Peuplement d'une ontologie guidé par l'identification d'instances de propriété

Driss Sadoun, Catherine Dubois, Yacine Ghamri-Doudane, Brigitte Grau

Peuplement d'une ontologie guidé par l'identification d'instances de propriété

Driss Sadoun[1,2] **Catherine Dubois**[3,4] **Yacine Ghamri-Doudane**[5] **Brigitte Grau**[1,3]

[1]LIMSI/CNRS, B.P. 133 91403 Orsay Cedex, France
prénom.nom@limsi.fr
[2]Université Paris-Sud, 91400 Orsay, France
[3]ENSIIE, 1 square de la résistance, 91000 Evry, France
prénom.nom@ensiie.fr
[4]CNAM-CEDRIC, 292 Rue St Martin FR-75141 Paris Cedex 03, France
[5]Laboratoire L3i, Université de La Rochelle, Av. Michel Crépeau, 17042, La Rochelle CEDEX 1, France.
prénom.nom@univ-mlv.fr

Résumé

Dans le but de formaliser des spécifications d'exigence écrites en langage naturel, nous avons choisi de modéliser les connaissances du domaine par une ontologie et de représenter formellement les spécifications par son peuplement. L'approche de peuplement est centrée sur l'identification d'instances de propriétés à partir des textes. Pour cela, des règles d'extraction sont acquises automatiquement à partir d'un corpus d'apprentissage, puis appliquées sur les textes pour l'identification de mentions d'instances de propriété représentées par des triplets. Ces règles exploitent les niveaux d'analyse lexicale, syntaxique et sémantique et sont engendrées à partir des chemins syntaxiques récurrents entre les termes pouvant dénoter des instances de concept ou de propriété. Nous montrons que l'identification d'instances de propriétés permet d'identifier de façon précise les instances de concepts énoncées de façon explicite ou implicite dans les textes.

1 Introduction

Tout processus de réalisation d'un système repose sur une phase de spécification des exigences. La vérification de la correction et la consistance de ces spécifications nécessitent l'utilisation de méthodes formelles qui peuvent être appliquées uniquement sur des spécifications formelles.

Or, en pratique, les spécifications d'exigences sont le plus souvent rédigées en langage naturel (LN) (Mich *et al.*, 2004) et sont donc non formelles. Leur vérification nécessite alors de les transformer en spécifications formelles. Se pose ainsi naturellement la question de l'automatisation du passage entre spécifications LN et spécifications formelles. Cette problématique n'est pas récente et a été abordée par différentes approches (Fougères et Trigano, 1997; Ilic, 2007; Ilieva et Boley, 2008; Kof, 2010; Bajwa *et al.*, 2012; Guissé *et al.*,). L'ensemble des approches pointe la difficulté d'une transformation directe et la nécessité de passer par un modèle intermédiaire palliant l'écart entre spécifications LN et spécifications formelles. Les modèles intermédiaires proposés dans la littérature sont en général semi-formels, comme UML ou SBVR.

Dans le cadre du projet ENVIE VERTE [1] dans lequel se place le travail décrit dans cet aricle, nous avons choisi de modéliser les connaissances du domaine par une ontologie en OWL-DL, une version décidable d'OWL2 et de représenter formellement les spécifications par son peuplement.

Une ontologie modélise les concepts et leurs propriétés, définissant ainsi le vocabulaire conceptuel d'un domaine. Un concept est la description d'un ensemble d'individus (d'instances) ayant une sémantique et des propriétés communes. Une instance de concept est une concrétisation d'un concept. Par exemple, dans notre modélisation du domaine *kitchen* correspond à une instance du concept *Location* [2]. Une propriété est définie entre deux concepts, i.e. une relation, ou entre un concept et un type de donnée, i.e un attribut. Une instance de propriété relie donc deux instances de concepts, par exemple *Occured-in(movement,kitchen)*, ou une instance de concept et une valeur d'attribut, par exemple *Has-value*(temperature,25). Une ontologie offre un cadre formel permettant d'associer

1. financé par DIGITEO, projet DIM LSC 2010.

2. Les exemples sont en anglais, car les textes que nous analysons sont rédigés dans cette langue

une sémantique aux termes issus des textes. L'association des instances de concepts et propriétés à leurs formulations dans les textes est régie à l'aide d'une ontologie lexicale en SKOS (Simple Knowledge Organization System) contenant la terminologie liée au vocabulaire conceptuel.

Peupler une ontologie consiste à y ajouter de nouvelles instances sans en changer la structure conceptuelle (Petasis *et al.*, 2011). Ces nouvelles instances sont associées aux concepts et propriétés reconnus dans les textes. L'identification peut être centrée sur la reconnaissance d'instances de concepts (Thongkrau et Lalitrojwong, 2012), ou sur la reconnaissance d'instances de relation (Nakamura-Delloye et Stern, 2011).

Dans cet article, nous proposons de guider le peuplement de l'ontologie par l'identification de triplets de termes dans les textes correspondant à des instances de propriétés. Nous distinguons deux types de triplets : complets et partiels. Les triplets complets contiennent la mention d'une instance de propriété ainsi que les mentions des deux instances de concepts qu'elle lie. En revanche, les triplets partiels ont pour vocation de reconnaître des propriétés pour lesquelles une des deux instances n'est pas explicitement mentionnée. Guider le peuplement de l'ontologie par identification d'instances de propriétés permet de résoudre des cas d'ambiguïté de termes et d'informations implicites pour lesquels les concepts associés sont trouvés par inférence dans l'ontologie à partir des instances de propriétés. De la sorte, l'identification d'instances de concepts ne repose pas seulement sur la reconnaissance de leurs mentions dans les textes mais aussi sur les propriétés conceptuelles auxquelles elles sont associées.

Les triplets correspondant aux instances de propriétés sont extraits à l'aide de règles acquises par amorçage à partir d'un corpus d'apprentissage et d'une terminologie de départ. Ces règles correspondent à des formes lexico-syntaxiques récurrentes. Nous montrons que cette approche permet d'identifier de manière fiable des instances de propriétés, et dans un deuxième temps d'inférer les instances de concepts qui leurs sont associées. De plus, l'approche que nous proposons peut s'adapter aisément à d'autres domaines d'application dès lors que l'on peut le décrire par une ontologie.

2 Travaux connexes

Le peuplement d'ontologie consiste à identifier et à classer les instances extraites des textes. Se pose le problème de la reconnaissance de mentions d'instances de concepts et de propriétés dans les textes. L'hypothèse généralement faite est que les paires d'entités apparaissant dans un même contexte peuvent être considérées comme des instances de la même relation. La définition du contexte peut être restreinte par la présence d'un verbe et la reconnaissance de son entourage (Lin et Pantel, 2001; Makki *et al.*, 2008). D'autres travaux se fondent sur la classification entre couples d'entités connues pour être liées par une relation sémantique (Hasegawa *et al.*, 2004; Nakamura-Delloye et Stern, 2011; Thongkrau et Lalitrojwong, 2012). La majorité des approches exploitent des connaissances lexicales et syntaxiques pour la définition d'un contexte représentant une relation sémantique. Cependant, dans (IJntema *et al.*, 2012), les auteurs avancent que les patrons lexico-sémantiques sont plus à même de capturer dans les textes le contexte sémantique. Alors qu'ils proposent un langage d'écriture manuelle de règles, nous proposons d'acquérir ce type de règle automatiquement, à partir des trois niveaux de connaissances lexicale, syntaxique et sémantique.

L'ensemble des approches identifient les instances de concept au niveau du texte. La méthode que nous proposons tire partie des connaissances sémantiques pour inférer au sein de l'ontologie l'appartenance d'une instance à un concept. De plus, nous proposons d'aller plus loin que la classification d'instances en identifiant dans les textes les mentions dénotant un même individu.

3 Notations

Dans la suite de l'article nous emploierons les notations suivantes. Les noms de concept et de propriété sont en *italique* et commencent par une majuscule. Les noms d'instances de concept sont en *italique* et commencent par une minuscule. Un concept sera noté C_i et les instances de concept i_{C_i}. Les propriétés sont définies sur un domaine et une image [3]. Les propriétés entre concepts sont notées $P_k(C_i, C_j)$ et les instances de propriété

3. Le domaine contient un concept ou un ensemble de concepts et l'image contient soit un concept ou un ensemble de concepts ou un littéral (entier, chaîne de caractères, ...)

sont notées $P_k(i_{C_i}, i_{C_j})$. Dans les textes, un triplet contenant la mention d'une instance de propriété, t_P, entre deux instances de concepts est notée (t_P, t_{C_i}, t_{C_j}) avec t_{C_i} et t_{C_j} les termes qui dénotent respectivement les instances du concept C_i et du concept C_j.

4 Acquisition des règles d'extraction

Dans les textes, les instances de concepts et de propriétés sont dénotées par des termes. Guider l'identification d'instances de concepts à partir des propriétés qui les définissent requiert de reconnaître des triplets (t_P, t_{C_i}, t_{C_j}). Aussi, chaque règle d'extraction a comme objectif de reconnaître la mention d'une propriété de l'ontologie et d'extraire du texte les triplets de termes qui la dénotent. Deux types de triplets sont à reconnaître :

- triplet complet : les mentions d'instances des domaine et image de la propriété sont explicites dans le texte, soit (t_P, t_{C_i}, t_{C_j}) ;
- triplet partiel : l'une des mentions d'instances des domaine ou image de la propriété n'est pas explicite. Elle correspondra à une inconnue dans le triplet, soit $(t_P, ?i, t_{C_j})$ ou $(t_P, t_{C_i}, ?i)$.

Par exemple à partir de la phrase : *when a person moves into the kitchen, switch on the light.*, on peut identifier le triplet $(Occured\text{-}in, move, kitchen)$ qui dénote une instance de la propriété $Occured\text{-}in$ liant une instance du concept $Phenomenon$ à une insatnce du concept $Location$. Néanmoins dans cette même phrase, l'agent qui doit allumer la lumière n'est pas explicité. Le triplet à identifier dans ce cas est $(Turn\text{-}on$[4]$, ?A, light)$ qui dénote une instance de la propriété $Turn\text{-}on$ liant une instance non mentionnée du concept *Actuator* à une instance du concept *Physical-process*.

4.1 Méthode d'acquisition des règles d'extraction

L'acquisition des règles d'extraction se fait automatiquement à partir d'un corpus d'apprentissage et d'une termino-ontologie amorce. Elles sont acquises à partir des chemins syntaxiques les plus fréquents entre des paires de termes. Ces termes sont issus de classes sémantiques connues pour être liées dans l'ontologie. L'extraction des deux types de triplets nécessite l'acquisition de deux types de règles d'extraction.

Dans le cas de triplets partiels, les règles sont acquises à partir des chemins les plus fréquents entre les termes qui dénotent les instances d'une propriété et les instances de son domaine ou de son image. Pour les triplets complets, les chemins sont constitués des deux chemins partiels entre paires de termes dénotant les instances de domaine et propriété et celles dénotant propriété et image.

L'algorithme 1 décrit le processus d'acquisition des règles. Chaque phrase du corpus est analysée et son arbre de dépendances syntaxiques est engendré. Puis trois fonctions sont appelées pour l'extraction de chemins syntaxiques entre termes. Elles permettent d'identifier des chemins liant trois types d'ensemble de termes : les triplets de termes d'une propriété, de son domaine et de son image, les paires de terme d'une propriété et de son image et les paires de terme d'une propriété et de son domaine. Les chemins extraits sont comparés en fonction de leurs dépendances et des formes lemmatisées des termes. Les chemins identiques les plus fréquents pour chaque type de paires sont retournés. Enfin, les règles d'extraction sont générées à partir des caractéristiques des chemins retournés (cf. section 4.3).

4.2 Chemin syntaxique

L'analyse syntaxique des phrases permet d'en extraire des arbres de dépendances syntaxiques cf. figure 1. Chaque nœud est étiqueté par un terme et sa catégorie morpho-syntaxique (nom, verbe, etc). Les nœuds sont reliés deux à deux par des dépendances syntaxiques qui constituent des liens orientés. Un chemin syntaxique est composé des dépendances syntaxiques liant deux termes dans un arbre de dépendances. La figure 1 représente l'arbre de dépendances de la phrase *"When a person moves into the kitchen, switch on the light."* [5]

Par exemple, le chemin syntaxique entre le terme *moves* dénotant une instance du concept *Phenomenon* et le terme *kitchen* dénotant une instance du concept *Location* est $prep(moves, into)$-$pobj(into, kitchen)$ (chemin en gras, figure 1).

4.3 Génération des règles d'extraction

Les règles d'extraction ont pour rôle d'identifier des instances de propriétés. Ainsi, elles

4. Turn−on est la dénomination préférée de swich on

5. produit à l'aide de DependenSee.jar http ://chaoticity.com/dependensee-a-dependency-parse-visualisation-tool/

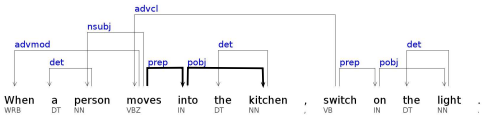

FIGURE 1: Arbre de dépendances syntaxiques

modélisent le contexte sémantique dans lequel les termes désignant des instances de concepts apparaissent. La génération de règles d'extraction est automatique. Elle exploite les caractéristiques issues des n chemins syntaxiques identiques les plus fréquents. Ces caractéristiques sont de trois types :

- dépendances syntaxiques ;
- catégorie termino-ontologique (sémantique) ;
- catégorie morpho-syntaxique.

Données: corpus, ontologie, skos
$Paths \leftarrow \emptyset$;
Variables : Ensemble de termes T_D, T_P, T_I;
pour *chaque phrase S du corpus* **faire**
  $Tr \leftarrow getDependencyTree(S)$;
  $C \leftarrow getConcepts(ontologie)$;
  for *each c in C* **do**
    $T_D \leftarrow getTerms(c, skos)$;
    $P \leftarrow getPropertiesOf(c, ontologie)$;
    for *each p in P* **do**
      $T_P \leftarrow getTerms(p, skos)$;
      $Img \leftarrow getImagesOf(p, ontologie)$;
      for *each img in Img* **do**
        $T_I \leftarrow getTerms(img, skos)$;
        // - instances de propriété complète -
        $Paths \leftarrow Paths \cup$ extrPaths(Tr,T_P,T_D, T_I);
        //- instances de propriété partielle -
        //domaine implicite
        $Paths \leftarrow Paths \cup$ extrPaths(Tr,T_P,T_I);
        //image implicite
        $Paths \leftarrow Paths \cup$ extrPaths(Tr,T_P,T_D);
      end
    end
  end
fin
$Chemins \leftarrow getMostFrequent(Paths)$;
pour *chaque ch de $Chemins$* **faire**
  $createCorrespondingRule(ch)$;
fin

Algorithm 1: Acquisition des règles d'extraction

Exemple : l'un des chemins récurrents entre les deux ensembles de termes des concepts *Phenomenon* et *Location* qui sont dans l'ordre le domaine et l'image de la propriété *Occured-in* est $prep(t_P, t_O) \wedge pobj(t_O, t_L)$, avec :

- les dépendances syntaxiques du chemin
  - $prep(t_P, t_O) - pobj(t_O, t_L)$
- les catégories morpho-syntaxiques
  - t_P est un verbe ;
  - t_L est un nom ;
  - t_O est une préposition ;
- les catégories termino-ontologiques
  - t_P dénote le concept *Phenomenon* ;
  - t_O dénote la propriété *Occured-in* ;
  - t_L dénote le concept *Location* ;

Les dépendances syntaxiques sont transformées en prédicats, les catégories morpho-syntaxiques et termino-ontologiques sont transformées en contraintes. Cela permet d'engendrer la règle d'extraction de la propriété $Occured\text{-}in(Phenomenon, Location)$ avec $T_{Location}, T_{Occured-in}, T_{Phenomenon}$ trois ensembles de termes issus de l'ontologie lexicale :

$prep(t_P, t_O) \wedge pobj(t_O, t_L) \wedge isPrep(t_O) \wedge isVerb(t_P) \wedge isNoun(t_L) \wedge T_{Location}(t_L) \wedge T_{Occured-in}(t_O) \wedge T_{Phenomenon}(t_P) \rightarrow Occured\text{-}in(t_P, t_L)$

5 Expansion de la terminologie

La terminologie est représentée en SKOS. Ce formalisme permet de définir pour chaque terme sa formulation préférée ainsi que sa liste de formulations synonymes.

Afin d'augmenter la terminologie amorce, nous appliquons les règles acquises pour extraire de nouveaux termes sur le corpus d'apprentissage. Chacune des règles engendre trois applications. Chaque application a comme objectif l'extraction de termes dénotant un ensemble sémantique représenté dans la règle. Soit R une règle d'extraction et $T_{Domaine}, T_{Image}, T_{Prop}$ trois ensembles de termes. R peut alors être considérée comme une fonction définie comme suit :

$R : T_{Domaine} \times T_{Prop} \rightarrow T_{Image}$
$R : T_{Image} \times T_{Prop} \rightarrow T_{Domaine}$
$R : T_{Domaine} \times T_{Image} \rightarrow T_{Prop}$

La pertinence d'un terme t extrait par des règles R_i est calculée selon la formule suivante :

$Pertinence(t) = (\sum_{i=1}^{n} freq(t, R_i)) * n$

avec $freq$ sa fréquence, et n le nombre de règles du même type à partir desquelles il est extrait. Cette formule permet de favoriser les termes extraits par plusieurs règles.

6 Peuplement de l'ontologie

Suite à l'identification des instances de propriété, vient la phase de classification des instances de concepts. À ce stade, les classes sémantiques des instances liées par les propriétés ne sont pas encore connues. La classification de ces instances est fondée sur le raisonnement permis par OWL sur leurs propriétés. De cette façon, les ambiguïtés liées aux termes sont résolues et les mentions d'instances implicites dans les textes sont déduites à partir des propriétés possédées par les instances de concept. Classer les instances dans l'ontologie ne suffit pas pour représenter de manière cohérente les connaissances issues des textes. Pour cela, il faut aussi être en mesure d'identifier de manière univoque chaque individu.

6.1 Classification des instances

La classification des individus consiste à les associer aux concepts qui les dénotent. Nous présentons dans cette section comment différents mécanismes d'inférence de OWL peuvent être exploités à cette fin.

6.1.1 Domaine et Image des propriétés

Lors du raisonnement sur les instances de l'ontologie, chaque instance qui participe à une propriété est associée à la classe sémantique du domaine ou de l'image de la propriété [6] de la manière suivante : Soit P_1 une propriété avec comme domaine D_1 et comme image I_2 et soit i_1 et i_2 deux instances de concepts. Alors si $P_1(i_1, i_2)$ on déduit : $i_1 \in D_1$ et $i_2 \in I_2$

6.1.2 Condition Nécessaire et Suffisante

OWL permet de définir des équivalences entre les concepts et certaines de leurs propriétés, formant des conditions nécessaires et suffisantes pour inférer le concept auquel appartient une instance à partir de ses propriétés. Par exemple l'axiome $C_1 \equiv P_2.C_2 \sqcap P_3.C_3$ définit une équivalence entre le concept C_1 et ses deux propriétés P_2 et P_3 ayant pour images dans l'ordre C_2 et C_3. Cet axiome permet l'inférence suivante : Soit $i_{C_2} \in C_2$ et $i_{C_3} \in C_3$ alors $P_2(i, i_{C_2}) \wedge P_3(i, i_{C_3}) \rightarrow i \in C_1$

6. Par défaut la classe sémantique est *Thing*

6.2 Identification des instances

Dans les ontologies en OWL, les instances identiques sont identifiées à l'aide de la propriété *SameAs*. L'inférence de *SameAs* exploite les propriétés qui, à l'instar des clés primaires que l'on trouve en base de données, permettent d'identifier de manière unique les individus. Les propriétés représentant une contrainte d'unicité peuvent être définies de deux manières.

6.2.1 Propriété fonctionnelle et inverse fonctionnelle

Une *propriété fonctionnelle* associe à chaque individu du domaine, un seul individu de l'image. Cela permet l'inférence suivante : Soit P_n une propriété fonctionnelle, i_i, i_j et i_k trois individus,

$P_n(i_i, i_j) \wedge P_n(i_i, i_k) \rightarrow SameAs(i_j, i_k)$

La *propriété inverse fonctionnelle* a, pour chaque individu de l'image, un seul individu de domaine possible. Cela permet l'inférence suivante : Soit P_n une propriété inverse fonctionnelle, i_i, i_j et i_k trois individus,

$P_n(i_j, i_i) \wedge P_n(i_k, i_i) \rightarrow SameAs(i_j, i_k)$

6.2.2 Contraintes en SWRL

Dans certains cas, plus d'une propriété est nécessaire pour représenter une contrainte d'unicité, par exemple une personne est identifiée par ses nom, prénom et date de naissance. Ce type de contrainte doit définir les propriétés que deux individus doivent nécessairement partager pour être inférés comme semblables. Ce type de contraintes peut être défini à l'aide de règles SWRL. Par exemple, la règle ci-dessous énonce que les deux individus i_x et i_y sont un même individu s'ils possèdent des valeurs similaires à travers les propriétés P_1 et P_2,

$P_1(i_x, i_{C_1}) \wedge P_1(i_y, i_{C_1}) \wedge P_2(i_x, i_{C_2}) \wedge P_2(i_y, i_{C_2}) \rightarrow SameAs(i_x, i_y)$

7 Expérimentation

Dans cette section, nous présentons le domaine d'application, les ressources utilisées, ainsi que les évaluations d'acquisition de termes et d'extraction d'instances de propriétés. Enfin, nous donnons un exemple de création, de classification et d'identification d'instances de concept à partir d'une phrase extraite de notre corpus de test.

7.1 Environnement intelligent

Un environnement intelligent est un ensemble d'objets communicants (capteurs, actionneurs et processus de contrôle), dont le comportement général est décrit ci-dessous :

- Un capteur détecte l'occurrence d'un type phénomène ou mesure un type phénomène dans une zone restreinte.
- Un capteur détecte ou mesure un type de phénomène s'il est localisé dans sa zone de capture.
- Un actionneur est connecté à un appareil (processus physique) de l'environnement qu'il peut actionner.
- La détection d'un phénomène peut conduire à l'activation d'un ou de plusieurs actionneurs et actionner une ou plusieurs actions (turn on, turn off, decrease ou increase) sur les processus physique qu'ils contrôlent.
- Un actionneur, pour être activé par un capteur, doit partager son type et être localisé dans sa zone de contrôle.

Afin de modéliser ce domaine nous avons défini une ontologie de haut niveau contenant 12 concepts, 15 propriétés entre concepts et 9 entre concepts et types de données. Cette ontologie comporte des instances initiales qui correspondent à un environnement physique d'un utilisateur, qui n'est pas amené à être modifié par celui-ci. Cette ontologie est suffisante pour représenter notre domaine, dans la mesure où nous nous intéressons au fonctionnement d'un réseau de capteurs, et elle peut-être reprise pour différentes configurations, car seules les instances de départ changeront (voir (Sadoun *et al.*, 2012) pour une justification de cette conceptualisation). L'ensemble des individus sont identifiables à partir de leurs propriétés de localisation et de type.

7.2 Description des ressources

En l'absence de corpus suffisamment grand portant sur la spécification d'exigences dans le domaine des environnements intelligents, nous avons constitué un corpus d'apprentissage d'environ 5 millions de mots à partir de livres électroniques (e-books) de domaines et styles littéraires différents, issus de la *Bibliothèque numérique Anacleto*[7]. La diversité de ce corpus permet d'acquérir des règles pour des styles d'écriture différents. Par ailleurs, les propriétés recherchées sont suffisamment générales et transdomaines pour qu'on puisse partir d'un corpus non spécialisé pour acquérir les règles d'extraction. En contrepartie, sa généralité limite l'extraction de la terminologie pour les concepts spécifiques au domaine. Afin de disposer d'un corpus d'évaluation, nous avons développé une plate-forme[8] de collecte de spécifications. Les spécifications collectées représentent environ 80 phrases (1558 mots).

7.3 Résultats

7.3.1 Acquisition des règles d'extraction

Nous avons acquis en tout 126 règles d'extraction, dont 31 pour l'identification d'instances de propriétés complètes et 95 pour l'identification d'instances de propriétés partielles. Chaque règle a été générée à partir des n chemins syntaxiques les plus fréquents. Nous avons fixé ce paramètre de façon expérimentale à 4. Néanmoins, les mentions de certaines propriétés sont moins fréquentes dans les textes et ce nombre peut s'avérer trop bas. En l'augmentant, des chemins non pertinents peuvent engendrer des règles. Afin d'éviter ce bruit éventuel, nous fixons une seconde limite correspondant au nombre de dépendances syntaxiques composant le chemin. En effet, plus un chemin est long, et donc plus les termes sont distants dans la phrase, moins il y a de possibilité que ce chemin représente une propriété sémantique. Dans nos expériences, le nombre de dépendances maximal d'un chemin a été fixé à 4.

7.3.2 Évaluation de l'acquisition de termes

La terminologie amorce contient 109 termes, 18 termes préférés et 91 alternatifs. Ces termes sont associés aux instances de concepts et propriétés modélisés dans l'ontologie. Certains termes sont la dénomination d'individus préexistant dans l'ontologie, par exemple les localisations et les différents types reconnus par les capteurs.

Le tableau 1 illustre l'acquisition des termes sur trois catégories sémantiques par expansion de la terminologie. Ces catégories sémantiques représentent dans l'ordre les phénomènes à reconnaître, les processus physiques qui correspondent aux appareils connectés aux réseaux de capteurs et les actions possibles[9]. Les règles d'ex-

7. (http ://www.gutenberg.us/)

8. http ://perso.limsi.fr/sadoun/Application/fr/SmartHome.php

9. Actuate-on a comme sous-propriété : Turn-on, Turn-off, Increase et Decrease

traction ont été appliquées sur le corpus d'apprentissage comme cela est décrit en section 5. Le première colonne exprime les termes amorces issus de la terminologie de départ. La seconde colonne le nombre de termes différents extraits et la troisième colonne les termes pertinents.

	Amorce	Extrait	Pertinent
Phenomenon	18	965	49
Physical-process	33	625	23
Actuate-on	19	413	27

TABLE 1: Termes extraits pertinents

En raison de la généralité du corpus et la spécificité du domaine, la précision de l'extraction est assez basse. La sélection des termes pertinents est réalisée manuellement à partir de l'examen des n premiers termes retournés par la formule du calcul de pertinence cf. section 5, avec n fixé à 25%.

7.3.3 Évaluation de l'extraction des triplets candidats

L'extraction des triplets résulte de l'application sur le corpus de test des règles acquises. L'application des règles complètes est prioritaire par rapport aux règles partielles. Cette extraction est effectuée en ne considérant que les termes pertinents dans la terminologie.

Les résultats de l'extraction de triplets candidats sont décrits dans le tableau 2. La première colonne indique le nombre d'instances à reconnaître. Les colonnes suivantes indiquent le nombre de triplets correctement identifiés, et les triplets identifiés à tort. La première ligne représente les résultats pour les trois propriétés *Located-in*, *Fixed-in* et *Occured-in* qui associent une localisation à chacun des concepts *Localisation*, de *Physical-process* et *Phenomenon*. La propriété *Has-type* associe un *Type* aux phénomènes. Les deux dernières lignes représentent les instances de concepts *Phenomenon* et *Actuator* qui sont classées et identifiées lors du raisonnement sur leurs instances de propriétés.

Nous observons que la précision est très élevée (0.95). Cela montre la pertinence des règles acquises. De plus le rappel obtenu (0.63) est relativement élevé compte tenu du fait que l'acquisition des règles d'extraction et de la terminologie s'est faite à partir d'un corpus non spécifique au domaine. À partir des instances de propriété créées dans l'ontologie, 22 instances de *Phenomenon* et

	Pertinent	Correct	Incorrect	P	R	F-M
Loc	115	75	9	0.89	0.65	0.75
Has-type	62	35	0	1	0.56	0.72
Actuate-on	90	51	0	1	0.56	0.71
Total	267	164	9	0.95	0.61	0.7
Phenomenon	62	22	0	1	0.35	0.51
Actuator	42	17	0	1	0.40	0.57

TABLE 2: Extraction de triplets candidats

17 instances de *Actuator* ont été classées correctement, aucune n'a été incorrectement classée. Les instances de *Phenomenon* ont été identifiées comme appartenant à 10 individus différents. Les instances d'*Actuator* ont été identifiées comme appartenant à 7 individus différents.

7.4 Application

Les instances de propriété sont créées à partir des triplets extraits des textes. Lors de leur création, les termes représentant les domaine et image de la propriété sont nommés de la manière suivante : Si la formulation préférée d'un terme dénote un individu de l'ontologie alors il prend le nom de l'individu. sinon son nom est composé à partir du numéro de la phrase dans laquelle il apparaît et de son numéro de nœud dans l'arbre de dépendances syntaxiques. Par exemple, si le terme apparaît dans la phrase numéro 2 et son numéro de nœud dans l'arbre syntaxique est 3 alors il sera nommé 2-3. Cela permet de nommer les instances de façon unique et de mettre au même niveau les instances de concept explicites ou implicites dans les textes, issues des règles complètes ou partielles. Ainsi l'instance 2-3 qui apparaît dans les propriétés *Occured-in(2-3 , kitchen)* et *Has-type(2-3 , movement)* est déduite comme appartenant au concept *Phenomenon* de deux façons : à partir du domaine des propriétés *Occured-in* et *Has-type* définies sur le concept *Phenomenon* (cf. 6.1.1) et à partir de l'axiome d'équivalence : *Phenomenon* $\equiv$ *Has-type.Type* $\wedge$ *Occured-in.Location* qui définit une contrainte nécessaire et suffisante (cf. 6.1.2) pour reconnaître une instance du concept *Phenomenon*. L'instance est ensuite identifiée par rapport aux autres instances du concept *Phenomenon* à partir de la règle SWRL suivante :
Occured-in $(i_{P_1}\ ,\ i_L) \wedge$ *Occured-in*$(i_{P_2}\ ,\ i_L) \wedge$ *Has-type*$(i_{P_1}\ ,\ i_T) \wedge$ *Has-type*$(i_{P_2}\ ,\ i_T) \rightarrow$ *Sa-*

meAs(i_{P_1}, i_{P_2})

Cette règle exprime la contrainte d'unicité (cf. 6.2.2) inhérente aux individus du concept *Phenomenon*. Lors du raisonnement elle s'exprime concrètement de la manière suivante :
Occured-in(2-3 , kitchen) $\wedge$ *Occured-in* (i_2 , kitchen) $\wedge$ *Has-type*(2-3 , movement) $\wedge$ *Has-type*(i_2 , movement) $\rightarrow$ *SameAs*(2-3,i_2)

8 Conclusion

Nous avons présenté une approche de peuplement d'ontologie visant à représenter de manière formelle des connaissances issues de spécifications d'exigences. Cette approche est centrée sur l'identification de mentions d'instances de propriété dans les textes. Cette identification est faite à l'aide de règles d'extraction acquises à partir des chemins syntaxiques récurrents entre les termes dénotant les instances de concept et de propriété. Elles exploitent des connaissances lexicales, syntaxiques et sémantiques. Ces règles permettent d'identifier des instances de propriétés même lorsque l'une des instances du domaine ou de l'image est implicite. De plus, ces règles permettent de lever l'ambiguïté des termes extraits en capturant le contexte sémantique dans lequel ils apparaissent. Lors du raisonnement au sein de l'ontologie, les propriétés permettent de classer les instances de concepts et de les identifier de manière unique grâce aux contraintes d'unicité modélisées sous OWL. L'approche proposée a été conçue pour être indépendante du domaine et s'adapter facilement à d'autres langues. A partir d'une ontologie modélisée, elle ne nécessite qu'un corpus d'apprentissage ainsi qu'un ensemble de termes de départ. De plus seul le parseur utilisé et l'ensemble de termes de départ sont dépendants de la langue des textes à analyser.

References

BAJWA, I. S., LEE, M. et BORDBAR, B. (2012). Resolving syntactic ambiguities in natural language specification of constraints. *In Proceedings of the 13th international conference on Computational Linguistics and Intelligent Text Processing - Volume Part I*.

FOUGÈRES, A.-J. et TRIGANO, P. (1997). Rédaction de spécifications formelles : Èlaboration à partir des spécifications écrites en langage naturel. *In Cognito-Cahiers Romans de Sciences Cognitives*, 1(8):29–36.

GUISSÉ, A., LÉVY, F. et NAZARENKO, A. From regulatory texts to brms : how to guide the acquisition of business rules ? *In Proceedings of the 6th international conference on Rules on the Web : research and applications*.

HASEGAWA, T., SEKINE, S. et GRISHMAN, R. (2004). Discovering relations among named entities from large corpora. *In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*.

IJNTEMA, W., SANGERS, J., HOGENBOOM, F. et FRASINCAR, F. (2012). A lexico-semantic pattern language for learning ontology instances from text. *Web Semantics : Science, Services and Agents on the World Wide Web*, 15:37–50.

ILIC, D. (2007). Deriving formal specifications from informal requirements. *In Proceedings of the 31st Annual International Computer Software and Applications Conference - Volume 01*, COMPSAC '07, pages 145–152. IEEE Computer Society.

ILIEVA, M. et BOLEY, H. (2008). Representing textual requirements as graphical natural language for uml diagram generation. *In SEKE'08*, pages 478–483.

KOF, L. (2010). Requirements analysis : concept extraction and translation of textual specifications to executable models. *In Proceedings of the 14th international conference on Applications of Natural Language to Information Systems*, NLDB'09.

LIN, D. et PANTEL, P. (2001). Discovery of inference rules for question answering. *Natural Language Engineering*.

MAKKI, J., ALQUIER, A.-M. et PRINCE, V. (2008). Ontology population via nlp techniques in risk management. *In ICSWE*.

MICH, L., FRANCH, M. et INVERARDI, P. (2004). Market research for requirements analysis using linguistic tools. *Requirement Engineering*, 9(1):40–56.

NAKAMURA-DELLOYE, Y. et STERN, R. (2011). Extraction de relations et de patrons de relations entre entités nommées en vue de l'enrichissement d'une ontologie. *In TOTh*.

PETASIS, G., KARKALETSIS, V., PALIOURAS, G., KRITHARA, A. et ZAVITSANOS, E. (2011). Ontology population and enrichment : State of the art. *In Knowledge-Driven Multimedia Information Extraction and Ontology Evolution'11*.

SADOUN, D., DUBOIS, C., GHAMRI-DOUDANE, Y. et GRAU, B. (2012). Formalisation en OWL pour vérifier les spécifications d'un environnement intelligent. *In Actes de la conférence RFIA*.

THONGKRAU, T. et LALITROJWONG, P. (2012). Ontopop : An ontology population system for the semantic web. *IEICE Transactions*, 95-D(4):921–931.