Construction d'ontologies à partir d'une collection de pages web structurées

Nathalie Aussenac-Gilles, Mouna Kamel, Davide Buscaldi, Catherine Comparot

Construction d'ontologie à partir d'une collection de pages web structurées

Nathalie Aussenac-Gilles[1], Mouna Kamel[1], Davide Buscaldi[2], and Catherine Comparot[1]

[1] IRIT - CNRS, Université de Toulouse - 118 route de Narbonne, 31062 Toulouse, France
prenom.nom@irit.fr
[2] LIPN - Université Paris Nord - av.Jean-Baptiste Clément, 93430 Villetaneuse, France
davide.buscaldi@lipn.univ-paris13.fr

Résumé : De nombreuses collections de documents disponibles sur le web décrivent les caractéristiques d'entités d'un même type (e.g. des produits, des plantes), chaque page présentant une de ces entités. Ces documents sont des sources de connaissances particulièrement adaptées pour la construction d'ontologies. Alors qu'ils partagent une même mise en forme régulière, ils contiennent moins de texte rédigé que des fichiers textes mais leur architecture est riche de sens. De ce fait, les méthodes linguistiques classiques pour identifier des concepts et des relations sont moins adaptées pour les analyser. Nous proposons une approche exploitant les diverses propriétés de ces documents, combinant analyse de la structure et de la mise en forme avec une analyse linguistique, et exploitant leur annotation sémantique.

Mots-clés : Construction d'ontologies, Enrichissement d'ontologies, Structure documentaire, Mise en forme de documents, Annotation sémantique

1 Introduction

Parmi les documents textuels du web, certains sont bien structurés et organisés dans des collections dans lesquelles le lecteur peut naviguer, comme les catalogues et certaines encyclopédies. Dans les catalogues, les documents portent généralement sur des produits commercialisés, des recettes, des conseils médicaux, etc. Dans les encyclopédies, ils correspondent à des informations encyclopédiques sur des éléments naturels (plantes, animaux, etc.) ou artificiels (films, véhicules, etc.). Ainsi, les sites `http://www.airliners.net/aircraft/` et `http://www.igluski.com/france/` décrivent respectivement des

avions et des stations de ski. Dans les deux cas, la collection détaille les caractéristiques d'entités d'un même type, et chaque page présente une entité. Ces collections forment des réservoirs de connaissances pertinents pour la construction de ressources sémantiques comme les ontologies.

Ces documents présentent des informations de façon synthétique et structurée : les entités sont décrites dans de courtes sections, réduisant ainsi les risques d'ambiguïté lors de la lecture. Pour le lecteur, la mise en forme joue le rôle de formulations linguistiques. De façon plus générale, l'architecture ou la matérialité d'un texte contribue à la construction de son sens : « la surcharge graphique d'un texte est dans de nombreux cas équivalente à la surcharge de la prosodie du discours » (Power *et al.*, 2003). Des parties de texte ont souvent une structure hiérarchique exprimée avec des marqueurs typo-dispositionnels. La structure documentaire a ainsi été étudiée pour différentes raisons, pour améliorer les systèmes de génération de textes (Power *et al.*, 2003) et de segmentation de textes (Hearst, 1997), et plus rarement pour construire des ressources linguistiques.

Une conséquence est que ces fiches contiennent beaucoup de connaissances implicites, non formulées sous forme lexicale ou grammaticale. L'analyse de ces documents selon les principes de l'*ontology-building layer cake*, i.e. en utilisant différents traitements du langage naturel pour extraire des composants d'ontologie (termes, concepts, relations taxonomiques et autres relations sémantiques, etc.), nécessiterait de revoir ces processus. Les techniques actuelles comme les patrons lexico-syntaxiques (Hearts, 1992; Montiel-Ponsoda & Aguado de Cea, 2010; Aussenac-Gilles & Jacques, 2008) ou la classification par apprentissage automatique (Cimiano *et al.*, 2004; Poelmans *et al.*, 2010), ne fonctionnent que si des analyseurs syntaxiques produisent des analyses pertinentes des textes (Navigli & Velardi, 2006; Schutz & Buitelaar, 2005). L'extraction de relations suppose par exemple que les concepts liés soient explicitement mentionnés dans les phrases, excluant ainsi les parties de texte plus complexes avec des références contextualisées, ou des formulations implicites.

Pour s'adapter aux contenus des fiches, ces processus pourraient combiner l'analyse du langage et la sémantique des caractéristiques de mise en forme. En prenant en compte la mise en forme et la structure des documents textuels du web, nous espérons disposer d'indices supplémentaires et ainsi améliorer le processus d'extraction d'information, comme l'ont fait Role & Rousse (2006) et J. O'Connor & Das (2011) pour des documents XML, et Groza *et al.* (2007) pour des fichiers LaTex.

Nous proposons une approche originale pour construire une ontologie

à partir d'une collection de documents web structurés tels que des formulaires, qui combine analyse du langage, analyse de la mise en forme et annotation sémantique. Cette approche est générique car elle fait appel à des caractéristiques de mise en forme conventionnelles des formulaires. Elle étend un travail précédent (Kamel & Aussenac-Gilles, 2009) à toute collection dont les documents vérifient les pré-requis suivants : (i) ils décrivent tous des entités d'un même domaine ; (ii) chaque document décrit une entité qui spécialise un concept plus général ; (iii) tous les documents ont la même mise en forme. L'ontologie est construite en trois étapes principales qui font chacune l'objet d'une partie de l'article : (1) pré-traitement des documents, (2) construction d'une ontologie noyau, et (3) enrichissement du noyau avec des concepts et des restrictions de relations.

2 Principes généraux

L'approche proposée s'appuie sur la sémantique véhiculée par les fiches, leur mise en forme et les expressions linguistiques élicitées via cette mise en forme. Nous commençons par expliciter cette sémantique, puis introduisons la méthode générale de construction d'une ontologie.

2.1 Analyse sémantique d'une fiche

Une fiche décrit généralement une entité au moins en indiquant sa dénomination (partie du document que nous appelons *Entité*) et des informations sur ses propriétés, sous forme d'une liste de *champs*. Chaque champ est composé d'un syntagme ou attribut (nom de la propriété) et d'un contenu qui décrit la propriété. La figure 1 est un exemple d'un tel document où le nom de l'entité est le contenu du champ en italique ; les attributs sont matérialisés par des caractères gras. De façon générale, la mise en forme permet au lecteur d'identifier les différents items de la structure.

L'analyse sémantique des fiches s'appuie à la fois sur le Modèle d'Architecture Textuelle ou MAT [1](Luc *et al.*, 1999), et sur les relations du discours de la RST (Mann & Thompson, 1988) [2]. Le MAT identifie les composants d'une structure textuelle et leur relation de composition, tandis que les relations du discours mettent en évidence des liens sémantiques supplémentaires entre ces composants.

1. La MAT fait référence à la façon dont les auteurs organisent l'information dans les textes, d'un point de vue spatial, visuel, logique et réthorique.

2. La RST vise à expliciter la cohérence du discours d'un point de vue rhétorique.

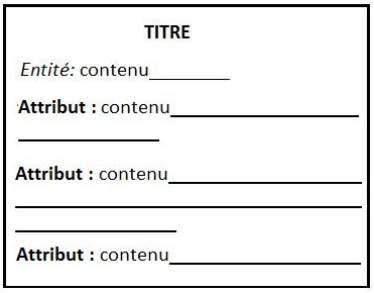

FIGURE 1 – Exemple de mise en forme et architecture visuelle d'une fiche

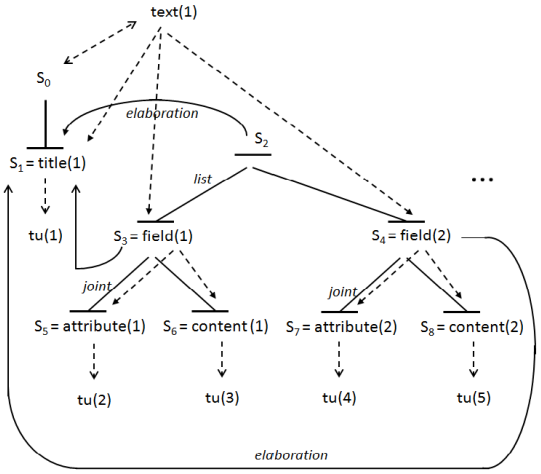

FIGURE 2 – Représentation sémantique de l'architecture d'une fiche (Si = segment ; tu = unité textuelle).

La figure 2 traduit une représentation sémantique des fiches. Les traits pleins et les flèches en pointillés correspondent respectivement aux relations de composition entre segments textuels et aux relations de discours. L'analyse du discours montre que les relations sémantiques entre les segments textuels véhiculant des attributs et leur contenu sont de type *joint* (e.g. entre les segments S5 et S6 de la figure 2), les relations entre segments textuels véhiculant des champs sont de type *list* (e.g. entre les segments S3 et S4 qui représentent deux champs, et S2 qui les aggrège), et que des relations *elaboration* associent les segments textuels véhiculant le titre et les champs (e.g. entre les segments S1 et S2). Nous nous intéressons aux relations d'élaboration entre le champ identifiant l'entité et les différents champs attributs car elles ont pour particularité de lier des segments textuels non contigus. La sémantique de ces relations est précisée par les noms des champs. Nous nous appuyons sur ces relations pour définir les relations sémantiques d'une ontologie. Par ailleurs, le contenu d'un champ peut faire référence à plusieurs concepts ou valeurs de propriété, généralement exprimés comme une liste. Les items de la liste peuvent avoir des relations différentes avec l'entité décrite.

La mise en forme du document de la figure 3 met en évidence des champs par une expression en casse gras (nom du champ) suivie du caractère ":", suivi de paragraphes formant son contenu. Cette structure

FIGURE 3 – Une fiche du site web http ://nature.jardin.free.fr

textuelle est conforme à la représentation sémantique de la figure 2 : `<nom latin>` est retenu comme champ identifiant l'entité décrite dans chaque page ; `<nom commun>`, `<catégorie>`, `<feuillage>`, etc. sont des élaborations de `<nom latin>` et fournissent des propriétés de cette entité. Sur la figure 3, le concept `Gladiolus` est défini par des relations avec d'autres concepts : `Gladiolus` aCategorie `bulbe`, `Gladiolus` fleuritEn `printemps`, `Gladiolus` aFormeFeuillage `long`, etc. ; ou par des proprétés avec des valeurs : `Gladiolus` aHauteur « 0.60-1.40 », `Gladiolus` aNomCommun « Glaïeul hybride », etc. Lorsque le contenu d'un champ est une liste de syntagmes, soit la même relation est distribuée sur tous les items de la liste (e.g. fleuritEn dans `<floraison>`), soit plusieurs relations spécifiques s'appliquent sur ces items (e.g. `<feuillage>` contient des informations sur la forme et sur la persistance du feuillage, ce qui conduit à définir deux types de relations : aFormeFeuillage et aPersistanceFeuillage).

2.2 Méthode générale pour la construction d'une ontologie

Nous proposons une méthode générale pour construire une ontologie qui corresponde au domaine décrit par une collection de documents possédant les propriétés décrites en section 1. Dans cette ontologie, un *concept*

principal représente la classe des entités décrites dans chaque document, formalisées comme autant de *concepts pivots*. Dans l'exemple, `Plante` est le *concept principal* et `Gladiolus` est le *concept pivot* de la fiche de la Figure 3. L'ontologie est construite en trois étapes :

1. **Pré-traitement des documents** (section 3) : (a) interprétation manuelle de la mise en forme des documents de la collection pour déterminer leur structure, les champs identifiant le *concept principal* et ses propriétés, et (b) implémentation d'une chaîne de traitements produisant les documents balisés conformes à un même modèle.
2. **Construction du noyau d'ontologie** (section 4) : ce noyau est constitué du *concept principal*, de ses propriétés et des concepts requis pour les représenter. Les relations sont identifiées grâce aux noms des champs ; les concepts sont élicités à partir des valeurs des champs.
3. **Enrichissement du noyau d'ontologie** (section 5) : On extraie automatiquement de chaque document (a) le *concept pivot* représentant une sous-classe du *concept principal*, et (b) à partir de chaque champ, des propriétés du *concept pivot* représentés comme des restrictions de relations entre le *concept pivot* et les concepts du noyau.

3 Pré-traitement des documents

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="document">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="field" maxOccurs="unbounded">
      <xs:complexType>
       <xs:sequence>
        <xs:element name="fieldName" type="xs:string"/>
        <xs:element name="fieldValue" type="xs:string"
                                maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="type" type="xs:string"/>
      </xs:complexType>
     </xs:element>
    </xs:sequence>
   </xs:complexType>
 </xs:element>
 </xs:schema>
```

FIGURE 4 – Modèle de fiche selon la syntaxe des schémas XML

```
<document>
 <field>
   <fieldName>Nom commun</fieldName>
   <fieldValue>Glaïeul hybride</fieldValue>
 </field>
 <field type="entity">
  <fieldName>Nom latin</fieldName>
  <fieldValue>Gladiolus</fieldValue>
 </field>
 <field>
  <fieldName>Feuillage</fieldName>
  <fieldValue>Feuillage long</fieldValue>
  <fieldValue>Feuillage caduc</fieldValue>
 </field>
</document>
```

FIGURE 5 – Document XML correspondant à la fiche de la figure 3

De chaque document de la collection, on fournit une représentation XML unifiée qui met en évidence sa sémantique et facilite l'extraction

d'informations. Les documents XML générés sont conformes au modèle de la figure 4 : <fieldName> et <fieldValue> balisent respectivement le nom et chaque unité textuelle d'un champ, dont un possède l'attribut *type* et identifie le concept pivot. Ces fichiers XML sont obtenus en deux temps :

1. **Balisage selon les caractéristiques sémantiques de la mise en forme** : L'ontologue identifie l'ensemble des marqueurs typo-dispositionnels et lexicaux communs à tous les documents de la collection (casse, ponctuation, etc.) et nécessaires pour repérer le *concept pivot* et ses propriétés. Il définit ensuite un ensemble de règles d'extraction qui exploitent ces marqueurs (patrons ou règles pour les formats textuels, ou transformations XSLT pour les documents XML). Pour les pages HTML du site web nature.jardin.free.fr, le patron <B>.* ?</B> (codé sous forme d'expressions régulières avec Java *regex*) a servi à extraire les noms des champs. L'identifiant de l'entité de chaque page est tiré de `<nom latin>`.
2. **Découpage syntaxique du contenu des champs** : Le contenu des champs, souvent exprimé sous forme de listes, est segmenté en Unités Textuelles Elémentaires (UTE) à partir de patrons de segmentation de listes adaptés de (Luc *et al.*, 1999). Les UTE sont repérées par la ponctuation ou des marqueurs lexicaux tels que les conjonctions *et, ou*. Chaque UTE est supposée contenir une valeur de propriété. Pour réduire leur ambiguïté et faciliter leur annotation sémantique (Section 5), les UTE constituées d'un seul mot ou de groupes adjectivaux sont ré-écrites en distribuant le nom du champ sur leur contenu (sur la figure 5, les UTE de `<Feuillage>` sont « Feuillage long », « Feuillage caduc » plutôt que « long », « caduc »).

4 Construction d'un noyau d'ontologie

Les fichiers XML sont ensuite exploités pour produire un noyau d'ontologie OWL décrivant le *concept principal* et ses propriétés, ainsi que des hiérarchies de concepts nécessaires pour représenter les propriétés.

4.1 Sémantique des champs

L'ontologue définit des ensembles de champs en fonction du type de connaissances qu'ils contiennent : F_O, champs porteurs de relations entre individus (`objectProperties`); F_D, champs porteurs de relations vers des données (`dataTypeProperties`); F_L, champs porteurs de

termes dénotant le *concept pivot* (`rdfs:label`), et contenant au moins le nom du champ identifiant l'entité. Dans le document de la figure 3, `<feuillage>` $\in F_O$, `<hauteur>` $\in F_D$ et `<Nom commun>` $\in F_L$. Des traitements particuliers seront appliqués pour extraire des connaissances de ces champs.

Afin de conserver une trace du processus d'élaboration de l'ontologie, tout nouveau concept et toute propriété est documenté à l'aide d'une annotation (`owl:annotationProperty`) appelée `sourceField`, qui contient les noms des champs desquels cette connaissance a été tirée. Cette annotation sera utilisée lors de l'enrichissement du noyau d'ontologie. L'annotation `sourceField` de la relation `aCouleur` contient entre autres `<floraison>`, `<feuillage>` et `<couleur>`.

4.2 Champs indiquant des relations entre concepts

Un champ $f \in F_O$ comporte des relations entre le *concept principal* et un ou plusieurs concepts identifiés à partir des termes extraits du contenu du champ. Le noyau d'ontologie contiendra une représentation structurée de ces concepts sous la forme de hiérarchies construites comme suit :

- un concept de haut niveau est identifié et créé ; son label est adapté du nom du champ ; une de ses annotations `sourceField` vaut f ;
- s'il existe sur la Toile une ontologie décrivant ce concept, les concepts et relations pertinents sont importés de cette ontologie ;
- des règles XSLT permettent d'extraire toutes les UTE d'un même champ f dans tous les documents de la collection ;
- un extracteur de termes n-gram est appliqué sur l'ensemble des ETU ;
- l'ontologue sélectionne des termes pour définir concepts et labels ;
- il organise ces concepts dans une nouvelle hiérarchie, ou dans une hiérarchie réutilisée. Il peut créer de nouveaux concepts intermédiaires ou adapter des labels de concepts.

Le *concept principal* est ensuite relié aux concepts de haut niveau de ces hiérarchies par des relations (`ObjectProperties`) dont les noms sont tirés des noms des champs. Comme indiqué dans la section 2.1, dans le cas le plus simple, un champ contribue à définir une seule relation entre le *concept principal* et le concept de haut niveau d'une hiérarchie (e.g. le champ `<couleur>`). Le *concept principal* `Plante` est relié au concept `Couleur` par la propriété `aCouleur`. Les cas plus complexes sont ceux où les valeurs de champs font référence à plusieurs concepts, comme le champ `<feuillage>` de la Fig.3. Un concept de haut niveau appelé `Caracteristique-`

`Feuillage` est créé. Lancéolée (forme des feuilles), alternée (disposition des feuilles) et persistante (persistance des feuilles) sont des caractéristiques de feuilles de nature différente. Les concepts `FormeFeuillage`, `DispositionFeuillage`, `PersistanceFeuillage` sont alors créés comme sous-classes de `CaracteristiqueFeuillage`. La relation `aFeuillage` est également spécialisée en `aFormeFeuillage`, `aDispositionFeuillage` et `aPersistenceFeuillage`.

4.3 Champs indiquant des propriétés valuées ou des labels

Les champs de F_L contiennent des unités lexicales identifiées comme labels du *concept principal* (i.e. `Plante`). Ces termes peuvent correspondre à des labels exacts, des synonymes, des traductions, etc. .

Les champs de F_D fournissent des valeurs à partir desquelles des relations entre le *concept principal* et des littéraux peuvent être définies (i.e. `aHauteur` entre `Plante` et des valeurs numériques). Dans cet article, nous ne détaillons pas le processus d'extraction de ces valeurs.

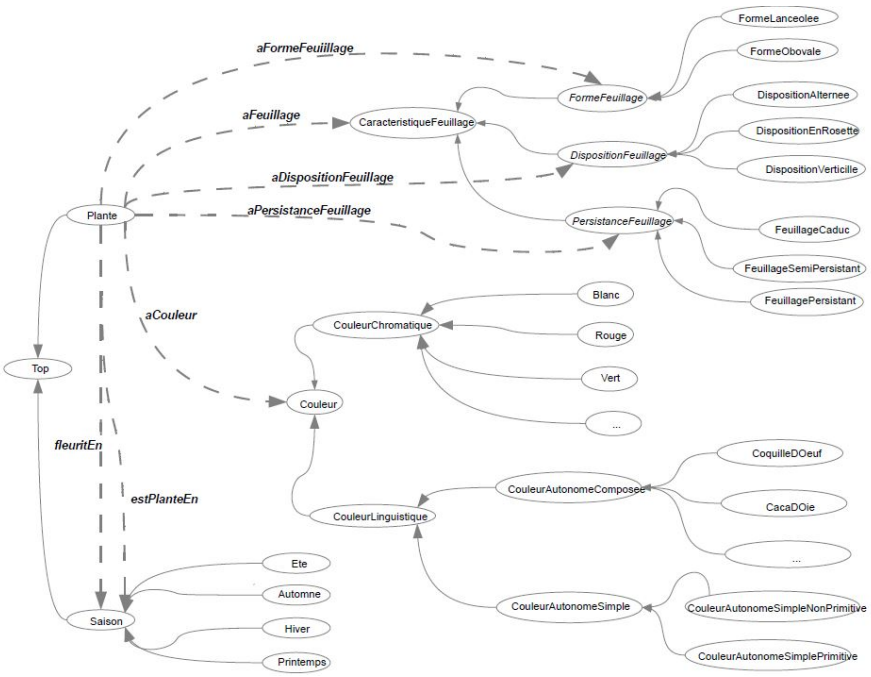

FIGURE 6 – Noyau de l'ontologie du Jardinage

Une partie du noyau d'ontologie produit à partir de la collection `http://nature.jardin.free.fr` est présenté sur la figure 6. Les traits pleins représentent des relations *est-un*, les traits en pointillés représentent les autres relations sémantiques.

5 Enrichissement de l'ontologie noyau

Enrichir le noyau consiste à définir de nouveaux concepts et ajouter des restrictions de relations à l'ontologie. Pour décrire ce processus, nous faisons appel aux définitions suivantes :

- D : ensemble de documents satisfaisant le schéma XML de la Fig.4,
- f : nom d'un champ d'un document XML,
- T_d^f : ensemble des UTE extraites du champ f du document d,
- T_d : ensemble des UTE extraites du document d,
- c_p^d : *concept pivot* du document d,
- c_m : *concept principal* de la collection,
- R_f : ensemble des relations portées par le champ f,
- $est-un(c_1, c_2)$: relation d'hyperonymie entre deux concepts (c_1 est sous-concept direct de c_2).

Le processus d'enrichissement utilise les fonctions suivantes :

- $creerConcept(c, I_d)$ crée un nouveau concept c identifié par I_d,
- $subsume(c_1, c_2)$ est vrai si c_2 est un concept fils de c_1,
- $construire(f, d)$ crée T_d^f,
- $ajouterLabel(c, l)$ ajoute le label l au concept c,
- $ajouterRelation(r(c_1, c_2))$ ajoute la relation r entre c_1 (domaine) et c_2 (co-domaine),
- $domaine(r)$ retourne le domaine de la relation r,
- $codomaine(r)$ retourne le codomaine de la relation r.

5.1 Enrichissement par des concepts

Pour chaque document, un *concept pivot* est créé. Il est ajouté à l'ontologie comme fils du *concept principal*, et tous les termes extraits des champs de F_L sont ajoutés comme des labels de ce nouveau concept. Ce processus correspond à l'algorithme suivant :

Pour tout $d \in D$;

T_d <- $\emptyset$

Pour tout $f \in F_L$: T_d^f = $construire(f, d)$; T_d = $T_d \cup T_d^f$

creerConcept (c_p^d, l_1) /* où l_1 est le premier label extrait du champ de type « entity » et contenu dans $\in T_d$ */
$ajouterRelation(est - un(c_p^d, c_m))$
Pour tout $l \in Td$: $ajouterLabel(c_p^d, l)$

5.2 Enrichissement par des restrictions de relations

Dans l'ontologie noyau, les relations associent le *concept principal* à des concepts de haut niveau. Ces relations s'appliquent aux sous-classes du *concept principal*. Cependant, les documents contiennent des informations (en particulier des propriétés) plus précises sur le *concept pivot* qu'ils décrivent. Ainsi, sur la figure 3, le concept `Gladiolus` (sous-classe de `Plante`) est lié au concept `Saison` par la relation `fleuritEn`. La fiche du Gladiolus suggère que `Gladiolus` devrait plutôt être relié par la relation `fleuritEn` à `Printemps` (sous-classe de `Saison`).

Cette étape vise donc la représentation automatique des restrictions de relations pour les ajouter à l'ontologie enrichie par les concepts. Le processus commence par l'identification des champs de F_O, puis annote chaque document par les concepts de l'ontologie enrichie par les *concepts pivots*, et finalement identifie les restrictions de relations. A l'aide des annotations `sourceField` de chaque `ObjectProperty`, on construit l'ensemble R_f des relations portées par chaque $f \in F_O$. Le système exploite les annotations sémantiques du contenu des champs de F_O. Pour tout concept c qui annote un segment textuel de f dans un document où c_p^d est le *concept pivot*, s'il existe dans l'ontologie une relation r de R_f dont le domaine est le *concept principal* et le codomaine est une super-classe de c, alors on ajoute à l'ontologie la restriction de r qui a pour domaine c_p^d et pour codomaine c, ce qui correspond à la formule suivante :

$\forall f \in F_O, \forall c$ annotant $f, \forall r \in R_f$
$est - un(c_p^d, domaine(r))$ et $subsume(codomaine(r), c)$
$\rightarrow ajouterRelation(r(c_p^d, c))$

6 Application et évaluation

6.1 Une ontologie du jardinage

Dans le projet MOANO [3], une ontologie est conçue pour faire de la recherche sémantique dans un guide de jardinage. Ce guide, rédigé en langue française, décrit une grande collection de plantes et de maladies de plantes [4]. Pour chaque plante, il fournit des informations sur son aspect, et donne quelques conseils de plantation et de protection contre les parasites et les maladies. L'ontologie est ensuite utilisée pour annoter le guide en utilisant le logiciel TextViz [5] (Reymonet *et al.*, 2009). L'ontologie doit donc être consacrée aux plantes non d'un point de vue botanique ou scientifique, mais du point de vue d'un jardinier. L'ontologie doit être représentée avec le standard OWL-DL du W3C.

La source de connaissances pour construire cette ontologie aurait pu être le guide lui-même. Pour disposer d'une connaissance plus consensuelle, nous avons décidé de combiner plusieurs sources de données. Nous avons envisagé de réutiliser des ontologies du domaine de la botanique disponibles sur le web. Mais ces ontologies (par exemple, l'ontologie Plant [6] ou bien le concept « Plante » et ses instances dans DBpedia [7] ou Dmoz [8]) reflètent un point de vue scientifique, décrivant les plantes par leur taxonomie (règne, ordre, genre, famille, espèce, etc.). Même si nous aurions pu réutiliser ces concepts, nous avions besoin d'ajouter des propriétés (conseils) sur la manière d'entretenir chaque variété de plante, au moins celles mentionnées dans le guide. Enfin, l'annotation sémantique avec TextViz exploite les concepts d'une ontologie et non les instances. Ceci implique que chaque variété de plante soit représentée comme une classe. Pour toutes ces raisons, nous avons recherché des sources de connaissances supplémentaires pour générer l'ontologie du jardinage.

Plusieurs sites web sont dédiés au jardinage. Parmi les sites francophones, nous avons retenu le site `Jardin! L'encyclopédie` [9] qui

3. MOANO est le projet ANR-2010-CORD-024-03 dont l'IRIT, le LIUPPA,le LIG et le LIFL sont partenaires. http ://moano.liuppa.univ-pau.fr

4. http ://www.vilmorin.fr/ présente une partie du guide en ligne.

5. TextViz a été développé dans le cadre du projet Dynamo, financé par le programme TechLog de l'ANR, http ://www.irit.fr/DYNAMO

6. http ://plantology.org

7. http ://dbpedia.org

8. http ://www.dmoz.org

9. http ://nature.jardin.free.fr/

fournit des descriptions très complètes des plantes de jardin. L'ontologie de jardinage a donc été construite à partir de la collection de documents structurés composant ce site. Le noyau d'ontologie contient le concept principal `Plante` et sept hiérarchies de concepts construites à partir des champs `<port>`, `<croissance>`, `<couleur>`, `<exposition>`, `<feuillage>`, `<entretien>` et `<sol>`. Vingt sept relations relient `Plante` aux concepts de haut niveau de ces hiérarchies. Des concepts et des propriétés spécifiques au domaine de la botanique ont été ajoutés, comme la taxonomie des plantes (règnes, ordres, espèces, etc.). Ces concepts et propriétés ont été adaptés d'ontologies comme GeoSpecies [10].

6.2 Evaluation et discussion

L'évaluation concerne uniquement le processus d'enrichissement par les relations, l'enrichissement par les concepts ne présentant pas de difficultés majeures. L'idée est de comparer les restrictions de relations obtenues manuellement à celles obtenues automatiquement. Ainsi, 1000 documents de « Jardin !L'encyclopédie » ont été traités pour produire des sous-classes de *Plante* qui ont été ajoutées au noyau d'ontologie et forment une ontologie de base. Parmi ces 1000 documents, 20 ont été choisis au hasard pour permettre à un ontologue de compléter cette ontologie par des restrictions de relation. L'ontologie ainsi enrichie constitue une ontologie de référence, correspondant aux résultats attendus d'une extraction correcte de relations.

Par ailleurs, l'ontologie de base a été enrichie selon le processus d'ajout de restrictions de relations à partir des mêmes 20 documents. L'ontologie obtenue est alors comparée à l'ontologie de référence. L'évaluation porte sur le nombre de restrictions définies avec un domaine et un codomaine appropriés, celles omises ou mal identifiées. Nous avons utilisé deux mesures : le rappel et la précision.

Sur les 27 relations présentes dans le noyau, 248 restrictions ont été correctement détectées, 76 restrictions n'ont pas été trouvées, et 62 restrictions ont été mal détectées. Nous avons obtenu un Rappel de 0.76 et une Précision de 0.8. Bien que ces résultats soient encourageants, nous envisageons de traiter plusieurs problèmes pour améliorer le processus. Certaines restrictions ne sont pas trouvées car le processus de découpage (« chunking ») n'est pas totalement fiable et certains labels sont omis. D'autres relations sont manquées car les intervalles de valeurs ne sont pas traités. Par

10. http ://www.geonames.org/ontology/ontology_v3.1.rdf

exemple, dans le champ `<floraison>`, l'expression de « juin à août » donnera lieu aux restrictions `Gladiolus` ***fleuritEn*** `Juin` et `Gladiolus` ***fleuritEn*** `Août`, sans mentionner les mois intermédiaires. D'autres problèmes conduisent à des relations erronées. Ainsi, l'annotation utilisant un algorithme de stemming, peut mal identifier des concepts à partir du texte dans le cas de termes ambigüs ou d'orthographes proches. Par exemple, plusieurs relations `aCouleurFeuillage` sont fausses, car les termes *feuille* et *feuillage*, très présents dans le corpus, sont annotés par le concept `Feu` appartenant à la hiérarchie des couleurs. Par ailleurs, des erreurs surviennent lorsque des relations ayant la même signature (mêmes domaine et co-domaine) peuvent être identifiées au sein d'un même champ. Dans ce cas, toutes les relations sont générées, même si elles sont contradictoires, comme les relations *nécessite* et *craint* qui relient `Plante` et `Sol`. Le fait de ne pas traiter les négations conduit également à produire des relations contraires à ce que dit le document. Un dernier problème survient lorsque le contenu d'un champ ne correspond pas à la sémantique attendue. Par exemple, dans le champ `<floraison>`, l'expression « odorante en juillet » sera interprétée comme `Gladiolus` ***fleuritEn*** `Juillet`.

6.3 Perspectives

Ces problèmes justifient de mieux prendre en compte les indices linguistiques, de traiter les négations ainsi que les intervalles de valeur pour identifier plus de relations présentes dans chaque champ. Pour gérer les négations, nous envisageons de réutiliser des travaux de Benamara *et al.* (2012) qui proposent une typologie des négations sur la base des opérateurs de négations, de quantificateurs négatifs et des négations lexicales. Pour gérer les intervalles, nous nous baserons sur les travaux menés sur la représentation des ensembles flous (Buche *et al.*, 2013).

Une réflexion doit ensuite être menée pour extraire plus de connaissances des grands champs rédigés. L'application de patrons lexico-syntaxiques après une analyse syntaxique serait plus efficace pour détecter des relations. Cependant, la structure doit être prise en compte pour identifier les arguments des relations non explicités. De plus, dans les parties rédigées, l'analyse syntaxique peut échouer car beaucoup d'UTE sont des syntagmes nominaux. Par exemple, dans le champ `<entretien>`, la 2e phrase est une liste d'actions pour lesquelles les relations de dépendances ne sont identifiables par aucun analyseur syntaxique. Même l'interprétation de la 1ère phrase (*arroser régulièrement en ajoutant un fertilisant tous les 15 jours*) est délicate pour déterminer si l'indication de temps porte sur

le verbe principal (*arroser*) ou sur le complément (*ajouter un fertilisant*).

Enfin, nous envisageons de comparer notre approche et des approches linguistiques ou statistiques classiques pour faire ressortir sa contribution positive. L'évaluation portera sur de plus grands ensembles de données dans le contexte des campagnes d'évaluation de BioNLP. Ainsi, la tâche GRO de la campagne 2013 offre à la fois une ontologie et un corpus scientifique. La tâche consiste à extraire les relations et leurs arguments à partir de textes. Deux copies annotées du corpus sont disponibles et peuvent être utilisées comme ensembles d'entrainement/d'apprentissage. Nous exécuterons le processus d'extraction présenté dans cet article et nous comparerons les concepts et restrictions obtenus avec les annotations du corpus.

7 Conclusion

L'originalité de notre approche est liée au fait qu'elle combine l'analyse du langage et de la structure des documents pour identifier des concepts, et l'annotation sémantique pour identifier des relations. Elle contribue à l'extraction de relations aussi bien pour construire que pour peupler des ontologies. Nous avons exploré certains apports de l'exploitation de la mise en forme textuelle, et de la sémantique qu'elle véhicule, pour identifier des relations qui auraient été omises par la seule analyse du langage dans les textes. La structure des documents peut être explicitée et intégrée sous forme d'indices pour extraire des relations précises. Cette approche est utilisable en plus des techniques d'extraction à partir du langage naturel comme les patrons linguistiques ou l'apprentissage automatique.

Remerciements

Nous remercions nos collègues du LIUPPA, partenaires du projet MONAO, (M.-N. Bessagnet, A. Royer et C. Sallabery) pour leur avis précieux lors de la construction et de l'enrichissement de l'ontologie.

Références

AUSSENAC-GILLES N. & JACQUES M.-P. (2008). Designing and Evaluating Patterns for Relation Acquisition from Texts with Caméléon. *Terminology, Pattern-Based approaches to Semantic Relations*, **14**(1), 45–73.

BENAMARA F., CHARDON C., MATHIEU Y., POPESCU V. & N. A. (2012). How do negation and modality impact on opinions ? In *Extra-propositional aspects of meaning in computational linguistics - Workshop at ACL 2012*, p. 8–17.

BUCHE P., DIBIE-BARTHÉLEMY J., IBANESCU L. & SOLER L. (2013). Fuzzy web data tables integration guided by an ontological and terminological resource. *IEEE Trans. Knowl. Data Eng.*, **25**(4), 805–819.

CIMIANO P., SCHMIDT-THIEME L., PIVK A. & STAAB S. (2004). Learning taxonomic relations from heterogeneous evidence. In P. BUITELAAR, P. CIMIANO & B. MAGNINI, Eds., *Ontology Learning from Text : Methods, Applications and Evaluation*, p. 59—73. IOS Press.

GROZA T., HANDSCHUH S., MÖLLER K. & DECKER S. (2007). Salt - semantically annotated latex for scientific publications. In *Proceedings of the 4th European Semantic Web Conference (ESWC 2007)*, volume 4519 of *LNCS*, p. 518 – 532, Berlin, Heidelberg : Springer-Verlag.

HEARST M. A. (1997). Texttiling. segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, **23**(1), 33–64.

HEARTS M. (1992). Automatic acquisition of hyponyms from large text corpora. In *Procs. of the 14th International Conference on Computational Linguistics COLING1992*, p. 539–545.

J. O'CONNOR, M. K. & DAS A. (2011). Acquiring owl ontologies from xml documents. In *Interna-tional Conference on Knowledge Capture. K-CAP 2011*.

KAMEL M. & AUSSENAC-GILLES N. (2009). How can document structure improve ontology learning ? In *Workshop on Semantic Annotation and Knowledge Markup collocated with K-CAP 2009 (SAAKM 2009)*, p. 1–8.

LUC C., MOJAHID M., VIRBEL J., GARCIA-DEBANC C. & PERRY-WOODLEY M.-P. (1999). A linguistic approach to some parameters of layout : A study of enumerations. In *AAAI 99*.

MANN W. C. & THOMPSON S. A. (1988). Rhetorical structure theory : Toward a functional theory of text organization. *Text*, **8**(3), 243–281.

MONTIEL-PONSODA H. & AGUADO DE CEA G. (2010). *Using natural language patterns for the development of ontologies*, In V. BHATIA, P. SÁNCHEZ HERNÁNDEZ & P. PÉREZ PAREDES, Eds., *Researching specialized languages*, p. 211–230. John Benjamins Pub.

NAVIGLI R. & VELARDI P. (2006). Ontology enrichment through automatic semantic annotation of on-line glossaries. In *15th International Conference EKAW 2006*, volume LNCS 4248, p. 126–140 : Springer.

POELMANS J., ELZINGA P., VIAENE S. & DEDENE G. (2010). Formal concept analysis in knowledge discovery : a survey. In *Proceedings of the 18th international conference on Conceptual structures : from information to intelligence*, ICCS'10, p. 139–153, Berlin : Springer-Verlag.

POWER R., SCOTT D. & BOUAYAD-AGHA N. (2003). Document structure. *Computational Linguistics*, **29**(2), 211–260.

REYMONET A., THOMAS J. & AUSSENAC-GILLES N. (2009). Ontology based information retrieval : an application to automotive diagnosis. In *Workshop on Principles of Diagnosis (DX 2009)*, p. 9–14.

ROLE F. & ROUSSE G. (2006). Construction incrémentale d'une ontologie par analyse du texte et de la structure du document. *Document numérique*, **9**(1), 77–91.

SCHUTZ A. & BUITELAAR P. (2005). Relext : A tool for relation extraction from text in ontology extension. In *4th International Semantic Web Conference (ISWC 2005)*, volume 3729, p. 593–606 : Springer : Berlin.