant le tableau de la figure 7. Si nous n'avions pas interrogé les relations génériques, nous aurions raté cette réponse et nous aurions peut-être

même obtenu aucune réponse dans l'hypothèse où aucune relation de l'ontologie ne contient la valeur *"campylobacter"*.

5 Evaluation de l'approche

Nous avons évalué notre approche d'enrichissement sémantique sur 61 documents XTab représentant des tableaux de données réelles. Ces derniers ont été extraits de documents collectés sur le Web. Un document XTab est sélectionné pour l'évaluation s'il contient au moins une valeur correspondant à un terme de l'ontologie Sym'Previus.

Dans la section 5.1 nous décrivons la chaîne de traitement des documents. Dans la section 5.2, nous présentons et interprétons les résultats de l'évaluation de la découverte de relations candidates.

5.1 Chaîne de traitement des documents évalués

L'application AQWEB est un service Web qui a été développé dans le but de faciliter la tâche de validation par un expert des différents modules développés dans le projet e.dot. Cette application permet de relier les différentes tâches intervenant dans la chaîne de traitement des documents, de leur collecte jusqu'à leur enrichissement sémantique.

Le service permettant de rechercher et de collecter des documents PDF [2] potentiels à partir du Web considère en entrée un tuple (aliment, microorganisme, expressions exactes, mots exclus, langue du document, table, références) et récupère des documents vérifiant la directive exprimée dans ce tuple. Les fichiers PDF sont convertis automatiquement en format Word avec l'outil Omnipage de Scansoft. Cette tâche est asynchrone. Les fichiers convertis en Word doivent être validés car il arrive parfois que les documents Word générés contiennent des erreurs, notamment dans les polices de caractères. La présentation des formats de tableaux peut être aussi simplifiée pour se ramener à un format de type XTAB (une seule ligne d'entête, sans groupements de cellules et plusieurs lignes de détail).

L'ensemble des tableaux de données se trouvant dans un fichier Word validé est converti automatiquement en un ensemble de documents XML au format XTAB. Ces documents XTab sont enfin enrichis sémantiquement par le module XTab2SML que nous avons développé.

5.2 Résultats de l'évaluation

Les résultats de l'évaluation de l'identification des relations sémantiques de l'ontologie dans les documents XTab ont montré que le rappel augmente significativement quand on conserve les relations partiellement identifiées. En effet, le rappel vaut 0,45 lorsque nous ne conservons que les relations parfaitement reconnue et vaut 0,65 lorsque nous gardons les relations parfaitement et partiellement reconnues. Ces résultats ont confirmé l'intérêt de cette première flexibilité dans la représentation des documents en SML.

Nous montrons dans ce qui suit les résultats de l'évaluation concernant la découverte de relations sémantiques candidates à l'enrichissement d'un entrepôt thématique. Ces relations

[2] la majorité des publications scientifiques diffusées sur le Web sont dans le format PDF

candidates à l'enrichissement peuvent être représentées par une relation comportant un ensemble d'attributs supplémentaires ou par une relation générique.

5.2.1 Relations comportant des attributs supplémentaires

Ces relations comportent les attributs reconnus et intégrés dans des relations sémantiques de l'ontologie et également les attributs supplémentaires. Nous avons pu voir que 24/61 (soit 40%) des tableaux testés contiennent des colonnes ne pouvant être associées à aucune relation sémantique identifiée ; ces colonnes sont donc représentées par des attributs supplémentaires. Ces derniers jouent essentiellement deux rôles : le premier consiste en la complétion d'une relation sémantique reconnue ce qui permet à l'utilisateur d'avoir une interprétation plus précise de la relation ; le second consiste en la représentation des données d'attributs non reconnus (oubliés) dans une relation déjà reconnue. Environ 67 % des attributs supplémentaires viennent compléter des relations sémantiques reconnues ou, par distribution des attributs, permettent de représenter de nouvelles relations. En effet, une distribution des attributs reconnus et des attributs supplémentaires peut représenter des relations sémantiques existantes ou nouvelles et candidates à l'enrichissement de l'entrepôt. Les résultats montrent que 16/49 (environ 33%) des attributs représentent des attributs oubliés dans des relations reconnues.

	Nombre d'attributs
Attributs supplémentaires	49
Capture des attributs oubliés	16
Complétion de relations	33

FIG. 10 – *Résultat de l'évaluation de l'identification des attributs*

5.2.2 une relation générique

Elle permet de représenter les liens sémantiques existant entre les données d'un tableau, dans le cas où aucune des relations sémantiques de l'ontologie n'est reconnue. Une relation générique contient soit des attributs reconnus mais qui ne peuvent pas être combinés afin d'instancier une relation existante, soit des attributs représentant des colonnes non reconnues. La figure 11 présente les statistiques concernant l'identification de relations sémantiques de l'ontologie dans les documents XTab testés. Dans ces tableaux, 34 relations génériques ont été générées, 93 relations sémantiques ont été reconnues correctes et 48 relations sémantiques de l'ontologie ont été oubliées.

Parmi les relations génériques candidates à l'enrichissement, nous distinguons trois cas :

- des relations déjà existantes dans l'ontologie pour lesquelles nous n'avons pas pu reconnaître les attributs nécessaires à leur instantiation (au moins deux attributs). Il s'agit dans ce cas de la découverte d'instances de relations sémantiques existantes mais oubliées ;
- des relations représentant des informations du domaine de l'ontologie (le risque microbiologique dans les aliments) et pour lesquelles la combinaison des attributs ne peut pas correspondre à une relation de l'ontologie. Il s'agit dans ce cas d'une relation du

domaine, candidate à l'enrichissement non seulement de l'entrepôt mais également de l'ontologie du domaine ;

– des relations génériques représentant des données d'un domaine connexe. Dans ce cas, nous pouvons envisager de proposer aux experts du domaine ces relations découvertes, en vue d'étendre l'ontologie existante en une ontologie plus générale, par exemple, en transformant l'ontologie du risque microbiologique dans les aliments en une ontologie traitant également du domaine du risque chimique dans les aliments. En effet, dans notre ensemble de tests, nous avons traité des tableaux qui concernent le domaine du risque chimique et des tableaux qui concernent l'épidémiologie. Nous devons toutefois signaler que nous avons également traité 23% de tableaux qui n'étaient guère pertinents pour le domaine (comme par exemple les carences nutritionnelles du poisson chat).

	Nombre de relations
Relations trouvées	147
Relations trouvées correctes	93
Relations oubliées	48
Relations incorrectes	54
Relations génériques	34

FIG. 11 – *Résultat de l'évaluation de la découverte de relations sémantiques*

Les résultats présentés dans la figure 12 montrent que 17% des relations génériques permettent de représenter des données de relations oubliées (parmi 48 relations oubliées). Ces résultats montrent également que 60% des relations génériques représentent des relations pertinentes pour l'enrichissement de l'entrepôt et donc intéressantes, par la suite, pour l'interrogation par les utilisateurs. Le faible nombre de tableaux et donc de relations génériques qui ne présentent pas d'interêt pour le domaine peut s'expliquer par la précision de la directive appliquée lors de la collecte des documents sur le Web. En effet, si les critères de sélection du document sont assez précis, les tableaux contenus dans le document ont de fortes chance d'être pertinents pour le domaine et donc de donner lieu à une relation générique intéressante.

	Nombre de relations
Relations non pertinentes	8
Relations de l'ontologie oubliées	6
Relations pertinentes	19

FIG. 12 – *Intérêt des relations génériques*

6 Conclusion

Il existe une multitude de travaux dans le domaine de la découverte de connaissances dans les documents. Parmi ceux qui se sont intéressés à la découverte de relations sémantiques,

nous citons les systèmes MeatAnnot Khelif et Dieng-Kuntz (2004) et Séguéla et Aussenac-Gilles (1999). Nous citons également les approches proposées par Pivk et al. (2005) et Ling et al. (1998).

MeatAnnot est un système d'extraction d'instances de concepts et de relations à partir de texte guidé par l'ontologie UMLS. MeatAnnot permet une génération semi-automatique d'annotations. Dans cette approche deux outils ont été utilisés : GATE pour l'étiquetage grammatical, Syntex pour l'extraction de syntagmes verbaux pouvant correspondre à des instances de relations. A partir de ces données, une grammaire d'extraction de relations est écrite manuellement. Le résultat de l'extraction est ensuite proposé à la validation de l'utilisateur et représenté en RDF.

Le système CAMELEON Séguéla et Aussenac-Gilles (1999) permet d'extraire des instances de relations sémantiques entre concepts en utilisant des marqueurs sur un corpus d'apprentissage (exemple : la présence de "X être article-indéfini Y" pour la relation /emphest-un).

Ces deux systèmes sont semi-automatiques. De plus, l'extraction de relations sémantiques est effectuée à partir de textes assez riches pour apprendre les règles d'extraction. Ce type d'approche ne peut être appliqué dans notre cas puisque le seul contexte dont nous disposons est celui du tableau (titres et contenu des lignes). Le travail de (Pivk et al. (2004)) propose une approche permettant de générer automatiquement un ensemble de méthodes (en utilisant une Frame-logique) permettant d'extraire une donnée d'un tableau. Chaque méthode est représentée par un ensemble de paramètres et par un type de retour et permet d'extraire des instances de relations sémantiques.

Dans le travail proposé par Ling et al. (1998), l'ontologie comporte un ensemble de termes du domaine, leurs caractéristiques lexicales, leur contexte d'apparition à l'aide de mots clés et de relations sémantiques entre les termes. A partir de l'ontologie, l'outil peut concevoir et alimenter automatiquement une base de données relationnelle par des informations reconnues et extraites à partir des documents donnés en entrée. L'ontologie est construite manuellement par un expert du domaine visé. Cette approche suppose une importante homogénéité des documents traités.

Nous avons proposé dans ce travail une approche complètement automatique de découverte de relations sémantiques candidates à l'enrichissement d'un entrepôt thématique. Ces relations peuvent être utilisées, dans un premier temps, pour enrichir les réponses de l'utilisateur. Il peut s'agir soit d'informations qui viennent s'ajouter à une relation connue, soit de liens sémantiques potentiels entre valeurs présentes dans une même ligne de tableau. Ces relations peuvent être, par la suite, proposées à un expert du domaine pour enrichir l'ontologie du domaine ou même pour l'étendre à une ontologie plus générale. En l'occurrence, il s'agirait ici d'étendre l'ontologie du risque microbiologique dans les aliments à une ontologie plus générale qui serait l'ontologie du risque alimentaire. Cette dernière représenterait les connaissances du domaine du risque microbiologique, du domaine du risque chimique et du domaine de l'épidémiologie. Une des améliorations possibles de cet outil est de représenter également en SML les parties moins structurées des tableaux de façon à pouvoir les exploiter au mieux lors des requêtes. Cet outil pourra être intégré à la plate-forme logicielle qui sera développée dans le cadre du projet WebContent.

Références

Buche, P., J. Dibie-Barthélemy, O. Haemmerlé, et M. Houhou (2004). Towards flexible querying of xml imprecise data in a dataware house opened on the web. In *Flexible Query Answering Systems (FQAS)*. Springer Verlag.

e.dot (2004). Progress report of the e.dot project. http ://www-rocq.inria.fr/ amann/edot/.

Gagliardi, H., O. Haemmerlé, N. Pernelle, et F. Saïs (2005). A semantic enrichment of data tables applied to food risk assessment. *DS '05, 8th International Conference on Discovery Science, Singapore, October 2005, Proceedings 3735*, 374–376. Lecture Notes in Computer Science.

Khelif, K. et R. Dieng-Kuntz (2004). Ontology-based semantic annotations for biochip domain. *EKAW*, 483–484.

Ling, T. W., S. Ram, et M.-L. Lee (Eds.) (1998). *Conceptual Modeling - ER '98, 17th International Conference on Conceptual Modeling, Singapore, November 16-19, 1998, Proceedings*, Volume 1507 of *Lecture Notes in Computer Science*. Springer.

Pivk, A., P. Cimiano, et Y. Sure (2004). From tables to frames. In *International Semantic Web Conference*, pp. 166–181.

Pivk, A., P. Cimiano, et Y. Sure (2005). From tables to frames. *Journal of Web Semantics 3*.

Séguéla, P. et N. Aussenac-Gilles (1999). Extraction de relations sémantiques entre termes et enrichissement de modèles du domaine. *IC, Ingéneirie des Connaissance, Palte-forme AFIA*.

Í

Summary

This work aims at enriching automatically a thematic data warehouse composed of heterogeneous documents which are extracted from the Web and which contain tables. This enrichment is guided by an ontology composed of a set of terms and a set of semantic relations. In this article, we present an approach of unidentified relations discovering as attributes discovering which can enrich or modify the interpretation of identified relations.

Etude des stratégies cognitives mises en œuvre lors de la recherche d'informations sur Internet.

Emma Holder *, Josette Marquer **

* holderemma@yahoo.com
** josette.marquer@univ-paris5.fr

Université René Descartes – Paris 5
71, avenue Edouard Vaillant
92100 Boulogne Billancourt

Résumé. L'objectif de ce travail est de mieux connaître le fonctionnement cognitif des sujets lorsqu'ils cherchent une information sur Internet. En adoptant une perspective centrée d'abord sur les stratégies cognitives individuelles, nous avons établi un modèle procédural par sujet et par tâche (Web vs. sites) puis nous avons regroupé ces modèles individuels en fonction du cheminement suivi jusqu'à l'information cible. Nous avons montré que le choix entre ces différents cheminements dépendait du type de tâche et qu'il existait une forte variabilité inter et intra individuelle. L'étude des représentations que les sujets ont des sites et du Web montre qu'il existe une liaison entre qualité des représentations et vitesse d'exécution, d'une part, et expérience antérieure d'Internet, d'autre part, mais seulement pour la recherche d'information sur l'ensemble du Web. Enfin, d'une manière générale, les stratégies les plus fréquemment mises en œuvre semblent être celles qui donnent lieu aux représentations les plus exactes.

1 Introduction

L'objectif de ce travail est de mieux connaître le fonctionnement cognitif des sujets lorsqu'ils cherchent une information sur Internet afin de proposer des aides à la navigation qui soient mieux ciblées.

Nombreuses sont les études qui, dans le domaine de l'interaction Homme-Ordinateur (HCI : Human-Computer Interaction), ont cherché à améliorer les aides à la navigation sur Internet en modifiant, par exemple, certaines caractéristiques des interfaces et des moteurs de recherche, afin d'aboutir à une plus grande efficacité et/ou un niveau de satisfaction plus élevé des utilisateurs. Or, dans ces recherches, les aides à la navigation sont très souvent conçues pour un utilisateur idéal : la diversité des utilisateurs d'Internet est généralement ramenée à un moyennage des performances et la variabilité interindividuelle, étant considérée comme entièrement aléatoire, est mise de côté. Cependant, la nécessité de s'adapter au plus grand nombre a été à l'origine d'un changement de point de vue dans l'étude de l'interaction Homme-Ordinateur, passant ainsi d'une optique proche de la psychologie expé-

rimentale générale à une perspective différentielle plus centrée sur les caractéristiques des sujets eux-même.

Ainsi, à la suite des travaux d'Egan et Gomez (1985) qui, les premiers, ont démontré le rôle de certaines caractéristiques propres aux individus dans la manipulation d'un logiciel, Allen (1999) a pu mettre en évidence le lien entre certaines aptitudes comme la vitesse perceptive et les stratégies de recherche d'information. De même, Moss et Hale (1999) ont observé une certaine constance dans les stratégies individuelles, stratégies dont le choix semble lié à l'expérience antérieure d'Internet des sujets. Parallèlement, l'étude publiée par Navarro-Prieto, Scaiffe et Roger (1999) montre qu'il existe, pour le choix des stratégies, une interaction entre le niveau d'expertise des sujets et la nature de la structure dans laquelle l'information cherchée est présentée sur le Web. En effet, selon ces auteurs, les novices auraient tendance à utiliser une stratégie mixte, quelle que soit la tâche de recherche, alors que les sujets expérimentés semblent adapter le choix de leur stratégie à la nature de l'information à trouver. Outre l'expérience antérieure du Web et les aptitudes cognitives des sujets, Ford (2000) met en évidence des interactions entre le style d'apprentissage (holistique/sériel) et la nature de l'information. De même, Palmquist et Kim (2000) étudient les interactions entre la dépendance/indépendance à l'égard du champ (DIC) et l'expérience antérieure d'Internet sur le choix des différentes stratégies de recherche sur le Web. Ces auteurs montrent que l'expérience est un facteur modulateur des performances des sujets dépendants du champ : de tels sujets voient ainsi leurs performances tendre vers celles des sujets indépendants, lorsqu'ils sont expérimentés.

C'est dans la suite de l'ensemble de ces travaux que nous situons l'objectif de cette recherche. Nous nous proposons d'adopter cette perspective différentielle, mais en nous centrant d'abord sur l'analyse des stratégies individuelles elles-mêmes, comme on a pu le faire par ailleurs dans le cadre de différents paradigmes du traitement de l'information (Marquer & Pereira, 2002 ; Marquer, 2005).

Ainsi, nous établirons d'abord un modèle procédural par sujet et par tâche. Puis nous étudierons les variations inter et intra individuelles des stratégies de recherche d'information en fonction des tâches. Enfin, nous nous intéresserons aux représentations qu'ont les sujets des sites et du Web au travers de la qualité de leurs anticipations lorsqu'ils appuient sur une touche du clavier ou sur la souris. Nous testerons l'hypothèse qu'il existe une liaison entre la qualité des représentations et la vitesse d'exécution des tâches ; nous nous demanderons également s'il existe un lien entre la qualité des anticipations et, d'une part, le niveau des sujets dans certaines aptitudes, d'autre part, leur expérience antérieure de la navigation sur le Web ; enfin, nous analyserons les liens entre le type de stratégie mise en œuvre et la qualité des anticipations.

2 Méthode

2.1 Sujets

Trente jeunes adultes, 21 femmes et 9 hommes, âgés de 18 à 27 ans ont participé à l'expérience. Tous sont étudiants de l'Enseignement Supérieur.

Chacun d'eux a déjà été amené (au moins au cours de ses études) à manipuler Internet ; cependant, la fréquence de l'utilisation et le niveau d'habileté varient en fonction des individus. Ce dernier aspect est évalué au moyen d'un questionnaire préliminaire aux sessions de recherche sur Internet.

2.2 Tâches

Quatre tâches de nature différente leur ont été proposées : deux tâches de recherche d'information portant sur l'ensemble du Web et deux tâches de recherche sur des sites déterminés.

Pour les sessions de recherche sur l'ensemble du Web, l'information à trouver consiste soit en un fait précis (la date de naissance du lauréat du Prix Nobel de Littérature 1950), soit en un ensemble de données (le maximum de longs métrages ayant porté sur l'histoire de Peter Pan d'après J.M. Barrie), à partir d'un navigateur imposé (yahoo.fr).

La recherche à effectuer sur des sites déterminés consiste à simuler l'achat de produits répondant à certains critères précis, en s'efforçant de trouver la meilleure solution possible. Ainsi, les sujets doivent trouver un miroir d'au moins 140 cm de hauteur le moins cher possible sur le site d'Ikea et une crème cosmétique anti-rides la moins chère possible sur le site Club des Créateurs de Beauté (CCB).

Tout au long du déroulement de chacune de ces quatre tâches, nous avons enregistré la séquence des opérations effectuées par chaque sujet et nous avons recueilli ses verbalisations concourantes. Nous avons ainsi essayé de savoir s'il avait ou non un « plan d'action » (une stratégie) plus ou moins précis pour guider sa navigation. Pour cela, après chaque appui sur le clavier ou la souris, l'écran était caché et le participant invité à dire ce qu'il espérait trouver sur la page sélectionnée. Ceci nous a ainsi permis d'évaluer la représentation que ces individus ont d'une recherche à effectuer au travers des bonnes et mauvaises « anticipations ».

Enfin, les sujets de notre échantillon ont été soumis à différents tests d'aptitudes comme le test de signification verbale des PMA de Thurstone, le test du GEFT permettant d'évaluer la dépendance/indépendance à l'égard du champ (DIC), le Stroop mesurant l'aptitude à maintenir l'attention sur une cible quand des distracteurs sont présents, et la tâche de mémoire spatiale du French Kit. Nous avons également relevé un ensemble de données relatives aux caractéristiques personnelles des individus participants (informations démographiques et expérience antérieure de l'utilisation d'Internet).

3 Résultats

3.1 Les stratégies de recherche sur Internet

Les stratégies relevées pour les quatre tâches ont fait l'objet d'une modélisation individuelle traduisant le cheminement suivi par chaque sujet jusqu'à l'information cible. Nous avons ensuite regroupé les modèles individuels similaires en catégories en fonction de la structure de ces stratégies, pour chaque tâche d'abord, puis en fonction du type de recher-

che (Web / sites), et enfin de façon globale pour l'ensemble des tâches. Le tableau 1 résume les résultats de cette analyse :

Recherches sur le Web	Stratégies	Recherches sur les sites de e-commerce
Navigateur, puis utilisation d'un ou des liens	**Stratégies hiérarchiques A**	Recherche par liens hiérarchiques Recherche par moteur de recherche interne
Plusieurs sessions navigateur puis utilisation des liens	**Stratégies arborescentes B**	Recherche par liens catégoriels et hiérarchiques
Alternance de recherches navigateur et recherche par liens Yahoo	**Stratégies alternées Arborescentes – hiérarchiques C**	Recherche par moteur de recherche interne puis nouvelle session par liens proposés Alternance de sessions de recherche par moteur de recherche interne et liens hiérarchiques
Sessions de recherche sur le navigateur avec utilisation des liens proposés en réponse et, éventuellement, du moteur de recherche interne à ces liens.	**Stratégie arborescente – Hiérarchique – arborescente D**	
	Stratégies alternées Hiérarchiques – arborescentes E	Alternance de sessions de recherche par liens hiérarchiques et par moteur de recherche interne
	Stratégies de Recherche directes F	Recours à un site connu antérieurement

TAB. 1 – Récapitulatif des catégories de stratégies mises en places lors des différentes sessions de recherche Internet.

3.2 Variabilité inter et intra individuelle

La variabilité interindividuelle des catégories de stratégies ne semble pas la même pour les 2 tâches sur le Web : pour rechercher les films inspirés de Peter Pan, les sujets peuvent mettre en œuvre 5 catégories de stratégies sur 6, avec une prédominance des stratégies arborescentes (catégorie B) alors que pour trouver une information sur un prix Nobel, ce sont essentiellement 2 catégories qui sont adoptées, les stratégies hiérarchiques (catégorie A) et arborescentes (catégorie B), avec des fréquences quasiment égales.

Pour les 2 tâches d'achat en ligne, la variabilité interindividuelle des stratégies semble très proche avec, dans l'ordre des fréquences d'utilisation, les stratégies hiérarchiques (A), alternantes entre hiérarchiques et arborescentes (C) et arborescentes (B).

Cependant, il existe une forte variabilité intra individuelle des stratégies puisqu'un seul sujet a procédé de la même manière quelle que soit la session de recherche d'information et que 4 sujets seulement ont utilisé la même stratégie pour 3 des 4 tâches.

Par ailleurs, un tiers des sujets ont adopté la même stratégie pour les 2 tâches sur le Web alors que 8 seulement conservent la même stratégie pour les 2 tâches sur les sites du CCB et d'Ikea.

Il semble donc nécessaire, au regard de ces observations, de tenir compte de la variabilité intra individuelle dans l'exploitation des données. Ainsi, les sujets utilisant la stratégie A pour une tâche ne sont pas forcément les mêmes que ceux qui utilisent la stratégie A lors d'une autre session de recherche de même nature, et a fortiori pour des tâches de nature différente.

3.3 Qualité de la représentation de la tâche à effectuer.

Afin d'étudier l'effet de la qualité des représentations sur les performances des sujets aux différentes tâches de recherche d'information sur le Web, nous avons relevé la proportion d'anticipations correctes émises chaque fois que l'écran est caché.

3.3.1 Liaison entre précision des anticipations et rapidité d'exécution de la recherche d'informations

Nous avons d'abord testé l'hypothèse d'une liaison entre la proportion de bonnes anticipations et le temps mis pour atteindre l'information-cible.

Pour la tâche "Peter Pan", nous ne pouvons pas tester cette hypothèse car 19 sujets sur 30 ont atteint la limite des 10 minutes sans avoir terminé la tâche.

Pour les trois autres tâches, il existe une corrélation significative entre le nombre de bonnes anticipations et le temps mis pour atteindre l'information cible. Pour les tâches "Prix Nobel", "CCB" et "Ikea", les coefficients de Bravais-Pearson, égaux, respectivement, à -.43, -.40 et -.62, sont significatifs au seuil unilatéral $p/2 < .025$.

La rapidité de la recherche des informations au cours de ces trois tâches semble donc bien liée à la qualité des représentations qu'ont les sujets soit du Web, soit des sites concernés, dans le sens suivant : plus les sujets font de bonnes anticipations et plus ils sont rapides. Une autre possibilité aurait été qu'ils privilégient la précision des anticipations au détriment de la vitesse.

3.3.2 Liaison entre précision des anticipations et niveau en aptitudes

Il n'existe de lien entre précision des anticipations et niveau en aptitudes que pour la tâche "Prix Nobel" et la dépendance/indépendance à l'égard du champ : pour cette tâche, plus les sujets sont indépendants du champ et plus ils ont tendance à faire de bonnes anticipations.

3.3.3 Liaison entre précision des anticipations et utilisation habituelle d'Internet

Il n'existe une liaison entre l'expérience antérieure d'Internet (de 1 à 8 ans) que pour les tâches de recherche d'information sur l'ensemble du Web.

Pour les tâches sur des sites particuliers, est-ce que le fait d'avoir déjà visité ces sites joue un rôle sur la précision des anticipations ? Cette hypothèse n'est confirmée pour aucun des deux sites : pour le site du CCB, la proportion moyenne de bonnes anticipations est égale à 0,76 pour ceux qui ne connaissent pas ce site contre 0,81 pour ceux qui le connaissent ($t(28) = -0,57$, NS à $p > .10$) ; pour le site d'Ikea, la proportion moyenne de bonnes anticipations est égale à 0,67 pour ceux qui ne connaissent pas ce site contre 0,69 pour ceux qui le connaissent ($t(28) = -0,40$, NS à $p > .10$)

3.3.4 Relation entre choix des stratégies et précision des anticipations

Pour les tâches utilisant les sites de commerce électronique, la proportion des anticipations correctes est plus forte pour les stratégies de la catégorie A, stratégies qui sont aussi les plus nombreuses. En d'autres termes, les sujets suivant un cheminement linéaire pour atteindre le produit demandé ont une meilleure représentation du site que les autres.

Pour la session "Prix Nobel", on observe une quasi égalité dans la précision moyenne des anticipations pour les stratégies hiérarchiques (A), arborescentes (B). Ces stratégies présentent la même fréquence d'utilisation et sont aussi les plus fréquemment mises en oeuvre.

En ce qui concerne la session "Peter Pan", les moyennes d'anticipations correctes diffèrent assez peu entre les différentes catégories de stratégies.

En résumé, sauf pour la tâche sur Peter Pan, quel que soit le type de recherche d'information, les stratégies les plus fréquemment mises en oeuvre sont également celles qui donnent lieu aux anticipations les plus exactes.

4 Conclusion

L'analyse des résultats met en évidence l'importance prédominante de la nature de la structure de l'information sur laquelle s'effectue la recherche dans le choix de la stratégie adoptée par les utilisateurs d'Internet. Ainsi, il existerait un lien entre le choix du « plan d'action » suivi par le sujet et le type de recherche à effectuer, la stratégie adoptée étant celle qui permet la meilleure représentation de la tâche, représentation elle-même liée à la rapidité d'exécution.

En ce qui concerne les liens éventuels entre aptitudes et choix des stratégies, nous n'avons trouvé un tel lien qu'entre la dépendance / indépendance à l'égard du champ et les procédures de recherche d'information sur le Prix Nobel de littérature. Nous projetons de poursuivre nos investigations en complétant l'évaluation des aptitudes par des tests portant sur une autre forme de style cognitif, l'impulsivité/réflexivité, mais également par des tests exécutifs, reconnus pour donner une indication sur certaines dimensions frontales responsables du raisonnement (OSPAN, Running Span, empans de chiffres, alphabétique, opérationnel, etc.).

L'analyse de la variable « qualité de la représentation » semble révéler l'existence d'anticipations de différentes natures, et ce aussi bien pour les réponses correctes que pour les erreurs. L'un des objectifs des travaux à venir portera sur l'étude détaillée de la répartition des ces anticipations en fonction de leur nature. Une telle analyse devrait fournir de nouvelles pistes pour l'amélioration des conditions de navigation.

Enfin, et parce que les sites sélectionnés ici sont conçus pour la vente à distance, nous nous attacherons à relever, à l'issue des passations, des données de nature conative comme le

niveau de satisfaction à l'égard du résultat de la recherche ou l'influence de l'appréciation de l'interface sur les performances.

Références

Allen, B. (2000). Individual differences and the conundrums of user-centered design : Two experiments. *Journal of the American Society for Information Science, 51 (6)*, 508-520.

Egan, E., Gomez, L.M. (1985). Assaying, Isolating, and Accomodating Individual Differences In Learning a Complex Skill. In : R.F. Dillon (Ed) *Individual Differences in Cognition, vol. 2*, 173-217. Orlando: Academic Press.

Ford, N. (2000). Cognitive styles and virtual environment. In : C. Chen ; M. Czerwinski ; R. Macredie (2000). Individual Differences In Virtual Environments – Introduction and overview. *Journal of the American Society for Information Science, 51 (6)*, 499-507.

Marquer, J. (2005). *Lois générales et variabilité des mesures en psychologie cognitive*. Paris :Editions l'Harmattan.

Marquer, J., Pereira, M. (2002). « Know the method your subject is using » and "Never average over methods" : an application of Newell's (1973) admonition to letter-matching. *Cognitive Science Quarterly, 2*, 141-162.

Moss, N., Hale, G. (1999) Cognitive Style and Its Effects on Internet Searching : A Quantitative Investigation. In : *Proceedings of the European Conference on Educational Research*, Lahti, Finland, September 22-25th.

Navarro-Prieto, R., Scaife, M., Rogers, Y. (1999). Cognitive Strategies in Web Searching. In : *Proceedings of the 5th Conference on Human Factors and the Web*. Gaithersburg, June 3rd.

Palmquist, R.A., Kim, K.-S. (2000). Cognitive style and on-line database search experience as predictors of web search performance. In : C. Chen ; M. Czerwinski ; R. Macredie (2000). Individual Differences In Virtual Environments – Introduction and overview. *Journal of the American Society for Information Science, 51 (6)*, 499-507.

Summary

As Internet extends, the need to fit the great majority of users becomes necessary. Analyzing individual cognitive strategies that people use to search the Web may provide new clues for adapting interfaces to this enlarged public.

Techniques structurelles pour l'alignement de taxonomies sur le Web

Hassen Kefi , Chantal Reynaud
Brigitte Safar

Université Paris-Sud XI, CNRS (L.R.I.) & INRIA (Futurs)
91405 Orsay cedex
{kefi, reynaud, safar}@lri.fr
http://www.lri.fr/~cr

Résumé. Ce papier porte sur la génération de mappings pour l'alignement de taxonomies du Web. L'objectif est de permettre un accès unifié aux documents d'un même domaine d'application. La recherche de documents s'appuie sur des taxonomies. Nous proposons d'aligner la taxonomie d'un portail Web avec celle de documents externes de façon à augmenter le nombre de documents accessibles à partir de ce portail sans en modifier l'interface d'interrogation. Ce papier présente les techniques structurelles de la méthode d'alignement que nous avons développée, des techniques qui s'appliquent en présence d'une dissymétrie dans la structure des taxonomies comparées. Des expérimentations ont été effectuées avec le prototype implémenté, TaxoMap.

1 Introduction

La recherche de documents pertinents sur le web est une tâche encore souvent laborieuse. Le Web sémantique devrait faciliter cette recherche en réalisant un appariement sémantique entre la requête de l'utilisateur et les documents indexés. Les techniques d'alignement des schémas de méta-données ou des ontologies sont au cœur du processus.

Notre travail porte sur de telles techniques, utilisables dans le contexte du Web. L'objectif est de permettre un accès unifié via le Web aux documents d'un même domaine d'application. La recherche de documents s'appuie sur des taxonomies de termes plus ou moins structurées. Nous proposons d'aligner la taxonomie d'un portail Web avec celle de documents externes de façon à augmenter le nombre de documents accessibles à partir de ce portail sans en modifier l'interface d'interrogation. Ce papier présente les techniques structurelles de la méthode d'alignement que nous avons développée, des techniques difficiles à appliquer a priori dans le contexte dans lequel on se situe car, si la taxonomie de concepts d'un portail Web est en général bien structurée, celle des autres documents accessibles ne l'est pas toujours. Les techniques que nous proposons s'appliquent donc en présence d'une dissymétrie dans la structure des taxonomies comparées. Ces techniques font partie d'une approche générique d'alignement de taxonomies mise en œuvre au travers d'un processus semi-automatique. Dans une première étape, des mises en correspondance dites probables sont automatiquement découvertes. Dans une seconde étape, des suggestions de mappings sont faites au concepteur. La découverte de mappings peut être vue comme un assemblage de techniques variées, appliquées dans un ordre bien défini : terminologiques, structurelles et sémantiques. Les techniques terminologiques, basées principalement sur des comparaisons de chaînes de caractères, sont appliquées en priorité car elles sont les plus à

même de fournir des mappings probables. Elles exploitent toute la richesse des noms des concepts. Même si elles sont efficaces, les techniques terminologiques ne peuvent cependant pas trouver l'ensemble des rapprochements possibles. Le système fait alors appel à des techniques structurelles et sémantiques[1]. Ce papier porte sur ces techniques.

Les techniques structurelles permettent de générer des mises en correspondance supplémentaires, moins sûres que celles générées par les techniques terminologiques et qui nécessitent d'être validées. Trois techniques structurelles sont proposées, basées sur des éléments de structure différents, mais ne consistant en aucun cas à rechercher des similarités structurelles entre les deux taxonomies, ce qui en fait toute leur originalité.

Ce papier est organisé de la façon suivante. Dans la section 2, nous décrivons l'approche d'alignement au sein de laquelle s'insèrent les techniques présentées. La section 3 présente successivement les trois techniques structurelles mises en œuvre. La section 4 porte sur les expérimentations réalisées. En section 5, des travaux proches sont cités et nous discutons, à la lumière de ces travaux, les caractéristiques des techniques retenues dans notre approche. Enfin, nous concluons.

2 Approche

L'objectif du processus d'alignement est de générer, le plus automatiquement possible, des appariements sur des taxonomies. Les critères utilisables pour déduire une mise en correspondance sont restreints. En effet, dans une taxonomie, un concept est uniquement défini par le label qui lui est associé (expression qui peut être composée de plusieurs mots) et par les relations de subsomption qui le relient à d'autres concepts.

Une taxonomie est un ensemble de concepts reliés par des relations *is-a*, représentée par des graphes acycliques. Les concepts sont représentés par des nœuds du graphe connectés par les liens orientés correspondant aux relations *is-a*.

Etant donné deux taxonomies, il s'agit de mettre en correspondance les éléments de l'une, appelée taxonomie source, avec les éléments de l'autre, appelée taxonomie cible. Le processus est orienté d'une taxonomie source vers une taxonomie cible. Les mappings à déterminer sont supposées être des relations de type 1:1. Le processus d'alignement a pour objectif de générer deux types de relations : des relations d'équivalence et des relations de spécialisation.

2.1 Deux types de relations

2.1.1 Relations d'équivalence

Une relation d'équivalence *is-equivalent* est un lien entre un élément d'une taxonomie source, $\mathbf{T_{source}}$, et un élément d'une taxonomie cible, $\mathbf{T_{Cible}}$, dont les noms sont similaires. Cette similarité recouvre des réalités variées. Il s'agit tout d'abord de relier des termes dont les noms sont rigoureusement identiques syntaxiquement. En effet, les taxonomies auxquelles nous nous intéressons sont spécifiques à des domaines d'application ; il n'existe

[1] Toutes les techniques présentées exploitent la structure de modèles. Parmi celles-ci, l'une exploite conjointement la structure et les relations sémantiques de WordNet. C'est une technique à la fois structurelle et sémantique.

que très peu d'homonymes. Il s'agit, par ailleurs, de relier des termes dont les noms sont des expressions composées de mots qui, bien que n'étant pas toujours ordonnées à l'identique, ont la même signification. Il en est ainsi de *Pork sausage (liver)* et *Pork liver sausage*. *Liver* est ici un qualificatif qui peut être soit placé devant le nom qu'il caractérise ou après, en apparaissant entre parenthèses.

2.1.2 Relations de spécialisation

Les relations de spécialisation sont les liens usuels *is-a* sous-classe/super-classe. Quand ils relient un élément de la taxonomie source à un super-élément de la taxonomie cible, le degré de généralité du lien est supposé être le même que dans le lien *is-a* reliant ce super-élément à d'autres sous-éléments dans la taxonomie cible. Ainsi, *Asparagus* de la taxonomie source pourra être relié à *Fresh fruit and vegetables* de la taxonomie cible tout comme *Carrots*, un autre terme de celle-ci.

2.2 Un assemblage de techniques

La découverte de mappings repose sur des techniques variées : terminologiques, structurelles et sémantiques. Ces différentes techniques sont composées de façon à rendre le processus de génération des mappings le plus efficace possible (Kefi et al. (2006)). Les techniques terminologiques sont appliquées en priorité. Elles permettent de générer les mappings les plus probables en exploitant toute la richesse des labels des concepts. Les techniques structurelles et sémantiques permettent de trouver des mappings supplémentaires, potentiellement vrais, lorsque l'exploitation des chaînes de caractères ne suffit pas (cf. FIG. 1). L'avantage d'une telle approche est de fournir une catégorisation des mappings suivant la façon dont ils ont été obtenus. Ceci est important aux yeux de l'expert puisque chaque ensemble de mappings n'a pas la même vraisemblance. Une telle catégorisation peut accélérer le processus de validation.

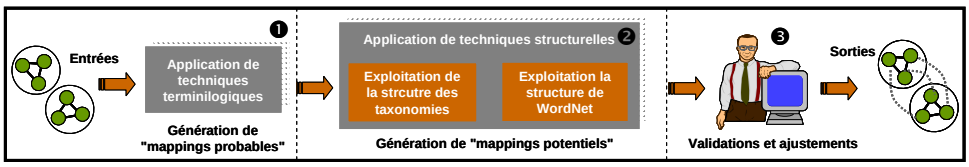


FIG. 1 - *Processus général d'alignement de taxonomies*

3 Exploitation de la structure des représentations

Les trois techniques présentées dans cette section sont utilisées quand des mappings probables n'ont pu être identifiés. Elles exploitent la structure de différentes représentations. La structure de représentation des connaissances la plus riche est supposée être celle de la taxonomie cible et la première technique appliquée s'appuie sur elle. En second, nous proposons d'utiliser une ressource externe, WordNet (Miller (1995)), et d'exploiter sa structure et ses relations sémantiques. Enfin, dans un dernier temps, nous proposons d'exploiter la structure de la taxonomie source combinée à celle de la taxonomie cible, sachant que la taxonomie de la source peut être très peu structurée.

L'objectif est de rattacher un élément e_s de la taxonomie source à un élément e_c de la taxonomie cible. Les mappings générés sont essentiellement des mappings de spécialisation. Ils s'appuient sur le calcul préalable de la similarité de cet élément e_s à tous les termes de la taxonomie cible. La mesure de similarité utilisée est celle de Lin (Lin (1998)) qui compare les chaînes de caractères. Elle a été adaptée pour prendre en compte l'importance des mots dans les expressions.

3.1 L'exploitation de la structure de la taxonomie cible

Dans cette première technique, nous travaillons sur *MC*, l'ensemble des termes candidats à un mapping avec l'élément e_S. Ces termes candidats ont été identifiés à partir du calcul de similarité. Ce sont les termes de $\mathbf{T_{Cible}}$ dont le nom est inclus dans le nom de e_S (INC) ou ceux qui ont une forte similarité avec e_S (seuls les 3 éléments les plus similaires sont retenus). Lorsqu'il n'a pas été possible de déduire un mapping probable avec l'un de ces éléments, l'idée consiste à exploiter leur position dans $\mathbf{T_{Cible}}$. Le sous-graphe représentant les éléments de *MC* au sein de $\mathbf{T_{Cible}}$ est analysé. Dans le meilleur des cas, si tous les éléments de *MC* ont le même père dans $\mathbf{T_{Cible}}$, l'élément e_S considéré a aussi probablement le même père. Dans le cas contraire, si tous les éléments de *MC* n'ont pas le même père, nous cherchons leur plus petit ancêtre commun (Lowest Common Ancestor, LCA). Ainsi, FIG. 2 représente le sous-graphe de $\mathbf{T_{Cible}}$ représentant les éléments de $MC = \{b_1, b_2, b_3\} \cup \{beef\}$ pour e_s = *beef adipose tissue* et INC = {*beef*}. Sur cette figure, l'élément *Fresh meat* représente le plus petit ancêtre commun à tous les éléments de *MC*. Si cet ancêtre commun est un nœud très haut placé dans la taxonomie, il ne sera pas très significatif car trop général.

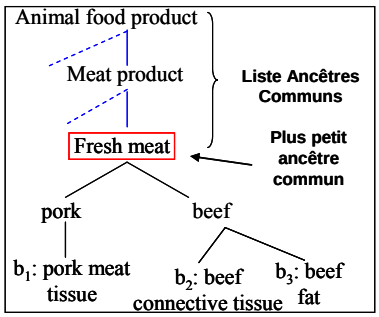


FIG. 2 - *Sous graphe représentant les éléments de MC au sein de* $T_{Cible.}$

Une fois le sous-graphe représentant les éléments de *MC* construit et leur plus petit généralisant au sein de $\mathbf{T_{Cible}}$ identifié, l'idée consiste à rechercher l'élément le plus pertinent qui pourrait être apparié à e_S dans ce sous-graphe. Pour obtenir des suggestions d'appariement les plus pertinentes, nous recherchons des ancêtres partiels, c'est-à-dire des nœuds qui sont les ancêtres d'un sous-ensemble d'éléments de *MC*. Pour chaque sous-ensemble d'éléments de *MC* partageant un ancêtre partiel commun, nous construisons le sous-graphe correspondant qui a pour racine l'ancêtre partiel considéré. Les sous-graphes sont construits comme suit. Nous identifions le plus petit ancêtre commun de chaque paire d'éléments de *MC* et nous étendons l'ensemble considéré élément par élément. Pour un sous-graphe dont la racine est l'ancêtre partiel *Anc*, nous calculons la distance relative *DR(Anc)*

entre les éléments de *MC* correspondant aux nœuds de ce sous-graphe (MC_{Anc}). L'intuition de la formule proposée pour calculer *DR(Anc)* est de tenir compte des trois critères suivants :

- le nombre d'éléments de *MC* dont l'élément *Anc* est l'ancêtre,
- la distance des éléments de *MC* à *Anc* en nombre d'arcs,
- la similarité terminologique des éléments de *MC* à e_S.

$$DR(Anc) = \frac{|MC| * \sum_{ec \in MCAnc} dist(e_c, Anc)}{|MC_{Anc}| * \sum_{ec \in MCAnc} Sim_{LIN\text{-}Like}(e_s, e_c)}$$

Le sous-graphe dont la valeur de *DR(Anc)* est la plus faible est considéré comme le plus pertinent. La distance relative d'un ancêtre partiel *Anc* est d'autant plus faible que :

- *Anc* est l'ancêtre d'un plus grand nombre d'éléments au sein de *MC* ($|MC_{Anc}|$),
- sa distance à ses descendants dans *MC* est faible ($\sum e_c \in MC_{Anc}$ dist(e_c, *Anc*)),
- *Anc* est l'ancêtre d'éléments très similaires de e_S (($\sum_{ec \in MCAnc}$ $Sim_{LIN\text{-}Like}(e_s, e_c)$).

Dans FIG. 2, l'élément *Fresh meat* est le plus petit ancêtre commun des quatre candidats au mapping, avec une distance de 7 ({(1) + (2)} + {(3) + (4)} + { (3) + (5)} +{(3)}) ce qui fait un résultat de (2+2+2+1). Mais un autre sous-graphe de racine *beef* et regroupant trois des candidats au mapping {*beef, beef connective tissue, beef fat*} peut être construit avec une distance égale à 2 ({(4)} + {(5)}). Ainsi, en appliquant la formule de calcul de distance relative, nous obtenons les résultats illustrés FIG. 3.

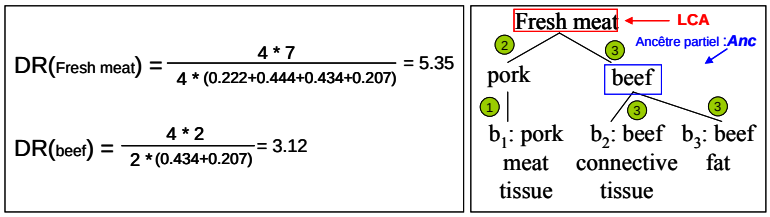


FIG. 3 - *Résultats du calcul de la distance relative pour les éléments Fresh meat et beef.*

Etant donné que la distance relative DR(*beef*) est la plus faible, le sous-graphe le plus pertinent est donc celui dont la racine est *beef*. Dans ce sous-graphe, le nœud qui a la valeur de similarité la plus élevée est noté C_{MaxAnc}. Il est considéré comme le candidat au mapping le plus proche de e_S. Si C_{MaxAnc} appartient à l'ensemble des termes dont le nom est inclus dans celui de e_S, il est considéré comme un père possible de e_S dans $\mathbf{T_{Source}}$. Dans le cas contraire, C_{MaxAnc} est considéré comme un frère possible et son père (qui n'est pas forcément le nœud racine du sous-graphe) est proposé comme un père possible de e_S. Sur l'exemple précédent, le groupement représenté par le sous-graphe dont la racine est *beef* est donc jugé plus pertinent et le nœud de ce sous-graphe qui a la valeur de similarité la plus élevée, C_{MaxAnc} : *beef connective tissue*, est proposé comme un frère de *beef adipose tissue* qui sera relié à l'élément *beef* dans $\mathbf{T_{Cible}}$ par une relation de spécialisation.

3.2 L'exploitation de la structure et des relations sémantiques de WordNet

Les techniques décrites précédemment ne sont pas suffisantes quand les concepts sont sémantiquement proches mais que leur nom est différent. Ainsi, aucune de ces techniques ne permet de rapprocher *cantaloupe* et *watermelon* alors que l'interrogation d'une source

linguistique, telle que WordNet, peut indiquer que ces concepts sont des melons. Dans notre approche, l'utilisation des synonymes de WordNet n'est pas suffisante. Nous proposons de combiner l'exploitation des relations sémantiques de WordNet avec la structure de la hiérarchie de WordNet afin de trouver, pour chaque élément de la taxonomie source, de quels éléments de la taxonomie cible il peut être sémantiquement proche (ceux avec lesquels il partage des généralisants dans WordNet). Cela permet, par exemple, de rapprocher *cantaloupe* et *watermelon* qui ne sont pas synonymes mais qui sont deux spécialisations du concept *melon*.

L'utilisation de WordNet s'effectue en deux étapes. La première étape consiste à construire un sous-arbre (appelé S_{WN}) à partir de l'ensemble des généralisants dans WordNet de chacun des éléments des deux taxonomies $\mathbf{T_{Cible}}$ et $\mathbf{T_{source}}$ que l'on cherche à rapprocher. La seconde étape consiste à utiliser une mesure de similarité (Wu et Palmer (1994)) pour identifier, au sein de la taxonomie commune S_{WN}, l'élément de $\mathbf{T_{Cible}}$ le plus proche de l'élément de $\mathbf{T_{source}}$ considéré,. Pour construire S_{WN}, nous interrogeons WordNet pour chaque élément de chacune des deux taxonomies. Pour chaque élément, et pour chacun de ses sens (chaque synset auquel il appartient), nous extrayons l'ensemble de ses généralisants jusqu'à atteindre le concept le plus général pour l'application considérée. Par exemple, le résultat de la recherche sur le terme *cantaloupe* donne les deux ensembles de généralisants suivants qui correspondent à deux sens différents du terme.

- Sens 1 : *Cantaloupe → sweet melon → melon → gourd → plant → organism → Living thing*
- Sens 2 : *Cantaloupe → sweet melon → melon → edible fruit → green goods → food*

Seul est conservé le sens qui contient le concept racine de l'application étudiée (sens 2 dans l'exemple car il contient *food*). Les généralisants de l'élément sont intégrés pour construire le sous-arbre de WordNet pertinent pour l'application, i.e., dont la racine est le terme le plus général de l'application. Les feuilles sont les termes issus des deux taxonomies initiales (dans des ovales sur FIG. 4). Les généralisants intermédiaires sont extraits de WordNet mais ils peuvent également appartenir à l'une des deux taxonomies.

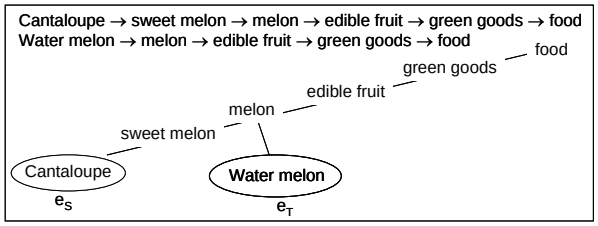


FIG. 4 - *Un sous-graphe de S_{WN} reliant cantaloupe et water melon dont la racine est food.*

Une fois S_{WN} construit, nous évaluons la similarité sémantique entre deux nœuds c_1 et c_2 de l'arbre en utilisant la mesure de similarité définie dans (Wu et Palmer (1994)). Wu et Palmer définissent la similarité sémantique entre deux concepts c_1 et c_2 en fonction de leur profondeur, $prof(c_i)$, $i \in [1,2]$, i.e. leur distance à la racine en nombre d'arcs et en fonction de la profondeur de leur plus petit ancêtre commun (LCA) :

$$Sim_{W\&P}(c_1, c_2) = \frac{2 * prof(LCA(c_1, c_2))}{prof(c_1) + prof(c_2)}$$

Cette mesure est plus précise qu'une mesure basée sur une simple distance. En effet, plus la profondeur du LCA de deux concepts est importante, plus les deux concepts partagent de

caractéristiques communes et plus ils sont proches. Ainsi, sur l'exemple FIG. 5, si l'on recherche dans S_{WN} le terme le plus proche de l'élément de la taxonomie source e_S parmi les nœuds de T_{cible}, X_1, X_2, Y et Z, les similarités calculées par la mesure sont, par ordre décroissant : $sim_{W\&P}(e_S,X_1) > sim_{W\&P}(e_S,Y) > sim_{W\&P}(e_S,X_2) > sim_{W\&P}(e_S,Z)$.

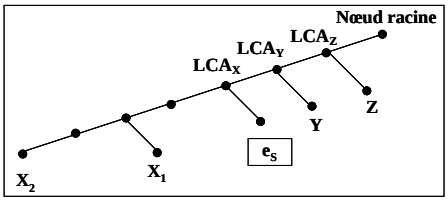


FIG. 5 – *Exemple de taxonomie S_{WN}.*

Du fait du mode de calcul de la mesure, la similarité est plus grande entre e_s et un de ses nœuds frères ou un des descendants proches de ce frère, qu'entre e_s et son grand-père, et ce, jusqu'à une certaine profondeur *p* du descendant, qu'il est possible de déterminer a priori, pour un élément e_s étant donnée sa profondeur. Notre stratégie de parcours de S_{WN} est basée sur cette propriété. Elle consiste tout d'abord à tester si le père P de l'élément e_s dans S_{WN} est un élément de la taxonomie cible, $\mathbf{T_{Cible}}$. Si c'est le cas, il est l'élément le plus proche cherché. Dans le cas contraire, il s'agit de rechercher un élément de $\mathbf{T_{Cible}}$ parmi les descendants du père de e_s jusqu'à la profondeur *p*. Cette profondeur atteinte, si aucun élément de $\mathbf{T_{Cible}}$ n'a été trouvé, il est nécessaire de tester alternativement les descendants de profondeur supérieur à *p* et les ascendants (puis les descendants des ascendants) du père. Cette stratégie permet de limiter le nombre de calculs de similarité à effectuer.

Sur l'exemple FIG. 5, le père de e_s n'appartient pas à $\mathbf{T_{Cible}}$, l'élément e_s est à la profondeur 4, son grand-père à la profondeur 2, la profondeur limite calculée pour e_s (de profondeur 4) est de 5. Parmi les descendants du père de e_s, aucun élément de profondeur inférieure ou égale à 5 n'appartient à $\mathbf{T_{Cible}}$. Après avoir vérifié que le grand-père de e_s n'appartenait pas à $\mathbf{T_{Cible}}$, on étudie les descendants du père de profondeur 6. X_1 appartenant à $\mathbf{T_{Cible}}$, il est donc l'élément recherché.

Cette technique permet d'établir des mises en correspondance entre des éléments connus de Wordnet, des expressions composées en général de peu de mots. Il s'agit de suggestions d'appariement faites à l'expert sans préciser le type de relation exacte reliant les concepts. L'expert peut les accepter en les validant ou les refuser.

3.3 L'exploitation conjointe de la structure des deux taxonomies

A cette étape du processus de découverte des mappings, nous proposons d'appliquer des heuristiques inspirées de celles proposées dans (Melnik et al. (2002), Madhavan et al. (2001), Doan et al. (2002). L'idée de base est de faire une proposition de mise en correspondance à partir de l'étude des mappings des nœuds voisins déjà établis. Ainsi, dans l'exemple représenté FIG. 6, le problème est de trouver un mapping pour le terme *Apple Cider with 12-14 Brix,* fils du concept *Fruit and fruit products* dans la taxonomie source. Sachant que la majorité des descendants du concept *Fruit and fruit products* dans la taxonomie source ont été reliés au concept *Drink* ou à l'une de ses spécialisations dans la taxonomie cible, il est vraisemblable que le terme *Apple Cider with 12-14 Brix* puisse également être rattaché à un élément du sous-arbre de racine *Drink*. Le problème est donc de déterminer, d'une part, de quel nœud général de la taxonomie cible le concept à apparier est le plus proche, puis s'il

doit être rattaché à ce nœud général (*Drink* dans FIG. 6) ou à un nœud plus spécifique (par exemple *Apple juice* dans FIG. 6).

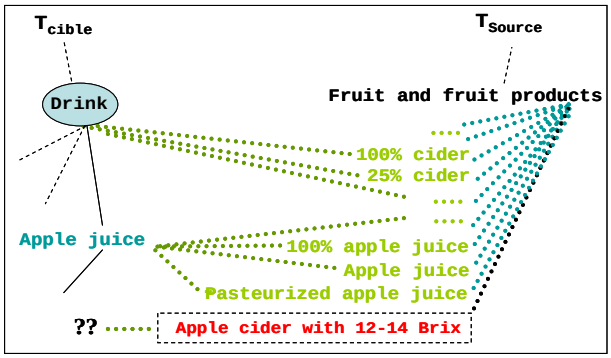


FIG. 6 - *Mappings des frères de Apple cider with 12-14 Brix*

Etant donné un concept e_S feuille de la taxonomie source et ses concepts frères dans cette taxonomie, nous définissons l'ensemble MappingsOfNeighbours (MoN) comme l'ensemble composé des termes de la taxonomie cible auxquels les concepts frères de e_S ont été rattachés par un mapping. Nous mémorisons aussi pour chaque élément de MoN le nombre de mappings établis avec un frère de e_S et nous retenons comme candidats au mapping pour le concept e_S les éléments de MoN intervenant dans au moins deux mappings. Soit CMoN cet ensemble. Nous recherchons ensuite dans la taxonomie cible, les pères des éléments $e_c \in$ CMoN et retenons comme nœuds généraux pertinents, s'ils existent, les nœuds qui sont les pères d'au moins un tiers des éléments de MoN (ou eux-mêmes, par exemple *Drink*). Les éléments de CMoN seront présentés à l'expert regroupés par nœuds généraux pertinents s'ils existent et ordonnés par nombre de mappings établis décroissant.

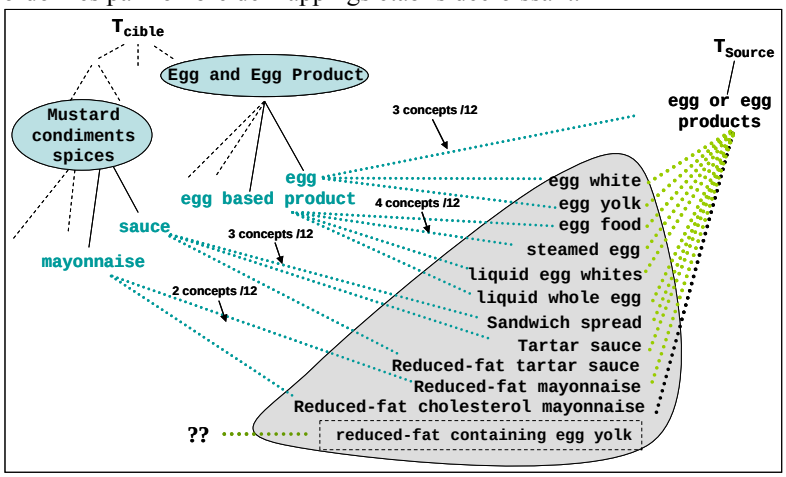


FIG. 7 - *Mappings des noeuds frères de reduced-fat containing egg yolk.*

Sur l'exemple FIG. 7, 3 des 12 frères du concept *reduced-fat containing egg yolk* ont été appariés avec *egg*, 3 avec *egg based product*, 3 avec le concept *sauce* et 2 avec *mayonnaise*. Dans la taxonomie source, *Egg* et *egg based product* ont pour père commun *Egg and Egg*

product. *Sauce* et *mayonnaise* ont pour père *Mustard, condiment, spices*. Le système fait deux propositions de mappings, chacune consistant à suggérer une mise en correspondance soit avec le nœud général (i.e. *Egg and Egg product* ou *Mustard, condiment, spices)*, soit avec l'une de ses deux spécialisations, mais sans toutefois être en mesure de choisir ou de préciser le type de relation exacte reliant le concept.

En revanche, sur l'exemple FIG. 6, sur les 11 frères de *Apple Cider with 12-14 Brix*, 5 ont été mis en correspondance avec *Apple juice*, 3 avec *Drink* et 3 avec *Fruit*. Dans la recherche des nœuds généraux pertinents, le concept *Fresh fruits and vegetables* père de *Fruit* n'est pas retenu car il ne recouvre que 3 sur 11 des éléments adjacents. En revanche, le concept *Drink* est retenu comme car il recouvre 8 sur 11 des éléments de MoN, 3 par lui-même et 5 en tant que père du concept *Apple juice.* Le système ne proposera donc qu'une direction d'appariement, pour le sous-arbre *Drink* et sa spécialisation *Apple juice.*

Si e_S n'est pas un élément feuille de la taxonomie source, la recherche des éléments de MoN est faite sur les fils de e_S, ses frères et leurs descendants. Un mapping sera proposé avec un élément $e_c \in$ CMoN si e_c est le plus petit généralisant commun à l'ensemble des termes de CMoN. Cette technique d'appariement donne des propositions pertinentes lorsque des mappings ont déjà été trouvés pour de nombreux concepts frères ou fils, même si les schémas à apparier sont assez éloignés structurellement.

4 Expérimentations

Un prototype, TaxoMap, a été implémenté en java. Il a permis de réaliser des expérimentations sur les taxonomies Sym'Previus et Com'Base dans le domaine du risque alimentaire[2]. Sym'Previus est l'ontologie cible, Com'Base est l'ontologie source.

La taxonomie de Sym'Previus est composée de 460 termes organisés en 7 niveaux. Com'base est une taxonomie de 172 termes organisée en 2 niveaux, le premier niveau comprenant uniquement 12 termes. Les deux taxonomies ont des parties qui se recouvrent mais ne sont pas structurées de la même façon : une branche peut être détaillée dans l'une et pas dans l'autre ou inversement. La différence de structure s'explique par le fait qu'elles traduisent des points de vue distincts. Enfin, le niveau de granularité n'est pas le même et le nombre de termes représentés est alors différent.

Afin d'évaluer les résultats obtenus en sortie de TaxoMap, un expert en microbiologie a défini manuellement les mappings entre les deux taxonomies, soit pour 172 éléments (nombre d'éléments de Com'Base). L'exécution de Taxomap a permis de générer 96 mappings probables. Ces mappings ont l'avantage d'avoir une précision très importante, supérieure à 90 % (Kefi et al. (2006)) au détriment du rappel (56 %). Les techniques structurelles sont alors très utiles pour compléter les résultats, même si les mappings générés sont moins sûrs. Leur précision, au niveau des expérimentations réalisées, confirme l'ordre d'application proposé dans notre méthode. Les résultats obtenus sont synthétisés dans TAB. 1.

Les mappings non trouvés par la technique exploitant la structure de la taxonomie cible correspondent tous à des mappings générés automatiquement mais qui ne sont pas pertinents. Ils s'expliquent en partie par la difficulté à traiter les labels comportant beaucoup de mots et surtout à distinguer, parmi ces mots, ceux qui font référence au concept sous-jacent et ceux

[2] Ce travail a été réalisé dans le cadre du projet RNTL e.dot (2003-2005).

qui ne font que le caractériser. En effet, cette technique exploite la structure de $\mathbf{T_{Cible}}$, mais elle repose aussi beaucoup sur les calculs de similarité entre éléments, ces calculs étant principalement basés sur des comparaisons de chaînes de caractères. Ce problème de traitement de labels mis à part, cette technique s'est montrée fort utile dans 28 cas sur 42, soit 66 % des cas.

Techniques structurelles	Nombre d'éléments de T_{Source} étudiés	Nombre de mappings suggérés	Nombre de mappings confirmés	Nombre de mappings non trouvés	Précision
Exploitation de la structure de T_{Cible}	42	42	28	14	66 %
Exploitation de la structure de WordNet	14	10	9	5	64 %
Exploitation de la structure de T_S et T_C	5	3	3	2	60 %

TAB. 1 – *Nombre de mappings trouvés par technique employée.*

La technique basée sur WordNet s'est révélée être tout à fait complémentaire des techniques précédemment appliquées. 9 mappings ont pu être trouvés, par exemple *100% cider* a pu être apparié avec *Drink, Cantaloupe* avec *Melon* et *Frankfurter* avec *Sausage*. Les 5 éléments pour lesquels nous n'avons aucun mapping à la fin de l'application de cette technique correspondent soit à des suggestions fausses (*Phosphate buffer* est mis en correspondance avec *Drink* alors que l'expert proposait que ce soit un fils de *Culture Medium*), soit à des éléments pour lesquels aucune suggestion n'a été faite : cas d'acronymes (ex : *TSB*) ou de mots techniques non reconnus par WordNet (ex : *Egyptian Kofta*).
Nous avons appliqué la dernière technique sur les 5 éléments restant à apparier : *Phosphate buffer, Egyptian Kofta, TSB, Tampeh, Pecan nuts*. Deux éléments, *TSB* et *Phosphate buffer*, n'ont pu être mis en correspondance. Ils sont frères l'un de l'autre. Ils ont un seul autre frère commun pour lequel un mapping a été trouvé avec *Culture Medium*. *Culture Medium* serait pertinent (d'après l'expert) pour *Phosphate buffer* et *TSB* mais, en réalité, aucune proposition n'est faite car la technique exige que l'élément proposé au mapping soit mis en correspondance avec au moins deux des frères de l'élément étudié. Pour les trois autres éléments, l'étude des mappings de leurs frères a permis de faire des propositions qui se sont révélées pertinentes. Par exemple, la suggestion est faite d'apparier *Tampeh* avec *Vegetable* (l'expert proposait *Fresh fruit and vegetables*). De même, *Pecan nuts* est apparié avec *Vegetable* (l'expert l'avait apparié avec un de ses fils dans Sym'Previus). Pour *Egyptian Kofta*, le système propose 3 directions d'appariement, l'une avec *Fresh meat*, une seconde avec *Meat-based product* et une troisième avec *Poultry*. La seconde correspond au mapping de l'expert.

5 Travaux proches et discussion

Il existe aujourd'hui de nombreux travaux qui visent à automatiser la génération de mappings. Une synthèse des techniques utilisées est présentée dans Rahm et Bernstein (2001) et Shvaiko et Euzenat (2004). Les techniques sont variées. Nous nous limiterons, dans cette section, aux travaux mettant en oeuvre des techniques structurelles, centrales dans ce papier.

Les techniques structurelles consistent à exploiter la structure des schémas comparés, souvent représentés sous forme de graphes. Les algorithmes mettant en œuvre ces techniques implémentent diverses heuristiques. Une heuristique consiste, par exemple, à considérer que des éléments de deux schémas distincts sont similaires si leurs sous-concepts directs, et/ou leurs sur-concepts directs et/ou leurs concepts frères sont similaires (Do et Rahm (2001), Noy et Musen (2001) (Thanh Le et al. 2004)).. Ces techniques structurelles peuvent être basées sur la notion de point fixe (Melnik et a. (2002)). Dans S-Match (Giunchiglia et Shvaiko (2004)), le problème de matching est vu comme un problème de satisfiabilité d'un ensemble de formules du calcul propositionel. Les graphes et les correspondances à tester sont traduits en formules de la logique propositionnelle en considérant la position des concepts dans le graphe et non seulement leur nom.

Notre travail se distingue des travaux précédemment cités du fait de la dissymétrie dans la structure des taxonomies à rapprocher. La recherche de structures similaires est impossible. Nous proposons donc d'exploiter les données structurelles différemment. La structure de la taxonomie cible, prise isolément, est tout d'abord utilisée pour déterminer le type du concept avec lequel un mapping peut être établi avec un élément e_S de la taxonomie source. Ceci s'effectue en localisant la partie de la taxonomie cible contenant vraisemblablement l'élément avec lequel un mapping pourra être établi. Cette localisation exploite la structure de la taxonomie cible mais s'appuie également sur les calculs de similarité préalablement effectués, basés sur des aspects purement terminologiques. Ne pouvant pas exploiter la structure de la taxonomie source qui, dans notre contexte de travail, est supposée peu structurée, une autre façon d'exploiter la structure est d'avoir recours à des ressources autres que les taxonomies comparées et d'exploiter la structure de ces ressources. C'est ce que nous faisons lorsque nous utilisons WordNet. Enfin, nous proposons une dernière technique basée sur l'exploitation de la structure des deux taxonomies, compte tenu de l'existence d'une dissymétrie. L'idée est d'étudier les mappings préalablement découverts et d'en déduire, compte tenu de la localisation des éléments mis en correspondance dans chacune des taxonomies, des mappings possibles.

6 Conclusion

Ce papier décrit trois techniques structurelles d'alignement de taxonomies supposées dissymétriques du point de vue de leur structure. Le contexte de travail rend impossible la recherche de similarités structurelles. Ainsi, nous proposons d'autres moyens d'exploiter ce type d'information : exploitation de la structure de la taxonomie cible uniquement, pour localiser la partie au sein de laquelle un mapping est possible, exploitation de la structure de Wordnet ou encore exploitation conjointe des informations structurelles des deux taxonomies combinée avec l'étude de mappings préalablement découverts. Ces techniques sont originales dans la mesure où elles se distinguent de la recherche d'une similarité structurelle entre les modèles à aligner. Elles sont applicables pour faire des suggestions de mappings au concepteur. Ces mappings n'ont pas la même vraisemblance que ceux générés par application de techniques terminologiques, c'est pourquoi notre méthode propose d'appliquer les techniques structurelles après les techniques terminologiques. Il s'agit néanmoins d'un bon complément comme les expérimentations l'ont montré.

Références

Doan, A., J. Madhavan, P. Domingos, A. Halevy (2002). *Learnig to map between ontologies on the semantic web*. WWW, pp. 662-673, N.Y., USA, ACM Press.

Do, H. H., E. Rahm (2001). *COMA – A system for flexible combination of schema matching approaches.* VLDB, pp. 610-621.

Giunchiglia, F., P. Shvaiko (2004). *Semantic Matching*. The Knowledge Engineering review, 18(3):265-280.

Kefi, H.,B. Safar, C. Reynaud (2006). *Alignement de taxonomies pour l'interrogation de sources d'information hétérogènes*. RFIA, Tours.

Lin, D (1998). *An Information-Theoretic Definition of Similarity*. ICML, Madison, pp. 296-304.

Madhavan, J., P. A. Bernstein, E. Rahm (2001). *Generic matching with Cupid*. VLDB Journal, pp. 49-58.

Miller, G. A. (1995). *WordNet: A lexical Database for English*. Communications Of the ACM, Vol. 38, n°11, P. 39-45.

Noy, N. F., M. A. Musen (2001). *Anchor-Prompt: Using non-local context for semantic matching*. Workshop on Ontologies and Information Sharing at IJCAI-2001, Seattle, WA.

Melnik, S., H. Garcia-Molina, E. Rahm (2002). *Similarity Flooding*: *A versatile Graph Marching Algorithm and its application to schema matching*. ICDE, San Jose CA.

Rahm, E., P. Bernstein (2001). *A survey of approaches to automatic schema matching*. VLDB Journal: Very Large Data Bases, 10(4):334-350.

Shvaiko, P., J. Euzenat (2004). *A survey of Schema-based Matching Approaches*. Technical report DIT-04-087, Informatica e Telecomunicazioni, Université de Trento.

Thanh Le, B., R. Dieng-Kuntz, F. Gandon (2004). *On Ontology Matching Problems for building a Corporate Semantic Web in a Multi-Communities Organization*. ICEIS (4), pp. 236-243.

Wu, Z., M. Palmer (1994). *Verb semantics and lexical selection.* Computational Linguistics, Las cruces, pp. 133-138.

Summary

This paper deals with generation of mappings in order to align Web taxonomies. The objective is to allow a uniform access to documents in a given application domain. Retrieval of documents is based on taxonomies. We propose to align the taxonomy of a Web portal with the ontology of external documents. That way, the number of accessible documents from the portal can increase without any change in its query interface. This paper presents structural techniques composing the alignment method when dealing with taxonomies which are very different from a structure point of view. A prototype, TaxoMap, has been implemented. It has supported experiments which we present.