Extraction Contextuelle de Concepts Ontologiques pour le Web Sémantique

Lobna Karoui, Marie-Aude Aufaure, Nacéra Bennacer Seghouani

Extraction Contextuelle de Concepts Ontologiques pour le Web Sémantique

Lobna Karoui, Marie-Aude Aufaure, Nacera Bennacer

Supélec, Plateau de Moulon 3 rue Joliot Curie
91192 Gif-sur-Yvette cedex, France
Lobna.Karoui@supelec.fr

Résumé : De nombreux travaux de recherche, s'intéressant à l'annotation, l'intégration des données, les services web, etc. reposent sur les ontologies. Le développement de ces applications dépend de la richesse conceptuelle des ontologies. Dans cet article, nous présentons l'extraction des concepts ontologiques à partir de documents HTML. Afin d'améliorer ce processus, nous proposons un algorithme de clustering hiérarchique non supervisé intitulé « Extraction de Concepts Ontologiques » (ECO) ; celui-ci utilise d'une façon incrémentale l'algorithme de partitionnement Kmeans et est guidé par un contexte structurel. Ce dernier exploite la structure HTML ainsi que la position du mot afin d'optimiser la pondération de chaque terme ainsi que la sélection du co-occurrent le plus proche sémantiquement. Guidé par ce contexte, notre algorithme adopte un processus incrémental assurant un raffinement successif des contextes de chaque mot. Il offre, également, le choix entre une exécution entièrement automatique ou interactive. Nous avons expérimenté notre proposition sur un corpus du domaine du tourisme en français. Les résultats ont montré que notre algorithme améliore la qualité conceptuelle ainsi que la pertinence des concepts ontologiques extraits.

Mots-clés : Ingénierie des connaissances, Ontologies, Contexte.

1 Introduction

De nombreux travaux de recherche s'intéressent à l'annotation, l'intégration des données, les services web, la recherche d'information, etc. et reposent sur les ontologies. Le développement du web sémantique ainsi que le succès de ces applications dépendent de la richesse et de la qualité conceptuelle des ontologies. Le développement d'approches pour extraire des concepts du domaine et leurs relations constitue un point crucial pour cette tâche complexe que représente la construction d'ontologies. Dans ce travail, nous nous intéressons à l'extraction des concepts ontologiques à partir de documents HTML. La connaissance extraite du web permet de l'enrichir par une sémantique. Afin d'améliorer ce processus, nous proposons un algorithme de clustering hiérarchique non supervisé intitulé « Extraction de Concepts Ontologiques » et noté « ECO » qui utilise d'une façon incrémentale l'algorithme de partitionnement Kmeans et qui est guidé par un contexte structurel. Notre définition contextuelle est fondée sur la structure HTML ainsi que la position du mot dans le document. Ce contexte est déduit des multiples analyses appartenant à la partie

prétraitement de notre système (Karoui et al, 2006). Notre représentation contextuelle explicite intitulée « Hiérarchie Contextuelle » (H.C) guide le clustering afin de délimiter le contexte de chaque mot en améliorant sa pondération, la similarité entre les paires de mots et la sélection des coocurrents sémantiquement proches. En instaurant un mécanisme incrémental et en divisant récursivement les classes, l'algorithme ECO raffine le contexte de chaque classe de mots et améliore la qualité conceptuelle des clusters finaux et par conséquence des concepts extraits. Notre algorithme offre également le choix entre une exécution automatique ou interactive. Nous avons expérimenté cet algorithme de clustering hiérarchique et contextuel avec des documents HTML du domaine du tourisme et en langue française. Les résultats obtenus ont montré que la définition d'un contexte approprié ainsi que le raffinement successif des contextes des classes par notre algorithme ECO améliorent considérablement la pertinence et la qualité conceptuelle des concepts ontologiques extraits en comparaison avec l'algorithme Kmeans.

Dans la section suivante nous présentons un état de l'art. Dans la section 3, nous proposons un algorithme d'extraction de concepts ontologiques (ECO) guidé par une définition contextuelle structurelle. Dans la section 4, nous expérimentons notre algorithme en le comparant à Kmeans. Dans la section 5, nous concluons.

2 Etat de L'art

Nombreuses sont les recherches qui ont utilisé la structure HTML. Par exemple, Buyukkoten (Buyukkoten et al, 2001) discute une méthode d'extraction du contenu des documents HTML en transformant une page Web en une hiérarchie d'unités textuelles sémantiques. Ces unités sont définies en analysant les caractéristiques syntaxiques d'un document HTML comme le texte contenu dans les balises (<p>, <frame>, etc.). De même, Kiyota et Kurohashi (Kiyota et Kurohashi, 2001) présentent un extracteur de phrases et un générateur de résumés basés sur une analyse syntaxique, la méthode du tf.idf de Salton et la structure HTML. Ils considèrent que les mots-clés appartenant aux titres et aux sous-titres sont plus importants que les mots-clés qui apparaissent dans les listes et les tables, c'est pourquoi ils leur associent un poids plus important. L'approche de (Cai et al, 2004) a pour but d'améliorer la pertinence de la recherche documentaire dans le Web.

Par ailleurs, des méthodologies et des techniques d'automatisation de la construction d'ontologies se sont développées. Dans (Faure et al, 1998), les auteurs présentent ASIUM, un système d'apprentissage à partir de textes techniques. Des clusters de base sont formés par des termes apparaissant avec le même verbe et avec le même rôle syntaxique ou la même préposition fournie par un analyseur syntaxique. Les auteurs ayant développé une méthode de clustering basée sur ces idées obtiennent en sortie une ontologie avec des relations taxonomiques. Le système WebOntEx (Hahn et Elmasri, 00) a pour objectif d'extraire semi-automatiquement des ontologies en analysant les pages web appartenant au même domaine. L'extraction des connaissances est basée sur les balises HTML (<b>, <h1>), les balises de lemmatisation (verbe, nom) et les balises conceptuelles (entité, attribut) en utilisant

WordNet et la programmation logique inductive. L'un des problèmes majeur de cette méthode est la lourde tâche manuelle à réaliser au cours de laquelle l'utilisateur devra développer le cœur de l'ontologie et extraire les patterns génériques à partir des pages Web. OntoMiner (Davulcu et al, 98) analyse des ensembles de sites web d'un domaine spécifique et génère une taxonomie de concepts particuliers ainsi que leurs instances. Cet outil utilise les régularités HTML des documents pour générer une structure hiérarchique codée en XML. Maedche et Staab (2001) proposent un environnement d'apprentissage d'ontologies (Text-To-Onto) basé sur une architecture générale de découverte de structures conceptuelles à partir de différentes sources. Cet environnement possède une librairie de méthodes d'apprentissage et des outils linguistiques pour extraire des concepts et leurs relations. DODDLE II (Sugiura, 2004) est un environnement de développement d'ontologies permettant d'extraire des relations taxonomiques en utilisant à la fois les termes du domaine et WordNet. Pour découvrir des relations non taxonomiques, les auteurs retrouvent les collocations d'un ensemble de 4 termes d'où la notion de collocation du domaine. Pour obtenir les relations non taxonomiques, ils sélectionnent les paires de concepts dont le produit de leurs vecteurs a dépassé un seuil fixé par l'utilisateur et utilisent également les règles d'associations afin d'extraire ces relations. ASIUM est un système qui dépend de la structure des documents analysés. Quant à Text-to-Onto, l'une de ses faiblesses est l'existence de bruit. En effet, certains termes, n'ayant pas de relations sémantiques et existants dans la même classe, peuvent nuire au processus d'interprétation de la classe et induire une perturbation au sein du groupe de mots. Pour DODDLE, il nécessite l'existence du vocabulaire du corpus à étudier dans WordNet à défaut l'opération de matching est impossible à réaliser. Cette dépendance à une connaissance à priori, pareille pour WebOntoEx, tout en étant un atout pour extraire une connaissance plus pertinente, représente une limite. Concernant la notion de contexte, ces systèmes focalisent leur analyse sur le contexte syntaxique d'un terme (limite l'existence d'un terme dans une phrase) à l'exception de DODDLE qui considère que le contexte d'un groupe de 4 mots est l'ensemble des 4 mots qui le précède directement. Pour le système OntoMiner, les auteurs se basent sur les régularités existantes dans les documents HTML alors qu'en réalité la majorité des documents Web manquent de régularités HTML. Concernant notre approche, notre analyse ne se limite pas à un contexte syntaxique ou à un ensemble de mots suivant ou précédant le mot en question mais présente une définition contextuelle plus riche (section 3.1). Elle applique un traitement incrémental en vue d'agir sur l'existence de bruit dans les classes de mots tout en étant indépendante de toutes connaissances à priori.

3 Algorithme d'extraction des concepts ontologiques

Dans cette section, nous définissons le contexte structurel. Ensuite, guidé par ce contexte, nous présentons notre algorithme d'extraction de concepts ontologiques.

3.1 Définition du Contexte Structurel

La question fondamentale qui se pose dans l'extraction des concepts en adoptant une approche statistique est la suivante : « comment attribuer une pondération à un terme reflétant son importance dans le domaine et mesurant la pertinence de ses co-occurrents dans la relation les liant sémantiquement» ?

L'existence d'une relation structurelle entre les éléments HTML peut révéler une relation sémantique implicite entre les termes associés. Le fait d'instancier le contexte par rapport au lien structurel permet de mieux cerner et révéler les concepts relatifs aux termes apparaissant dans, par exemple, les balises <h1> → <p> ; <caption> → <td> (titre d'un tableau → cellule d'un tableau) ; <TITLE_URL> (titre d'un lien hypertexte) → les titres d'une partie d'un document ; <TITLE_URL> → les titres du document référencés ; etc.

Nous distinguons deux types de liens structurels : un lien physique qui dépend de la structure du document HTML (entre la balise <h1> et la balise <p> associée) et un lien logique qui n'est pas visuel puisque les éléments ne sont pas nécessairement consécutifs (entre <TITLE_URL> et les titres du document référencé par exemple). Pour caractériser les liens entre les balises, nous avons défini deux notions : la « hiérarchie contextuelle » (H.C.) basée sur les balises HTML et la « cooccurrence par liaison ». Une hiérarchie contextuelle est une hiérarchie de balises. Elle illustre les relations possibles dans les documents HTML et entre eux. La cooccurrence est définie par le fait d'avoir deux mots dans le même contexte (paragraphe, texte, etc.). Dans notre étude, le contexte est variable et il est déduit de la hiérarchie contextuelle. En respectant la structure de H.C, nous établissons des liaisons entre les termes si :
(1) les termes sont encadrés par la même balise bloc (Table 1. Exemple 1), dans ce cas, on parle de *cooccurrence par voisinage* et le contexte est fixé à la balise elle même (<H1>) ; (2) les termes sont encadrés par des balises qui à leur tour sont reliées par un lien physique ou logique schématisé dans la hiérarchie contextuelle. Dans ce deuxième cas (Table 1. Exemple 2), nous parlons de *cooccurrence par liaison* (balises non consécutives) et le contexte est l'association des deux balises (<title> + <h1>).

Table 1. Exemples de contextes d'utilisation

Exemple 1	Exemple 2
<H1> événement maritime </H1>	<TITLE> Catégories de logements et d'établissements d'hébergement </TITLE> <KEYWORDS> *** </KEYWORDS> <HYPERLINK> *** <TITLE_URL> *** <H1> Résidences de tourisme </H1> <P> un établissement touristique ayant certaines caractéristiques communes avec un hôtel…… </P>

La cooccurrence par voisinage permet de retrouver les cooccurrents d'un mot dans un seul contexte (phrase, balise, etc.) alors que la cooccurrence par liaison est une cooccurrence pour laquelle les cooccurrents d'un mot dépendent à la fois de la position du mot dans un contexte et de la relation de ce contexte avec les autres contextes existants. Ainsi, le contexte est générique et sera instancié selon l'appartenance du terme à une balise (prendra des valeurs différentes par exemple dans un cas où le contexte est une balise comme <B> et dans un autre cas où le contexte est l'association de <H1> et <Title>, etc.). Dans l'exemple 2 (Table 1), si

nous considérons le terme « logement », en respectant la liaison logique existante entre <TITLE> et <H1> (figurant dans H.C), nous trouvons les co-occurrents de « logement » dans la réunion des deux balises bien qu'elles soient éloignées. Ce second type de liaison (logique) est sémantique puisqu'un titre de document aura une relation avec les sous titres du même document. Si nous considérons le terme « résidences », nous retrouvons ses co-occurrents dans l'association des deux balises <h1> et <p>, reliées par un lien physique conformément à H.C. Ces deux balises représentent le contexte instancié pour le terme « résidences» en respectant son appartenance à <h1>. Si ce même terme existe dans une autre balise, le contexte sera différent et générera une nouvelle instance.

Dans les cas ou nous ne retrouvons ni un lien logique ni un lien physique entre deux balises, nous considérons la balise seule en tant qu'unité contextuelle et nous appliquons la cooccurrence par voisinage dans cette même balise html. Dans l'exemple 1 (Table 1), nous avons comme co-occurrent de « événement » le terme « maritime » dans la balise <h1>. Cette balise représente le contexte du terme « événement ».

L'application du contexte générique en relation avec la structure html et les liens sémantiques existants entre les balises permet de représenter l'adaptabilité d'un terme dans le corpus et modélise un contexte dynamique. Ainsi, notre modèle contextuel respecte la position d'un terme afin de pouvoir prendre en considération diverses situations dans lesquelles le terme a été cité. Le calcul de pondération d'un terme par rapport à son co-occurrent prendra en compte les différents contextes (instanciés grâce à H.C) dans lesquels le mot apparaît.

La pondération d'un terme est calculée en utilisant l'indice d'équivalence (Michelet, 1988) qui permet d'évaluer la force de lien entre deux termes. C_i : occurrence du terme i / C_{ij}: cooccurrences des deux termes i et j

$$E_{ij} = C^2_{ij} / (C_i \times C_j) \qquad (1)$$

3.2 Principes Algorithmiques

Dans ce qui suit, nous présentons un algorithme de clustering hiérarchique non supervisé intitulé « Extraction de Concepts Ontologiques » (ECO) permettant d'extraire des concepts ontologiques à partir des documents HTML. Il est basé sur une utilisation incrémentale de l'algorithme de partitionnement Kmeans (MacQueen, 1967) et est guidé par les contextes structurels dans lesquels les mots apparaissent. Nous avons choisi d'adapter Kmeans à notre problème puisque c'est un algorithme non supervisé capable de classifier un énorme volume d'objets dans des cours délais d'exécution. Nous avons implémenté quatre mesures de similarités (Nakache et Confais, 2005) dans notre algorithme ECO à savoir la distance de Manathan, la distance Euclidienne, la distance Khi2 et la divergence de KullBack Leibler (Kullback et Leibler, 1951). Dans ce qui suit, nous présentons la description de notre algorithme dans la Fig.1 et les notations utilisées par notre algorithme d'apprentissage de concepts ontologiques dans la Table 2.

Notre algorithme de clustering est incrémental. Il calcule les occurrences de chaque mot et sélectionne leurs cooccurrents sémantiquement proches en respectant la

définition contextuelle. Ensuite, il divise les classes obtenues à chaque étape afin de raffiner le contexte de chaque classe de mots. Ainsi, l'algorithme ECO raffine en même temps le contexte de chaque terme et le contexte de chaque classe de mots. Notre approche contextuelle se ramène à un problème de sélection de variables connu en statistique et un problème de représentation transparente des mots qui reflète efficacement leur sémantique. Notre algorithme ECO offre, également, la possibilité à l'utilisateur de choisir entre une exécution complètement automatique ou interactive. Si l'utilisateur décide le premier mode d'exécution, il devra définir certains paramètres ou choisir ceux définis par défaut et résultant de nos expérimentations empiriques. Ces paramètres sont le nombre de classes K, le plus grand nombre de mots par classe exprimant le plafond P, la marge M acceptée par l'utilisateur et exprimant le nombre de mots supplémentaires qu'il tolère et la mesure de similarité S. Si l'utilisateur préfère évaluer les classes intermédiaires, il devra choisir le mode d'exécution interactif. Dans ce dernier cas, l'algorithme lui permet d'analyser les classes de mots à chaque phase de clustering intermédiaire afin qu'il puisse définir la valeur k' et décide si il veut poursuivre le mode interactif avec un contrôle permanent ou lancer une exécution automatique pour le reste de la procédure. En adoptant un mode interactif, le processus prend plus de temps que celui automatique mais offre l'opportunité à l'utilisateur d'intervenir afin d'avoir de meilleures classes hiérarchiques de mots.

Table 2. Notations utilisées par notre algorithme d'Extraction de Concepts Ontologiques

F	Le fichier d'entrée pour notre algorithme
WC	Ensemble de classes de mots WC ={C1, C2, C3, …, CT} avec T: Nombre total de classes
K	Nombre de classes
P	Le plus grand nombre de mots par classes qui soit accepté par l'utilisateur
M	Le nombre possible de mots pouvant exister dans une classe comme un supplément d'information qui est défini et toléré par un utilisateur
S	La mesure de similarité
Di	La distribution des mots dans les différentes classes Ci avec Di ={C1, C2, C3, …, Ci}
Ci	C'est la classe de mots appartenant à Di avec Ci Є Di
Nombre-Mots(Ci)	C'est le nombre de mots dans la classe Ci
W i	C'est chaque mot appartenant à Ci avec Ci = {W 1, W 2 , …, W i}
Cc-c	C'est la classe de mots ayant le plus proche centre de l'individu désigné W i

Lors de la phase d'initialisation de notre algorithme, l'utilisateur choisit le fichier de données qui va être traité. Puis, il définit le nombre de classes K et le mode d'exécution. Dans ce qui suit, nous expliquons le déroulement de l'algorithme ECO dans un cas d'exécution automatique :

Etape 0. Cette étape permet de calculer le contexte structurel et de donner un fichier contenant les mots, leurs cooccurrents ainsi que les pondérations associées.

Etape 1. Le but de cette étape est l'exécution de l'algorithme Kmeans afin d'obtenir une première distribution des mots Di. Après ces itérations, nous obtenons K classes de mots.

Etape 2. Cette étape permet d'obtenir des classes de mots respectant le critère de plafond P défini par l'utilisateur. Pour chaque exécution intermédiaire de Kmeans, il faut définir le nombre de classes K' à obtenir en éclatant une classe en sous classes. Cette valeur intermédiaire K' dépend de chaque classe obtenue et ne peut pas être définie par l'utilisateur lors de l'initialisation. Afin de résoudre ce problème, nous définissons une fonction de proportionnalité qui permet de définir automatiquement la valeur K' et qui est basée sur des expérimentations empiriques et les connaissances de l'expert du domaine. En utilisant ces informations (définissant les valeurs *a* et *b*) et en résolvant l'équation suivante, nous pouvons calculer la valeur K'. L'équation définie est : K' = a *ln (Nombre-Mots(Ci) * b).

Dans les cas ou le nombre de mots par classe (Nombre-Mots(Ci)) est inférieur à (1/b), la valeur de K' n'est plus calculée mais elle est prise par défaut comme étant 2. Quand le nombre de mots par classe Ci est inférieur ou égal à P, la classe Ci est incluse dans l'ensemble WC.

Etape 3. Le problème des classes à un seul mot apparaît lorsque nous éclatons une classe en plusieurs sous classes en suivant un processus répétitif. Notre idée est d'associer automatiquement chaque mot seul à une des classes déjà obtenues dans l'étape 2. Un autre problème se présente quand l'algorithme affecte un nombre de termes relativement grand à une même classe. Dans ce cas, nous trouvons des classes avec un nombre trop grand de mots. Afin d'éviter ce problème, l'utilisateur définit la valeur de la marge M qui exprime son degré de tolérance du dépassement de la valeur P c-à-d l'utilisateur n'accepte que les classes dont la taille est inférieure ou égale à P+M. Ainsi, l'algorithme ECO choisit la classe dont le centroide est le plus proche du mot à affecter tout en vérifiant la valeur P+M. Si l'algorithme voulant affecter un mot à une classe dont la taille dépasse cette valeur (P+M), alors il sera amené à choisir une seconde classe dont le centre est le second plus proche. Dans le cas ou toutes les classes sont saturées, l'algorithme tolère l'existence d'une classe avec un seul mot.

```
0: Algorithme Extraction Concepts Ontologiques (In: F, K, P, M, S, Out: WC)
1 : Appliquer notre définition de contexte et calculer les occurrences des termes {/*Etape0*/}
2: Di ← Φ {/* Etape 1*/}
3: Choisir aléatoirement les K premiers centres
4: Affecter chaque mot à la classe ayant le plus proche centre
5: Recalculer les positions des centres
6: Si (les positions des centres ne changent plus) Alors
7:    aller à ligne 10
8: Sinon
9:    aller à ligne 4
10: Di ← Di U {C1, C2, C3, …, Ck}
11: Pour chaque Ci Є Di do {/* Etape 2*/}
12:    Si (Nombre-Mots (Ci) ≤ P) Alors
13:    WC ← WC U {Ci}
14:    Di ← Di \ {Ci}
```

```
15:  Sinon
16:    Di ← Di \ {Ci}
17:    Déverrouiller les mots Wi appartenant à la classe Ci
18:    Calculer la valeur de K
19:    Aller aux lignes 3, 4, 5 et 6
20: Pour chaque Ci Є WC Faire {/* Etape 3*/}
21:  Si (Nombre-Mots (Ci) = 1 et Wi Є Ci) Alors
22:    Calculer la position de Wi aux centres des classes existantes Ci Є WC
23:    Si (Nombre-Mots (Cc-c) > P+M) Alors
24:      Choisir une autre classe Cc-c ayant le second plus proche centre de l'objet désigné
25:      Aller à ligne 20
26:    Sinon
27:       WC ← WC \ {Cc-c, Ci}
28:       Affecter Wi à la classe Cc-c
29:       WC ← WC U {Cc-c}
30: Retourner (WC)
31: End
```

Fig. 1 – La description de l'algorithme d'Extraction de Concepts Ontologiques (ECO)

Si l'utilisateur choisit le mode d'exécution interactif, il sera amené à intervenir après l'étape 1 et durant l'étape 2 et 3. Dans l'étape 2, quand une classe est divisée, l'algorithme ECO raffine le contexte de chaque classe en prenant en compte uniquement les attributs associés aux mots appartenant à la classe en question. En appliquant cette méthode, la similarité calculée sera plus représentative du degré d'association entre les mots candidats. En appliquant l'étape 3, ECO évite les cas de classes avec un seul mot ainsi que celles contenant uniquement des mots seuls.

4 Expérimentations de l'algorithme d'extraction des concepts ontologiques

Dans cette section, nous évaluons les résultats de l'algorithme ECO en comparaison avec ceux obtenus avec l'algorithme Kmeans. Afin de comparer les résultats des deux algorithmes, nous les avons exécutés avec le même fichier de données calculé en utilisant notre définition de contexte. Nous avons choisi la distance euclidienne en tant que mesure de similarité et nous avons limité nos expérimentations aux deux premiers niveaux de la hiérarchie contextuelle à savoir les balises clefs + les titres + les sous titres (Karoui et al, 2006). Notre fichier de données est composé de 872 mots. ECO a été expérimenté avec plusieurs valeurs pour ces paramètres. Dans ce papier, nous présentons les résultats obtenus (les plus significatifs) avec respectivement pour K, P et M les valeurs 20, 10 et 22. L'algorithme ECO et Kmeans donnent respectivement 162 classes et 156 classes.

Dans un processus de clustering, la qualité d'une classe est généralement basée sur l'homogénéité ou la compacité. Dans (Vazirgiannis et al, 2003), des critères d'évaluation statistique de l'apprentissage non supervisé sont définis. Cependant, les applications liées à l'extraction de connaissance et à la construction d'ontologie ne peuvent pas appliquer ces standards définis pour d'autres applications. En effet,

l'homogénéité de la classe n'implique pas que les mots lui appartenant sont sémantiquement proches ou que le label associé satisfait l'expert du domaine. Concernant la découverte de connaissances, l'évaluation reste un challenge. Dans (Holsapple et Joshi, 2005), les auteurs ont proposé une méthode d'évaluation basée sur une ontologie construite manuellement. Dans (Navigli et al, 2004), les auteurs proposent une évaluation qualitative par les experts de domaine qui répondent à un questionnaire dans lequel ils évaluent la qualité des concepts découverts. Dans d'autres travaux, l'évaluation et le processus de labellisation sont basés sur un thesaurus. Mais le thesaurus ne couvre pas forcément tous les aspects spécifiques d'un domaine. Dans notre cas, certains termes de nos classes n'apparaissent pas dans le Thesaurus de OMT (Organisation Mondiale du Tourisme). Ainsi, nous avons proposé une évaluation manuelle par deux experts du domaine. Nous présentons les résultats à ces deux experts. D'abord et individuellement, chacun d'entre eux évalue et labellise manuellement les classes de mots ce qui revient à lui associer un concept appartenant au domaine et relatif à son contenu. Ensuite, ils travaillent ensemble pour discuter des résultats de leurs propositions de labels et nous fournissent une évaluation unique sur laquelle ils se sont mis d'accord. Pour évaluer et présenter les résultats de l'expertise, nous avons défini six critères : la distribution des termes, la pondération de paires de termes, la similarité de paires de termes, les concepts extraits, l'interprétation sémantique et le degré de généralité des concepts extraits. Concernant la pondération des termes et la similarité entre eux, nous avons établi des expérimentations pour évaluer notre contexte structurel par rapport à un contexte statique (fenêtre de taille 10) en utilisant l'algorithme Kmeans (Karoui et al, 2006).

Dans cet article, nous présentons les résultats obtenus suite à la comparaison des deux algorithmes ECO et Kmeans :

Distribution des termes. Avec l'algorithme Kmeans, nous obtenons 13% des données dans une même classe. Alors qu'avec l'algorithme ECO, nous obtenons uniquement 3.66% des mots dans la même classe.

Interprétation sémantique. Lors de l'évaluation, l'expert de domaine constate qu'il existe trois types de classes résultantes : les classes acceptables, incorrectes et inconnues. Une classe acceptable est une classe que l'expert est capable de labelliser et dans laquelle les termes appartenant au même groupe sont proches sémantiquement. Une classe incorrecte est une classe qui présente l'un des deux cas suivants : soit elle contient certains termes qui n'ont pas de relations avec le concept extrait de cette classe, soit elle contient plusieurs concepts clairement identifiés par l'expert. Une classe inconnue est une classe dont les termes n'ont aucune relation sémantique ; l'expert ne peut pas en donner une interprétation sémantique. Grâce aux paramètres P et M, pour chaque classe de mots, le pourcentage de mots considérés comme du bruit a énormément diminué. Enfin, nous avons obtenu 68.52% de classes acceptables avec l'algorithme ECO par rapport à uniquement 53.2% pour l'algorithme Kmeans (voir Table 3).

Table 3. Le détail des concepts extraits

	Classes Acceptables	Classes Incorrectes	Classes inconnues
Algorithme Kmeans	53. 2 %	26. 28 %	20. 51 %
Algorithme ECO	68.52%	16.66%	14.81%

Concepts Extraits. Nous prenons en compte uniquement les classes acceptables dans les deux cas et nous calculons la précision. Dans notre étude, « la précision est le ratio des termes pertinents ayant entre eux une importante similarité sémantique par rapport à l'ensemble des termes d'une classe donnée». Comme résultats, nous obtenons respectivement 86.18% et 86,61% avec les algorithmes Kmeans et ECO.

Degré de généralité des concepts extraits. Un autre élément qui détermine la qualité d'un concept extrait est son degré de généralité. Dans une ressource ontologique, il est important d'avoir un ensemble important de concepts généraux qui résument le domaine et forment les concepts les plus génériques de l'ontologie. Ainsi, nous focalisons notre intérêt sur les concepts extraits des classes acceptables. Nous établissons une évaluation manuelle en nous basant sur le thésaurus de l'Organisation Mondiale du Tourisme (OMT). Ce thésaurus contient des termes génériques représentant les concepts clés du domaine. Comme résultats, nous obtenons respectivement avec l'algorithme Kmeans et ECO 78.31 % et 85.58% de concepts généraux. Nous avons, également, remarqué que les classes obtenues avec notre algorithme sont plus riches sémantiquement. Par exemple, avec l'algorithme Kmeans nous avons la classe C1 :{événement, festival, musique} alors qu'avec l'algorithme ECO nous avons C2 : {événement, festival, musique, fête}.

Impact du paramètre P et de l'étape 2 de l'algorithme ECO. Une première exécution de Kmeans avec 20 classes (étape 1), nous donne une grande classe avec 68.69% des données initiales. En définissant le paramètre de plafond P, nous décidons de diviser les classes de mots et d'établir un clustering intermédiaire qui assurera une sélection des variables communes au P mots. Donc, l'algorithme ECO raffine le contexte de chaque classe afin d'obtenir de meilleures similarités entre les mots. Par exemple, avec l'algorithme Kmeans, nous avons trouvé le terme « civilisation » dans une énorme classe dont les mots n'ont aucune relation. Alors qu'avec notre algorithme ECO, nous retrouvons ce même terme avec des mots proches sémantiquement tels que « archéologie », « ethnologie », « population », etc.

Impact de l'étape 3 de l'algorithme ECO. Notre algorithme permet d'affecter ces mots seuls à des classes respectant déjà le paramètre de plafond P. Grâce à cette troisième étape, nous avons remarqué que certains mots seuls ont été bien affectés à leurs classes appropriées (dont les mots sont proches sémantiquement). Par exemple, nous obtenons respectivement avant et après cette étape 3 les classes {académie, club, golfeur} et {académie, club, golfeur, golf}.

Discussion. Dans cette section, nous avons montré que notre algorithme d'Extraction de Concepts Ontologiques (ECO) procure de meilleurs résultats par rapport à l'algorithme Kmeans. Notre contexte générique, riche et ne se limitant pas à une phrase (comme pour les approches linguistiques), est basé sur une hiérarchie contextuelle constituée par des liens entre les balises HTML. Il ne permet pas de grouper uniquement des termes avec leurs variations terminologiques ou leurs synonymes mais aussi des termes sémantiquement proches pouvant contribuer à la

formation d'un concept de domaine. Afin de maintenir notre méthode fonctionnelle, nous devons avoir un minimum de structure dans les documents. Par contre, l'analyse de structure (Karoui et al, 2006) adapte la définition du contexte en prenant en compte la majorité des balises HTML utilisées. L'absence de certaines balises n'affecte pas le fonctionnement de notre méthode. Par exemple, si nous ne disposons pas de balise sous titre <h1>, la méthode reste opérationnelle puisqu'elle va chercher les cooccurrents du mot dans les niveaux inférieurs de la hiérarchie contextuelle (dans les balises <p>, <td>, etc.). Lorsque, la balise <title> (titre), qui permet d'initier le processus de clustering n'est pas présente, nous avons d'autres balises clefs qui permettent d'apporter de l'information. Dans ce travail, nous ne pouvons pas utiliser la précision et le rappel puisque nous utilisons une collection de documents qui n'a pas des pré-clusters. Mais, notre architecture (Karoui et al, 2006) fournit un support de prétraitement des documents HTML qui nous permet d'appliquer l'algorithme ECO que ce soit sur un benchmark relié à un domaine bien spécifique (dans ce cas, nous pourrons utiliser les critères d'évaluations standards comme la précision) ou sur une collection de domaine qu'elle soit collectée par le chercheur ou donnée par une tierce partie (industrie, institution, etc.).

5 Conclusion

L'acquisition des connaissances est une tâche difficile et lourde au regard de la diversité langagière du Web et de ses connaissances. Dans ce papier, nous nous sommes focalisés sur le processus d'extraction de concepts ontologiques en exploitant la structure HTML, les relations entre les balises HTML et la position du mot. Le contexte structurel améliore la pondération des termes, la sélection des cooccurrents sémantiquement proches et le résultat du clustering puisqu'il raffine le contexte de chaque terme. Nous avons défini, ensuite, un algorithme de clustering hiérarchique intitulé « Extraction de Concepts Ontologiques » (ECO) basé sur une utilisation incrémentale de l'algorithme de partitionnement Kmeans et guidé par notre définition contextuelle structurelle. L'algorithme ECO procède d'une manière incrémentale pour raffiner le contexte de chaque cluster de mots et par conséquent améliorer la qualité conceptuelle des connaissances extraites. ECO offre le choix entre une exécution automatique ou interactive. Les résultats ont montré que l'algorithme ECO fournit de meilleurs résultats par rapport à une définition contextuelle statique et un algorithme de clustering existant. Dans ce travail, nous avons évalué la qualité conceptuelle des classes de mots. En perspectives, nous allons évaluer la qualité sémantique des liens entre les classes de mots dans la hiérarchie. Egalement, nous allons poursuivre nos expérimentations concernant les niveaux inférieurs de la hiérarchie contextuelle et nous allons définir un contexte linguistique et un contexte documentaire puis les combiner avec le contexte structurel afin de tenir compte des cas où les documents HTML sont pauvres en structure, d'améliorer la finesse de décomposition des clusters et de construire une hiérarchie de clusters. Egalement, nous allons tenir compte d'autres types de documents dans le web (XML, XHTML, Word, etc.).

Références

Buyukkokten, O., Garcia-Molina, H. and Paepcke, A. (2001). Accordion summarization for end-game browsing on PDAs and Cellular Phones. In Proc. Of Conference on Human Factors in Computing Systems.

Cai, D., Yu, S., Wen, J. and Ma, W. (2004). Block-based web search. Proceedings of the 27 th annual international ACM SIGIR conference on research and development in information retrieval, pages 456-463.

Davulcu, H., Vadrevu, S. and Nagarajan, S. (1998). OntoMiner: Boostrapping ontologies from overlapping domain specific web sites. In AAAI'98/IAAI'98 Proceedings of the 15th National Conference on Artificial Intelligence.

Faure, D., Nedellec, C. and Rouveirol, C. (1998). Acquisition of semantic knowledge uing machine learning methods: the system ASIUM. Technical report number ICS-TR-88-16, inference and learning group, University of Paris-sud.

Han, H. and Elmasri, R. 2000. Architecture of WebOntEx: A system for automatic extraction of ontologies from the Web". WCM.

Holsapple, C. and Joshi, K.D. (2005). A collaborative approach to ontology design. Communications of ACM, 45(2): 42-47.

Karoui, L., Aufaure, M-A. and Bennacer, N. (2006). A New Extraction Concepts based on Contextual Clustering. The IEEE International Conference on Computational Intelligence for Modelling, Control and Automation.

Kiyota, Y. and Kurohashi, S. (2001). Automatic summarization of Japanese sentences and its application to a WWW KWIC index. Proceedings of the 2001 Symposium on applications and the internet, page 120.

Kullback, S and R. A. Leibler. (1951). On information and sufficiency. Annals of Mathematical Statistics 22(1):79–86.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of 5th Berkeley Symposium on Mathematics, Statistics and Probability, 1:281- 298.

Meadche, A. and Staab S. (2001). Ontology learning for the semantic Web. IEEE journal on Intelligent Systems, Vol. 16, No. 2, 72-79.

Michelet, B. 1988. L'analyse des associations. Thèse de doctorat, Université de Paris VII, UFR de Chimie.

Nakache, J.P. and J. Confais. (2005). Approche pragmatique de la classification : arbres hiérarchiques, partitionnements. Editions Technip.

Navigli, R., Velardi, P., Cucchiarelli, A. and Neri, F. (2004). Quantitative and qualitative evaluation of the ontolearn ontology learning system. In Proc. Of ECAI-2004 Workshop on ontol ogy learning and population.

Sugiura, N, N., Izumi and T. Yamaguchi (2004). A supprt environment for domain ontology development with general ontologies and text. IEEE Computational Intelligence Bulletin, February 2004, Vol.3 No.1.

Vazirgiannis, M., Halkidi, M. and Gunopoulos, D. (2003): uncertaintly handling and quality assessmen in data mining. Springer.