

Annotation sémantique de pages web

Sylvain Tenier*,** Amedeo Napoli** Xavier Polanco* Yannick Toussaint**

*Institut National de l'Information Scientifique et Technique
54514 Vandoeuvre-lès-Nancy, France
{polanco,tenier}@inist.fr
<http://www.inist.fr/uri/accueil.htm>

**Laboratoire Lorrain de Recherche en Informatique et ses Applications
BP 239, 54506 Vandoeuvre lès Nancy Cedex, France
{napoli,toussaint,tenier}@loria.fr
<http://www.loria.fr/equipes/orpailleur>

Résumé. Cet article présente un système automatique d'annotation sémantique de pages web. Les systèmes d'annotation automatique existants sont essentiellement syntaxiques, même lorsque les travaux visent à produire une annotation sémantique. La prise en compte d'informations sémantiques sur le domaine pour l'annotation d'un élément dans une page web à partir d'une ontologie suppose d'aborder conjointement deux problèmes : (1) l'identification de la structure syntaxique caractérisant cet élément dans la page web et (2) l'identification du concept le plus spécifique (en termes de subsumption) dans l'ontologie dont l'instance sera utilisée pour annoter cet élément. Notre démarche repose sur la mise en oeuvre d'une technique d'apprentissage issue initialement des wrappers que nous avons articulée avec des raisonnements exploitant la structure formelle de l'ontologie.

Le système que nous présentons permet d'automatiser l'annotation sémantique de pages web. Notre objectif est de classer des pages concernant des équipes de recherche, afin de pouvoir déterminer par exemple qui travaille où, sur quoi et avec qui. La classification s'appuie sur des mécanismes de raisonnement qui nécessitent une représentation formelle du contenu des pages ; nous exploitons ainsi une ontologie qui représente les concepts du domaine et les relations entre les concepts dans un langage de représentation des connaissances.

Notre système génère des *annotations sémantiques* qui sont des métadonnées sur les éléments d'un document liées à une ontologie. Pour cela nous devons résoudre deux grandes questions. La première est d'identifier automatiquement, dans une page web, les éléments qui sont pertinents. La seconde est de déterminer quels sont les concepts de l'ontologie les plus spécifiques possible, pour annoter chacun de ces éléments.

L'automatisation repose sur un apprentissage à partir d'un corpus constitué d'éléments marqués par un expert. Le marquage associe à chaque concept de l'ontologie des éléments de la page en rapport avec ce concept. L'apprentissage génère un wrapper capable d'annoter des éléments du document sous la forme d'instances de concepts et de rôles de l'ontologie fournie. Des mécanismes de raisonnement exploitant l'ontologie sont utilisés pour déterminer le concept le plus spécifique avec lequel un élément doit être annoté. L'annotation est donc totalement dépendante de l'ontologie fournie.

Annotation sémantique de pages web

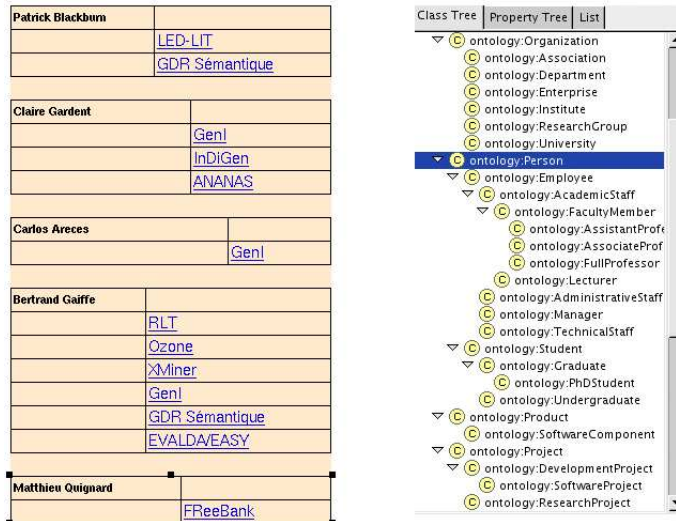


FIG. 1 – page web et ontologie présentées à l'expert

Dans une première section, nous présentons le processus de marquage de la page par un expert. La seconde section présente l'algorithme d'apprentissage exploitant la structure arborescente d'une page web. La section 3 présente l'annotation de documents dont la structure est similaire. Enfin, nous évaluons notre méthode par rapport aux systèmes d'annotation sémantiques existants.

1 Génération d'annotations primaires par marquage

La première étape du processus est un marquage permettant de former un corpus d'apprentissage ; il s'agit de fournir au système quelques exemples d'éléments pertinents à partir desquels le système apprend à reconnaître l'ensemble des éléments à annoter. Pour cela, un expert marque des éléments pertinents de la page web, c'est-à-dire correspondant à des concepts de l'ontologie. Il dispose à cet effet d'un outil de visualisation, à la manière d'un navigateur web, qui lui permet de sélectionner un élément dans la page et de choisir dans l'ontologie le concept qui lui correspond. Pour chaque concept, un nombre suffisant d'éléments pouvant y être associés doivent être marqués ; ce nombre dépend de la régularité de la page d'apprentissage et des pages à annoter ; pour des pages très régulières, 2 ou 3 exemples suffisent pour chaque concept.

De manière interne, la page est représentée par son arbre DOM (W3C) dans lequel les noeuds contiennent les éléments de structure HTML et les feuilles les éléments de texte. Un chemin unique est ainsi défini depuis la racine jusqu'à chaque feuille. Lorsque l'expert marque un élément, 3 propriétés sont enregistrées au format XML : la chaîne de caractères sélectionnée est enregistrée dans une balise `<text>`, le chemin de la feuille contenant la chaîne est enregistré dans la balise `<nodeloc>` et le concept de l'ontologie associé est enregistré dans la balise `<instof>`.



FIG. 2 – Arbre DOM et annotations primaires issues du marquage

La figure 2 présente un exemple de marquage à partir de la page présentée en figure 1 et de l'ontologie *SWRC*¹, qui modélise notamment les personnes, organismes et projets d'une équipe de recherche. L'ensemble de ces éléments marqués sont des *annotations primaires* qui jouent ainsi le rôle de corpus pour l'algorithme d'apprentissage.

2 Apprentissage exploitant une structure arborescente

Définition d'un chemin dans l'arbre DOM L'algorithme d'apprentissage est dérivé des travaux de Kushmerick et al. (1997) sur l'induction de wrappers. Un wrapper est une procédure utilisant les régularités syntaxiques d'un document pour identifier des éléments. Là où les travaux initiaux s'appuyaient sur des structures à plat, en considérant le document comme une suite de chaînes de caractères, notre système exploite la structure arborescente fournie par la représentation DOM de la page web.

Le DOM permet de définir le chemin de chaque élément (noeud ou feuille) de l'arbre. Pour chaque élément, nous définissons ce chemin comme un ensemble d'étapes depuis la racine. Chaque étape est un couple (balise : position) défini à partir de l'étape précédente (on considère l'étape 0 comme étant la racine du document). La position est le numéro du fils du noeud défini à l'étape précédente tandis que la balise est la balise HTML que le noeud représente. Par exemple, une page web contient un élément racine `<html>` qui a deux fils, `<head>` et `<body>`. Le chemin de l'élément `<body>` est donc `body : 1`. Cette définition de chemin est celle employée pour les annotations primaires présentées fig. 2.

¹<http://ontoware.org/projects/swrc/>

A partir de cette définition du chemin d'un élément de l'arbre, on définit la notion de *chemin similarisé*. Un chemin similarisé est la factorisation des chemins de plusieurs éléments. Le chemin ainsi généré est ainsi un chemin de plusieurs éléments. Pour cela, les étapes sont comparées 2 à 2 et les différences marquées par une astérisque. Prenons l'exemple des deux premières annotations primaires présentées figure 2. Le chemin du premier élément est $body : 2, table : 0, tbody : 0, tr : 1, td : 1, b : 0$ tandis que le deuxième élément a pour chemin $body : 2, table : 0, tbody : 0, tr : 0, td : 0, b : 0$. Le chemin similarisé de ces deux éléments est $body : 2, table : 0, tbody : 0, tr : *, td : *, b : 0$. La génération d'un chemin similarisé à partir de plus de deux éléments se fait de manière incrémentale.

Relations entre la représentation arborescente de la page web et l'ontologie Cette définition de chemin est fondamentale pour l'apprentissage. Notre stratégie d'annotation est fondée sur l'hypothèse d'une corrélation entre la représentation arborescente d'un document et les concepts et rôles définis par l'ontologie. Ces hypothèses sont les suivantes :

- chaque instance de concept est exactement une feuille de l'arbre,
- les instances de rôles sont contenues dans des sous-arbres

De ces hypothèses, on déduit qu'identifier une instance de l'ontologie revient à déterminer le chemin depuis la racine vers une feuille de l'arbre pour une instance de concept et vers un noeud, racine du sous-arbre, pour une instance de rôle. L'apprentissage consiste donc à déterminer un chemin similarisé pour chaque concept et chaque rôle de l'ontologie.

Apprentissage de chemins similarisés Pour chaque concept dont des exemples ont été marqués par l'expert, le chemin similarisé du concept est généré à partir de l'ensemble des chemins enregistrés dans les annotations primaires pour ce concept. Dans l'exemple figure 1, 5 annotations primaires sont définies pour le concept *Project*. En factorisant les chemins deux à deux, le chemin similarisé obtenu est $body : 2, table : *, tbody : 0, tr : *, td : 1, a : 0, font : 0$. Il ressort ainsi que les éléments correspondant au concept *Project* sont situés dans la deuxième colonne des tableaux du document.

Pour les instances de rôles, une première étape consiste à déterminer les racines des sous-arbres de chaque rôle tel que :

- il existe un rôle R_{AB} dans l'ontologie reliant des concepts A et B ,
- au moins une instance de A et une instance de B ont été marquées.

Alors pour chaque instance marquée de A :

- le plus petit parent commun (pppc) dans l'arbre de cette instance avec chaque instance de B est déterminé,
- le noeud le plus profond dans l'arbre parmi ces pppc est alors un noeud racine pour le rôle R_{AB} .

Le chemin similarisé des noeuds racines générés est alors inféré. La sortie de l'apprentissage est donc un chemin similarisé de chaque concept et de chaque rôle de l'ontologie présents dans le document.

3 Annotation par génération d'instances de l'ontologie

Annotation par application des chemins similarisés Les chemins similarisés sont appliqués sur une page dont la structure DOM est similaire à la page d'apprentissage. Les noeuds

reconnus par le chemin similarisé appris pour chaque rôle $R_{A,B}$ sont les racines des sous-arbres en dessous desquels chaque instance de a est liée à une instance de b par une instance de $R_{A,B}$. Les feuilles reconnues par le chemin similarisé d'un concept sont des candidates pour être instanciées par ce concept. Deux cas sont possibles : si une feuille n'est reconnue que par un seul chemin similarisé, cette feuille est instanciée par le concept correspondant à ce chemin. Pour toutes les feuilles situées dans un sous-arbre, une relation est générée entre les instances de concepts définies par le rôle. Si plusieurs chemins similarisés conduisent à la même feuille, un mécanisme de raisonnement doit être appliqué pour déterminer à quel concept cette feuille appartient. On atteint les limites d'une méthode purement syntaxique.

Dans notre exemple, le chemin similarisé du concept *Project* décrit ainsi le fait qu'il est associé aux éléments contenus dans la colonne de droite des tableaux tandis que les concepts *Lecturer* et *FacultyMember* sont associés au contenu de la colonne de gauche.

Annotation par un concept plus général dans l'ontologie Lorsqu'un même élément peut être annoté par deux concepts différents, un raisonnement est effectué au niveau de l'ontologie pour déterminer le concept subsumant les deux concepts candidats. Dans notre exemple, le raisonneur Pellet (Sirin et Parsia (2004)) est utilisé pour classifier les concepts de l'ontologie et déterminer le concept subsumant *Lecturer* et *FacultyMember* dans *SWRC*. Il s'agit de *AcademicStaff*. Une instance de ce concept sera donc générée pour les éléments reconnus par les chemins similarisés appris à partir des concepts *Lecturer* ou *FacultyMember*.

4 Evaluation du système

L'originalité de notre approche est qu'elle permet de générer des instances de rôles en plus des instances de concepts de l'ontologie. L'objectif initial des travaux sur l'annotation de documents était de permettre le travail collaboratif entre les personnes. Dans le cadre de l'annotation de pages web, Annotea (Kahan et Koivunen (2001)) est un système développé par le W3C générant des annotations en RDF, un langage permettant de décrire des ressources et les relations entre ces ressources. Des travaux inspirés d'Annotea ont conduit à des systèmes d'annotation par rapport à une ontologie en DAML+OIL (S-CREAM, Handschuh et al. (2002)) puis en OWL (SMORE, Kalyanpur et al. (2003)) L'utilisation de ces langages d'ontologie implémentant les logiques de description permettent ainsi aux machines de raisonner sur les annotations produites. Cependant, l'annotation manuelle par un expert étant une source d'erreurs, des systèmes supervisés à partir d'apprentissage ont été proposés, notamment S-CREAM et MnM (Vargas-Vera et al. (2002)); ceux-ci reposent sur le système d'extraction d'information Amilcare (Ciravegna et al. (2002)) qui génère des règles d'extraction à partir d'un corpus de documents fourni en entrée. Cette technique permet de générer des instances de concepts mais n'exploite pas la structure globale du document; les relations entre les instances ne sont donc pas extraites. Enfin, des systèmes non-supervisés comme Amardillo (Ciravegna et al. (2004)) ou C-PANKOW (Cimiano et al. (2005)) visent à sortir totalement l'humain de la boucle d'annotation en exploitant la redondance de l'information sur le web. Ces systèmes sont toutefois inadaptés aux pages contenant plusieurs instances d'un même concept reliées à des instances d'un autre concept. Dans notre approche, l'exploitation de la structure arborescente de la page présente toutefois certaines limites en fonction de la régularité de la page. Elle s'applique à

des documents de type tabulaire contenant de multiples instances des concepts de l'ontologie. Les pages des équipes de recherche exploitées dans notre cadre d'application permettent généralement cette exploitation.

Références

- Cimiano, P., G. Ladwig, et S. Staab (2005). Gimme' the context : context-driven automatic semantic annotation with c-pankow. In *WWW '05 : Proceedings of the 14th international conference on World Wide Web*, New York, NY, USA, pp. 332–341. ACM Press.
- Ciravegna, F., S. Chapman, A. Dingli, et Y. Wilks (2004). Learning to harvest information for the semantic web. In *ESWS*, pp. 312–326.
- Ciravegna, F., A. Dingli, Y. Wilks, et D. Petrelli (2002). Adaptive information extraction for document annotation in amilcare. In *SIGIR '02 : Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pp. 451–451. ACM Press.
- Hands Schuh, S., S. Staab, et F. Ciravegna (2002). S-cream-semi-automatic creation of metadata. *Proc. of the European Conference on Knowledge Acquisition and Management*. Springer Verlag (submitted version).
- Kahan, J. et M.-R. Koivunen (2001). Annotea : an open rdf infrastructure for shared web annotations. In *WWW '01 : Proceedings of the 10th international conference on World Wide Web*, New York, NY, USA, pp. 623–632. ACM Press.
- Kalyanpur, A., B. Parsia, J. Hendler, et J. Golbeck (2003). SMORE - semantic markup, ontology, and RDF editor.
- Kushmerick, N., D. S. Weld, et R. B. Doorenbos (1997). Wrapper induction for information extraction. In *IJCAI (1)*, pp. 729–737.
- Sirin, E. et B. Parsia (2004). Pellet : An owl dl reasoner. In *Description Logics*.
- Vargas-Vera, M., E. Motta, J. Domingue, M. Lanzoni, A. Stutt, et F. Ciravegna (2002). Mnm : Ontology driven semi-automatic and automatic support for semantic markup. In *EKAW*, pp. 379–391.
- W3C. Le Document Object Model (DOM).

Summary

This article presents an automatic webpages semantic annotation system. Legacy systems are mainly syntactic, even when they aim at producing semantic annotations. Taking into account semantic information from the domain to annotate an element in a webpage implies solving two problems : (1) identifying the syntactic structure behind this element in the webpage and (2) identifying the most specific concept of the ontology which is used to annotate this element. Our approach relies on a wrapper-based machine learning algorithm combined with reasoning making use of the formal structure of the ontology