ONTOLOGY LEARNING BY ANALYZING XML DOCUMENT STRUCTURE AND CONTENT

Nathalie Aussenac-Gilles, Mouna Kamel

IRIT, Université Paul Sabatier de Toulouse, 118, route de narbonne, 31062 Toulouse Cedex9, France
aussenac @irit.fr, kamel@irit.fr

Keywords: Natural language processing for ontology learning, extraction of semantic relations, information extraction from structured documents.

Abstract: Most existing methods for ontology learning from textual documents rely on natural language analysis. We extend these approaches by taking into account the document structure which bears additional knowledge. The documents that we deal with are XML specifications of databases. In addition to classical linguistic clues, the structural organization of such documents also contributes to convey meaning. In a first stage, we characterize the semantics of XML mark-up and of their relations. Then parsing rules are defined to exploit the XML structure of documents and to create ontology concepts and semantic relations. These rules make it possible to automatically learn a kernel of ontology from documents. In a second stage; this ontology is enriched with the results of text analysis by lexico-syntactic patterns. Both ontology learning rules and patterns are implemented in the Gate platform.

INTRODUCTION

Ontology learning from text has been investigated from around 2000, with early works like the Terminae (Aussenac-Gilles, Despres and Szulman, 2008) and the Text-to-Onto methods and tools, and several reference books like (Buitelaar, Cimiano and Magnini, 2005). These methods define how to select and combine relevant natural language processing (NLP) tools to find out linguistic clues for ontology items, or, better, to learn and enrich automatically an ontology. High level tasks, like term or relation extraction (Bourigault, 2002), combine several basic text processing. Relation extraction plays a major role to structure the ontology with hierarchical and other kinds of semantic relations, to assign properties to concepts and also to identify concepts. Relation extraction techniques (Grefenstette, 1994) include statistics (looking for repeated segments or meaningful predicate argument structures (Hindle, 1990)), robust or shallow linguistic analyses (mainly pattern matching on syntactically tagged corpora) (Giuliano, Lavelli and Romano, 2006) and learning (to learn new patterns from tagged corpora) (Nédellec and Nazarenko, 2003). A recent state-of-the-art on pattern-based relation extraction from text (Auger and Barriere, 2008) shows that a pattern may correspond to very different characterizations of how a semantic relation may be expressed in a given language and corpora. A pattern defines a way to explore a sequence of words, lemmas, POS, syntactical relations, or semantic classes. These patterns are often defined or checked by manual text browsing, although many linguists tend to use simple and efficient tools like concordancers (Daoust, 1996), KeyWordsInContext like SystemQuick (Ahmad and Holmes-Higgin, 1995), or basic text browsing functions in text editors.

A major assumption is that each pattern occurrence should appear within one sentence. But a text is much richer that a list of sentences (Charolles, 1997): its material presentation (Virbel and Luc, 2001), the sentence and paragraph sequencings (the discourse structure) (Asher, Busquet and Vieu, 2001), as well as the context surrounding the reader contribute to the interpretation process. Such features also contribute to relation identification and should be included in pattern definitions.

We propose here an approach which takes into account both the material structure of a document and its textual content. In fact, structural tags implemented in a document (section title, sub-section title, enumeration, etc.) express hierarchical relations on which we rely to elaborate a first ontology kernel. Furthermore, a text analysis allows enriching this ontology. We test our approach on database specification documents (in the scope of the GEONTO project) where the database structure is reflected by the document structure and constraints on the database content are expressed in natural language. A first evaluation of the tool that implements the method shows some strengths and limitations that draw directions for future works.

METHODOLOGY

We propose a method for ontology learning that combines two complementary document analyses: the first one bears on the document structure when it is described using languages such as HTML, SGML, XML taking advantage of the semantics of tags and their relations; the second one explores the document textual content by processing natural language. Each process is carried out independently thanks to a specific set of rules that lead to the definition of concepts and relations in an ontology.

Rules for parsing XML Document

The markup language provides a description of both the text structure and the relationships between the tagged textual units thanks to tree structure of the tags. In the case where tags mark textual units which are short phrases that correspond to linguistic formulations of concepts or relations, semantic relations can be defined thanks to specializations of the following prototypical rule:

When - *A and B are tags, B being covered by A*

- *C_1 and C_2 are concepts respectively labelled by the text marked by A and B*

Then *a semantic relation exists between C_1 and C_2.*

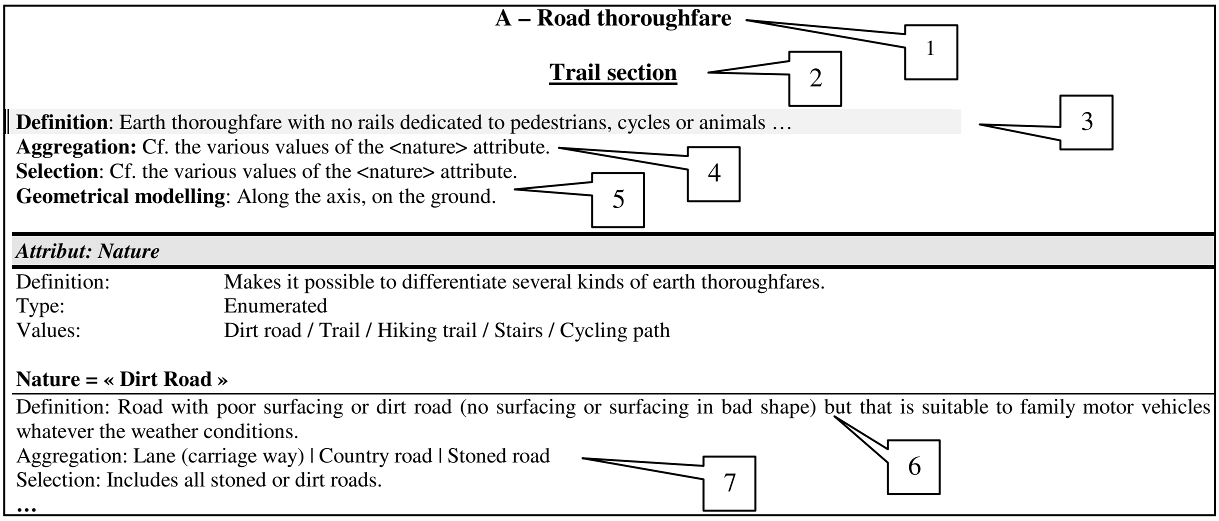
Specializing this rule requires human reading and interpretation of the tags and their relations to define a set of extraction rules. Indeed, the semantics conveyed by the tags in the tag tree depends on the context. But once these rules are written for a type of document compliant with an XML schema, they can automatically analyse any valid corpus compliant with this Schema, and provide a core ontology for each document of that type.

Rules for Natural Language Processing

The body of an XML document corresponds to natural language text and may contain relevant information for enriching the ontology obtained at the end of the previous step. According to Barrière and Agbado (2006), knowledge-rich contexts are text fragments that contain linguistic marks of semantic relation. We choose to use lexico-syntactic patterns to identify semantic relations in these text fragments. A lexico-syntactic pattern describes a regular expression, composed with words, syntactic or semantic categories, and typographic symbols to identify text fragments matching this format. These features are assigned by various NLP tools (tokenizer, parser, tagger, etc.). We defined a set of patterns for three basic semantic relations: hypernymy, meronymy, functional relations. Text analysis with these patterns leads to enrich the ontology kernel with new concepts and relations.

EXPERIMENTAL CONTEXT

Within the GEONTO project (http://geonto.lri.fr/), one of the partners owns heterogeneous geographical databases and aims at reaching interoperability among them. The GEONTO partners have planned an ontology-based solution: one ontology will be built up for each database and should reflect its content as much as possible; then these ontologies will be mapped to a unique reference ontology.



A – Road thoroughfare

Trail section

Definition: Earth thoroughfare with no rails dedicated to pedestrians, cycles or animals …
Aggregation: Cf. the various values of the <nature> attribute.
Selection: Cf. the various values of the <nature> attribute.
Geometrical modelling: Along the axis, on the ground.

Attribut: Nature

Definition:	Makes it possible to differentiate several kinds of earth thoroughfares.
Type:	Enumerated
Values:	Dirt road / Trail / Hiking trail / Stairs / Cycling path

Nature = « Dirt Road »

Definition: Road with poor surfacing or dirt road (no surfacing or surfacing in bad shape) but that is suitable to family motor vehicles whatever the weather conditions.
Aggregation: Lane (carriage way) | Country road | Stoned road
Selection: Includes all stoned or dirt roads.
…

Figure 1: Translated excerpt of specifications related to the "Trail Section" class

In most previous works on ontology learning from database schema (Grcar, Klein and Novak, 2007), the resulting ontologies have a low level of depth due to the flat structure of the relational schema. The only constraints which can be taken into account are those allowed by the definition language (Tirmizi, Sequeda and Miranker, 2008). As database specifications are richer than database schemas, we assume that GEONTO document analysis will provide richer ontologies.

For each database specifications is an XML document validated by the same XML Schema. This XML schema is compliant with the INSPIRE1 standard which is a directive of the European Parliament and of the Council (2007) establishing an Infrastructure for Spatial Information in the European Community.

3.1 GEONTO Document Features

The experiment reported in this paper bears on the BDTopo database. A translated excerpt of the original French MSWord document is shown in Figure 1 (related to the "Trail section" (*Tronçon de chemin*) class) and the corresponding extract of the XML specification appears in Figure 2.

In the BDTopo specifications, object classes (2 in Figure 1) are distributed over 9 information areas (titles as (1) in Figure1). The objects of a particular class listed in the *Aggregation* feature (4) share a single definition (3), the same kind of geometry (5) and the same list of attributes. In turn, each attribute value has its own definition (6) and can list object names (7).

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<featureCatalogue> <package>
 <packageName>A - Road Thoroughfare
 </packageName>
 <class>
  <className>Trail section</className>
   <description type="definition">Earth
              Thoroughfare … </description>
  …
   <description    type="extensionalDefinition">
Cf.
       the various values of the <nature>
       attribute </description>
  …
   <attributes>
    <attribute>
      <attributeName> Nature </attributeName>
      <descrition type="definition"> Makes it
         possible to differentiate several kinds
         of … </description>
      <valueType> Énumerated</valueType>
      <enumeratedValues >
        <value name="Dirt Road">
         <valueName> Dirt road </valueName>
         <description type="definition">
            Road with poor surfacing …
         </description>
       <description type="extensionalDefinition"
>
           Lane (carriage way)|Track|
           Stoned road </description >
         </value>
         …
      </enumeratedValues>
         …
    </attribute>
   </attributes>
  </class></package> </featureCatalogue>
```

Figure 2: Excerpt of the XML document in Figure 1

XML Document Processing

Exploiting Tags and Tag Dependencies

A systematic study of the XML schema proved that tags and their dependencies could be interpreted with regularity as indices of domain concepts, semantic relations and concept properties. We report here below how to identify ontology elements.

Concepts: each of the terms inside tags like <*PackageName*>, <*className*>, <*attributeName*> (for qualitative attributes), <*valueName*> or <*TermList*>, lead to a concept definition with the corresponding term as label.

Hierarchical Relations: they are derived from dependencies between some specific tags (i.e. <*PackageName*> and <*className*>, <*className*> and <*TermList*>>, <*attributeName*> and <*valueName*>, <*valueName*> and <*TermList*>) and from the concepts identified thanks to these tags.

Properties: properties are identified thanks to the terms inside <*attributeName*> tags when these attributes are quantities, and they relate to the concepts identified thanks to the terms marked by <*className*> tags.

Non-hierarchical semantic relations: semantic relations are identified thanks to the terms marked by <*attributeName*> tags when these attributes are qualitative, and they relate to the concepts identified thanks to the terms marked by <*className*> tags.

The semantics of properties and non-hierarchical relations cannot be determined unless the text marked by <*valueType*> tags is analysed.

The identification of all correspondences between XML specifications and ontology elements did not raise any major difficulty as long as tag labels convey their own semantics and relations can be easily identified with some common-sense knowledge.

Text Analysis with Lexico-Syntactic Patterns

The specification document of the BDTopo database contains very short and very synthetic paragraphs, where expressions of conceptual relations are quite sparse. Nevertheless, the definition field of any section may contain sentences that express hierarchical relations, metonymy relations or even some property definitions (Rebeyrolle and Tanguy, 2000). We propose to exploit these short paragraphs to enrich the core ontology.

a) **Semantic relation identification**

Corpus analysis shows that the most recurrent lexical relation is the meronymic one. Meronymy is characterized by key words like *part of*, *portion of*, *section of*, etc.

To find the maximum number of ways to express this relation, we use dictionaries and WordNet synsets. We show below a part of the pattern (Rule 1) that matches all

[1] INSPIRE : http://inspire.jrc.ec.europa.eu/

these formulations (patterns are written according to the JAPE[2] syntax):

```
({Concept} ({Token.lemma== "portion"}|{Token.lemma== "part"}|...)
({Token.categoty== "PREP"}) ({Token.category==”DET:ART”})?
({Term}) :annot)
- -> annot.ANNOT1 = { rule="Rule1"}
```

Rule1 looks for an ontology concept (obtained by construction in 3.2.1) followed by a sequence of one of the *part*, *portion*, etc. lemmas, a preposition, eventually a determinant, then by a term annotated *Term* (recognized by a term extractor, here TermoStat[3]). If such a sequence is recognized, the term will be tagged as *ANNOT1*.

Then a parser will read those tags, define a concept C_2 from this term tagged as *ANNOT1*, and have to set a *part-of* relation between C_2 and the concept C_1 where this definition occurs. Two cases may arise:

- if C_2 already exists in the ontology, the relation is established between C_1 and C_2.

- If C_2 does not exist, we create it and apply one of these processes (Buitelaar, Olejnik and Sintek, 2004):

a) if C_3 is a concept of the ontology the label of which is included in that of C_2 (C_2 may be considered as more specific than C_3), C_2 becomes a child of C_3.

b) if C_3 is a concept of the ontology the label of which includes that of C_2 (C_3 may be considered as more specific than C_2), C_2 becomes a father of C_3.

c) otherwise C_2 becomes a child of the ontology Top concept.

Then the relation is stated between C_1 and C_2 .

For instance, take the following definition: "*Road section: part of thoroughfare dedicated to cars*". By construction, *Road section* is a concept of the ontology. Rule1 annotates *thoroughfare* as a term. A concept labelled *Thoroughfare* is created and linked to the *Top* concept as a child. The *part-of* relation is then established between the *Road section* and *Thoroughfare* concepts.

A further stage consists in making the concept *Thoroughfare* as the father concept of the concept *Earth thoroughfare* (which is yet under the concept *Top*) using the lexical inclusion principle.

b) **Property identification**

A property expression is characterized by an adjective or a noun adjunct when it is associated to a term. We give below the pattern that identify these cases:

```
({Concept} {Term}
(((({Token.category==”VER:pper”}{Token.category==”PREP”}
{Terme}) : annot1) | ({Token.category==”ADJ”}):annot2)
) - - > annot1.ANNOTA = {rule="Rule2"},
        annot2.ANNOTB = {rule="Rule2"},
```

When applied to the definition of the *Road section*, this pattern identifies a new property *dedicated to cars* which will associated to the *Road section* concept.

Implementation

This first step of the ontology learning process has been implemented within the GATE[4] NLP platform. A process in GATE is defined as a pipeline that runs more basic text processing tools. Each step adds new annotations in the input corpus, and the result is an annotated corpus. In a further stage, these annotations can be read or processed by JAPE Rules or Java programs. As long as GATE makes available an *Ontology API*, it is easy to build up an ontology by processing text annotations. Moreover, GATE considers that the XML tags of an input document are annotations. Hence, the GATE platform makes it possible to define a unified process that supports the exploitation of (1) structural tags to get a first ontology kernel, and (2) annotations resulting from processing resources to enrich the ontology.

Figure 5 presents an extract of the resulting concept hierarchy in the ontology. The *Trail section* (*Tronçon de chemin*) concept is a child of *Road thoroughfare (Voies de Communication Routière*) and has *Trail* (*Chemin*), *Dirt road* (*Chemin empierre*), *Stairs* (*Escalier*), *Cycling_path* (*Piste cyclable*) *and Dirt_road* (*Sentier*) as children concepts. The three children concepts of *Dirt road* are *Carriage lane* (*Allée carossable*), *Track* (*Piste*), *Dirt road* (*Route empierrée*).

Furthermore, rules (1) and patterns (2) state new concepts, relations and properties. Let’s consider the concept *Trail section*.

(1) it has an attribute (represented as DataTypeProperty) labelled *has-name* and is of type *String*. *Trail section* is related to the concept *passage* (*Franchissement*) by the semantic relation *has-Crossing* modelled as ObjectType Property.

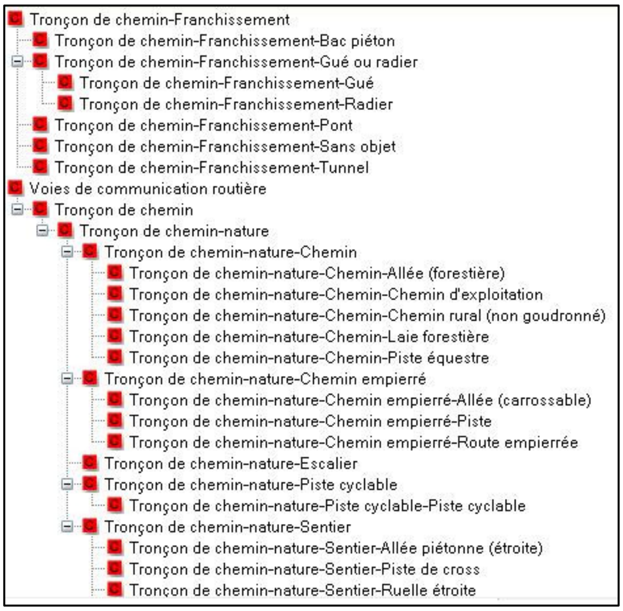
(2) The property *dedicated to pedestrians* is linked to the concept *Trail section*.

As a same term may appear in several parts of the text with different definitions or properties, we choose to differentiate these concepts by labelling a concept with its father concept labels and its own.

[2] JAPE : Java Annotation Pattern Engine

[3] TermoStat : developed at the Montreal University http://www.mapageweb.umontreal.ca/drouinp/index_en.html

[4] General Architecture for Text Engineering : Natural language processing platform developed at the Sheffield University (http://gate.ac.uk)



- Tronçon de chemin-Franchissement
  - Tronçon de chemin-Franchissement-Bac piéton
  - Tronçon de chemin-Franchissement-Gué ou radier
    - Tronçon de chemin-Franchissement-Gué
    - Tronçon de chemin-Franchissement-Radier
  - Tronçon de chemin-Franchissement-Pont
  - Tronçon de chemin-Franchissement-Sans objet
  - Tronçon de chemin-Franchissement-Tunnel
- Voies de communication routière
  - Tronçon de chemin
    - Tronçon de chemin-nature
      - Tronçon de chemin-nature-Chemin
        - Tronçon de chemin-nature-Chemin-Allée (forestière)
        - Tronçon de chemin-nature-Chemin-Chemin d'exploitation
        - Tronçon de chemin-nature-Chemin-Chemin rural (non goudronné)
        - Tronçon de chemin-nature-Chemin-Laie forestière
        - Tronçon de chemin-nature-Chemin-Piste équestre
      - Tronçon de chemin-nature-Chemin empierré
        - Tronçon de chemin-nature-Chemin empierré-Allée (carrossable)
        - Tronçon de chemin-nature-Chemin empierré-Piste
        - Tronçon de chemin-nature-Chemin empierré-Route empierrée
      - Tronçon de chemin-nature-Escalier
      - Tronçon de chemin-nature-Piste cyclable
        - Tronçon de chemin-nature-Piste cyclable-Piste cyclable
      - Tronçon de chemin-nature-Sentier
        - Tronçon de chemin-nature-Sentier-Allée piétonne (étroite)
        - Tronçon de chemin-nature-Sentier-Piste de cross
        - Tronçon de chemin-nature-Sentier-Ruelle étroite

Figure 5. Extract of the ontology resulting from processing the specifications given in Figure1.

EVALUATION

We have tried to estimate the gain brought by our method on this corpus compared with similar known approaches. Given the formulation of the specifications, any statistical method cannot provide significant results: very short natural language paragraphs with few redundancies, very few occurrences of each term, many terms used in list without meaningful linguistic context. For similar reasons, linguistic approaches are not efficient as long as there are quite few written paragraphs (most of the phrases marked by tags are terms that define concept labels or attributes).

A previous work had been carried out on the same specification document (BDTopo) and had led to a first ontology (Laurens, 2006). This work explores the text visual lay-out (paragraph style, character type and caption, frames, etc.) considered as bearing semantics. The overall process produces a taxonomy.

So we have compared the effectiveness of both methods by comparing the quality of the resulting ontologies.

Comparison of the two resulting ontologies

Onto_SV and Onto_ST, the two ontologies obtained respectively by Laurens using the visual structure and by us using our approach, result from the same specification document of the BDTopo database. In both cases, setting up the document analysis process requires a precise manual interpretation of the semantics of original tags and of the way they are entwined. Human interpretation was also required at several other stages of the Onto_SV development process: to select/validate geographical terms, to clean up the XML hierarchy before the automatic generation of the concept taxonomy, to reorganise and improve the OWL representation of this taxonomy. An opposite option has been selected to build the Onto_ST ontology: the ontologist is supposed to modify only the ontology once it is automatically generated, to correct inconsistencies due to errors in the specification. Table1 compares several features of both ontologies.

Table 1: Features of the two ontologies

	Onto_SV	Onto_ST
Number of concepts	615	1251
Depth	6	6
Hierarchical IS_A relations	yes	yes
Properties	No	yes
Meronymy relation	No	yes
Other semantic relations	No	yes
Learning process	Supervised	Unsupervised

Onto_ST is built up automatically, contains more concepts (because our method is able to differentiate concepts with the same label but with different properties) and more relations (non hierarchical relations are extracted) than Onto_SV. Onto_ST is not the best domain ontology regarding concept definitions, relations or its hierarchical structure, but it is the closer one to the domain knowledge as expressed in the specification document.

Limitations and Advantages of our Approach

The quality of the resulting ontology depends entirely on the quality of the specification document: when inconsistencies appear in the specification file, human interpretation is required to correct their consequences in the ontology. This is one of the advantages of formalization: it helps localize any fuzzy information or inconsistency within highly structured documents like these specifications. Whatever the effort made by their authors, meaning variations (whether lexical, syntactical or related to the text material presentation) are one of the features of natural language in text. While processing the document, several such cases occurred: either the semantics of the relation was not the expected one, or one of the items of an enumeration had a different status from others, etc. A detailed study is given in (Kamel and Aussenac, 2009).

CONCLUSION AND FUTURE WORK

We have shown that, in the very positive context where texts are structured with well-defined tags with a clear semantics, it is possible to define a text processing chain that results efficient for the automatic construction of an ontology. This chain, implemented with the GATE platform, includes rules that exploit together several features of the document: its explicit structure through available tags and its content in natural language. The ontology obtained with this automatic process results rich in concepts and relations, and each of its element is precisely connected to the text from which it originates. This method is applicable to all XML documents referring database specifications and validated by the INSPIRE standard.

We are aware that this ontology contains inconsistencies that should be manually corrected. In the scope of the GEONTO project, ontology manual cleaning is planned.

For the time being, we feel like enriching the ontology automatically built up, in particular thanks to a more systematic analysis of definitions (especially when they contain conjunctions or disjunctions) and the text material presentation (we have identified several kinds of typographic marks that were not considered yet).

REFERENCES

[1] Ahmad, K., Holmes-Higgin, P.R., 1995. SystemQuick : A unified approach to text and terminology. In *Terminology in Advanced Microcomputer Applications. Proceedings of the 3rd TermNet Symposium.*. 181-194. Vienna, Austria.

[2] Asher, N., Busquet, J., Vieu, L., 2001. La SDRT: une approche de la cohérence du discours dans la tradition de la sémantique dynamique. *Verbum 23, 73-101.*

[3] Auger, A., Barriere, C., 2008. Pattern based approaches to semantic relation extraction: a state-of-the-art. *Terminology,* John Benjamins*,* 14-1,1-19.

[4] Aussenac-Gilles, N., Despres, S., Szulman, S. 2008. The TERMINAE Method and Platform for Ontology Engineering from texts. Bridging the Gap between Text and Knowledge - *Selected Contributions to Ontology Learning and Population from Text.* P. Buitelaar, P. Cimiano (Eds.), IOS Press, p. 199-223.

[5] Barrière, C., Agbado, A. 2006. TerminoWeb: a software environment for term study in rich contexts. *International Conference on Terminology, Standardization and Technology Transfert (TSTT 2006),* Beijing (China), p. 103-113.

[6] Bourigault, D., 2002. UPERY: un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. *TALN 2002,* Nancy, 24-27 juin 2002

[7] Buitelaar, P., Olejnik, D., Sintek, M., 2004. A Protégé plug-in for ontology extraction from text based on linguistic analysis. *In Proceedings of the 1st European Semantic Web Symposium (ESWS)*, p. 31-44.

[8] Buitelaar, P., Cimiano, P., Magnini, B., 2005. Ontology Learning From Text: Methods, Evaluation and Applications. *IOS Press.*

[9] Charolles, M., 1997. L'encadrement du discours: Univers, Champs, Domaines et Espaces. *Cahier de Recherche Linguistique,* LANDISCO, URA-CNRS 1035, Univ. Nancy 2, n°6, 1-73.

[10] Daoust, F ., 1996. SATO (Système d'Analyse de Texte par Ordinateur). Version 4.0. *Manuel de référence, Service d'Analyse de Texte par Ordinateur (ATO).* Montréal : Université du Québec

[11] Giuliano, C., Lavelli, A., Romano, L., 2006. Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. *In Proc. EACL* 2006.

[12] Grcar, M., Klein, E., Novak, B., 2007. Using Term-Matching Algorithms for the Annotation of Geo-services. *Post-proceedings of the ECML-PKDD 2007* Workshops, Springer, Berlin – Heidelberg – New York.Boston, MA: Kluwer Academic Plubisher .

[13] Grefenstette G. (1994), Explorations in Automatic Thesaurus Discovery. Boston, MA: *Kluwer Academic Plubisher.*

[14] Hindle, D., 1990. Noun classification from predicate argument structures. *In proc. of the 28th Annual Meeting of the Association for Computational Linguistics (ACL'90),* Berkeley USA.

[15] Jacquemin, C., 1997. Présentation des travaux en analyse automatique pour la reconnaissance et l'acquisition terminologique. *In Séminaire du LIPN,* Université Paris 13, Villetaneuse.

[16] Kamel, M., Aussenac, N., 2009. Construction automatique d'ontologies à partir de spécifications de bases de données. *Ingénierie des Connaissances*, Hammamet Tunisie 2009.

[17] Laurens, F., 2006. Construction d'une Ontologie à partir de Textes en Langage Naturel. *Rapport de Stage Master 1 en linguistique-Informatique*, September 2006.

[18] Nédellec, C., Nazarenko, A., 2003. Ontology and Information Extraction. *in S. Staab & R. Studer (eds.) Handbook on Ontologies in Information Systems*, Springer.

[19] Rebeyrolle, J., Tanguy, L. 2000. Repérage automatique de structures linguistiques en corpus: le cas des énoncés définitoires. *Cahiers de Grammaire*, 25, 153-174

[20] Tirmizi, S., Sequeda, S., Miranker, J.F, 2008. Translating SQL Applications to the Semantic Web. *Dexa 2008*, Turin , Italie, 450-464.

[21] Virbel, J., Luc, C., 2001. Le modèle d'architecture textuelle: fondements et expérimentation. *Verbum,* Vol. XXIII, N. 1, p. 103-123.

ONTOLOGY LEARNING BY ANALYZING XML DOCUMENT STRUCTURE AND CONTENT

Nathalie Aussenac-Gilles, Mouna Kamel

IRIT, Université Paul Sabatier de Toulouse, 118, route de narbonne, 31062 Toulouse Cedex9, France
aussenac @irit.fr, kamel@irit.fr

Keywords: Natural language processing for ontology learning, extraction of semantic relations, information extraction from structured documents.

Abstract: Most existing methods for ontology learning from textual documents rely on natural language analysis. We extend these approaches by taking into account the document structure which bears additional knowledge. The documents that we deal with are XML specifications of databases. In addition to classical linguistic clues, the structural organization of such documents also contributes to convey meaning. In a first stage, we characterize the semantics of XML mark-up and of their relations. Then parsing rules are defined to exploit the XML structure of documents and to create ontology concepts and semantic relations. These rules make it possible to automatically learn a kernel of ontology from documents. In a second stage; this ontology is enriched with the results of text analysis by lexico-syntactic patterns. Both ontology learning rules and patterns are implemented in the Gate platform.

INTRODUCTION

Ontology learning from text has been investigated from around 2000, with early works like the Terminae (Aussenac-Gilles, Despres and Szulman, 2008) and the Text-to-Onto methods and tools, and several reference books like (Buitelaar, Cimiano and Magnini, 2005). These methods define how to select and combine relevant natural language processing (NLP) tools to find out linguistic clues for ontology items, or, better, to learn and enrich automatically an ontology. High level tasks, like term or relation extraction (Bourigault, 2002), combine several basic text processing. Relation extraction plays a major role to structure the ontology with hierarchical and other kinds of semantic relations, to assign properties to concepts and also to identify concepts. Relation extraction techniques (Grefenstette, 1994) include statistics (looking for repeated segments or meaningful predicate argument structures (Hindle, 1990)), robust or shallow linguistic analyses (mainly pattern matching on syntactically tagged corpora) (Giuliano, Lavelli and Romano, 2006) and learning (to learn new patterns from tagged corpora) (Nédellec and Nazarenko, 2003). A recent state-of-the-art on pattern-based relation extraction from text (Auger and Barriere, 2008) shows that a pattern may correspond to very different characterizations of how a semantic relation may be expressed in a given language and corpora. A pattern defines a way to explore a sequence of words, lemmas, POS, syntactical relations, or semantic classes. These patterns are often defined or checked by manual text browsing, although many linguists tend to use simple and efficient tools like concordancers (Daoust, 1996), KeyWordsInContext like SystemQuick (Ahmad and Holmes-Higgin, 1995), or basic text browsing functions in text editors.

A major assumption is that each pattern occurrence should appear within one sentence. But a text is much richer that a list of sentences (Charolles, 1997): its material presentation (Virbel and Luc, 2001), the sentence and paragraph sequencings (the discourse structure) (Asher, Busquet and Vieu, 2001), as well as the context surrounding the reader contribute to the interpretation process. Such features also contribute to relation identification and should be included in pattern definitions.

We propose here an approach which takes into account both the material structure of a document and its textual content. In fact, structural tags implemented in a document (section title, sub-section title, enumeration, etc.) express hierarchical relations on which we rely to elaborate a first ontology kernel. Furthermore, a text analysis allows enriching this ontology. We test our approach on database specification documents (in the scope of the GEONTO project) where the database structure is reflected by the document structure and constraints on the database content are expressed in natural language. A first evaluation of the tool that implements the method shows some strengths and limitations that draw directions for future works.

METHODOLOGY

We propose a method for ontology learning that combines two complementary document analyses: the first one bears on the document structure when it is described using languages such as HTML, SGML, XML taking advantage of the semantics of tags and their relations; the second one explores the document textual content by processing natural language. Each process is carried out independently thanks to a specific set of rules that lead to the definition of concepts and relations in an ontology.

Rules for parsing XML Document

The markup language provides a description of both the text structure and the relationships between the tagged textual units thanks to tree structure of the tags. In the case where tags mark textual units which are short phrases that correspond to linguistic formulations of concepts or relations, semantic relations can be defined thanks to specializations of the following prototypical rule:

When - *A and B are tags, B being covered by A*

- *C_1 and C_2 are concepts respectively labelled by the text marked by A and B*

Then *a semantic relation exists between C_1 and C_2.*

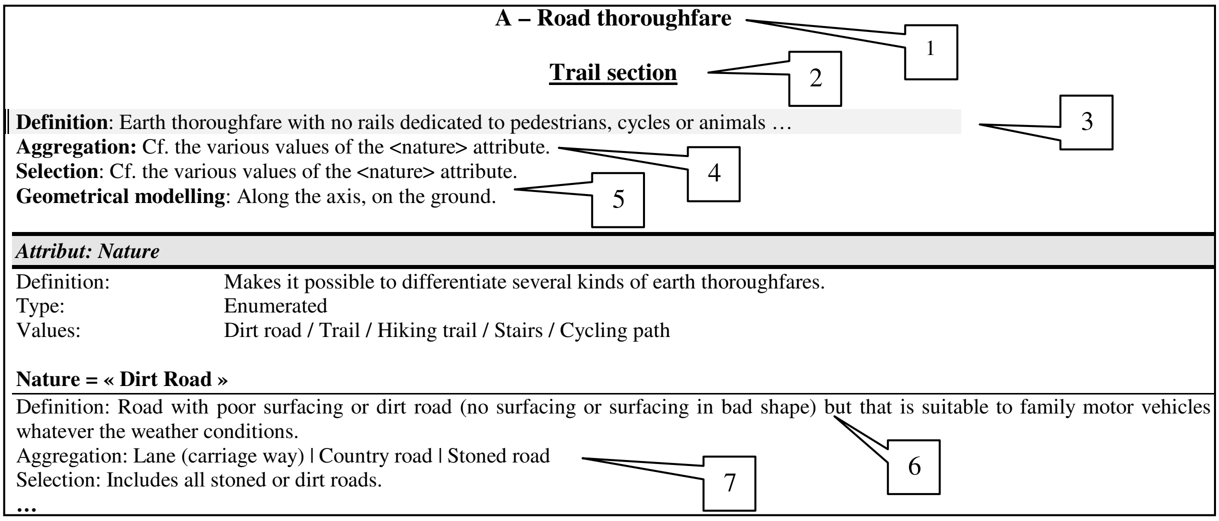
Specializing this rule requires human reading and interpretation of the tags and their relations to define a set of extraction rules. Indeed, the semantics conveyed by the tags in the tag tree depends on the context. But once these rules are written for a type of document compliant with an XML schema, they can automatically analyse any valid corpus compliant with this Schema, and provide a core ontology for each document of that type.

Rules for Natural Language Processing

The body of an XML document corresponds to natural language text and may contain relevant information for enriching the ontology obtained at the end of the previous step. According to Barrière and Agbado (2006), knowledge-rich contexts are text fragments that contain linguistic marks of semantic relation. We choose to use lexico-syntactic patterns to identify semantic relations in these text fragments. A lexico-syntactic pattern describes a regular expression, composed with words, syntactic or semantic categories, and typographic symbols to identify text fragments matching this format. These features are assigned by various NLP tools (tokenizer, parser, tagger, etc.). We defined a set of patterns for three basic semantic relations: hypernymy, meronymy, functional relations. Text analysis with these patterns leads to enrich the ontology kernel with new concepts and relations.

EXPERIMENTAL CONTEXT

Within the GEONTO project (http://geonto.lri.fr/), one of the partners owns heterogeneous geographical databases and aims at reaching interoperability among them. The GEONTO partners have planned an ontology-based solution: one ontology will be built up for each database and should reflect its content as much as possible; then these ontologies will be mapped to a unique reference ontology.



A – Road thoroughfare

Trail section

Definition: Earth thoroughfare with no rails dedicated to pedestrians, cycles or animals …
Aggregation: Cf. the various values of the <nature> attribute.
Selection: Cf. the various values of the <nature> attribute.
Geometrical modelling: Along the axis, on the ground.

Attribut: Nature

Definition: Makes it possible to differentiate several kinds of earth thoroughfares.
Type: Enumerated
Values: Dirt road / Trail / Hiking trail / Stairs / Cycling path

Nature = « Dirt Road »

Definition: Road with poor surfacing or dirt road (no surfacing or surfacing in bad shape) but that is suitable to family motor vehicles whatever the weather conditions.
Aggregation: Lane (carriage way) | Country road | Stoned road
Selection: Includes all stoned or dirt roads.
…

Figure 1: Translated excerpt of specifications related to the "Trail Section" class

In most previous works on ontology learning from database schema (Grcar, Klein and Novak, 2007), the resulting ontologies have a low level of depth due to the flat structure of the relational schema. The only constraints which can be taken into account are those allowed by the definition language (Tirmizi, Sequeda and Miranker, 2008). As database specifications are richer than database schemas, we assume that GEONTO document analysis will provide richer ontologies.

For each database specifications is an XML document validated by the same XML Schema. This XML schema is compliant with the INSPIRE1 standard which is a directive of the European Parliament and of the Council (2007) establishing an Infrastructure for Spatial Information in the European Community.

3.1 GEONTO Document Features

The experiment reported in this paper bears on the BDTopo database. A translated excerpt of the original French MSWord document is shown in Figure 1 (related to the "Trail section" (*Tronçon de chemin*) class) and the corresponding extract of the XML specification appears in Figure 2.

In the BDTopo specifications, object classes (2 in Figure 1) are distributed over 9 information areas (titles as (1) in Figure1). The objects of a particular class listed in the *Aggregation* feature (4) share a single definition (3), the same kind of geometry (5) and the same list of attributes. In turn, each attribute value has its own definition (6) and can list object names (7).

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<featureCatalogue> <package>
 <packageName>A - Road Thoroughfare
 </packageName>
 <class>
  <className>Trail section</className>
   <description type="definition">Earth
              Thoroughfare … </description>
  …
   <description    type="extensionalDefinition">
Cf.
       the various values of the <nature>
       attribute </description>
  …
   <attributes>
    <attribute>
      <attributeName> Nature </attributeName>
      <descrition type="definition"> Makes it
         possible to differentiate several kinds
         of … </description>
      <valueType> Énumerated</valueType>
      <enumeratedValues >
        <value name="Dirt Road">
         <valueName> Dirt road </valueName>
         <description type="definition">
            Road with poor surfacing …
         </description>
       <description type="extensionalDefinition"
>
           Lane (carriage way)|Track|
           Stoned road </description >
         </value>
         …
      </enumeratedValues>
         …
    </attribute>
   </attributes>
  </class></package> </featureCatalogue>
```

Figure 2: Excerpt of the XML document in Figure 1

XML Document Processing

Exploiting Tags and Tag Dependencies

A systematic study of the XML schema proved that tags and their dependencies could be interpreted with regularity as indices of domain concepts, semantic relations and concept properties. We report here below how to identify ontology elements.

<u>Concepts</u>: each of the terms inside tags like <*PackageName*>, <*className*>, <*attributeName*> (for qualitative attributes), <*valueName*> or <*TermList*>, lead to a concept definition with the corresponding term as label.

<u>Hierarchical Relations</u>: they are derived from dependencies between some specific tags (i.e. <*PackageName*> and <*className*>, <*className*> and <*TermList*>>, <*attributeName*> and <*valueName*>, <*valueName*> and <*TermList*>) and from the concepts identified thanks to these tags.

<u>Properties</u>: properties are identified thanks to the terms inside <*attributeName*> tags when these attributes are quantities, and they relate to the concepts identified thanks to the terms marked by <*className*> tags.

<u>Non-hierarchical semantic relations</u>: semantic relations are identified thanks to the terms marked by <*attributeName*> tags when these attributes are qualitative, and they relate to the concepts identified thanks to the terms marked by <*className*> tags.

The semantics of properties and non-hierarchical relations cannot be determined unless the text marked by <*valueType*> tags is analysed.

The identification of all correspondences between XML specifications and ontology elements did not raise any major difficulty as long as tag labels convey their own semantics and relations can be easily identified with some common-sense knowledge.

Text Analysis with Lexico-Syntactic Patterns

The specification document of the BDTopo database contains very short and very synthetic paragraphs, where expressions of conceptual relations are quite sparse. Nevertheless, the definition field of any section may contain sentences that express hierarchical relations, metonymy relations or even some property definitions (Rebeyrolle and Tanguy, 2000). We propose to exploit these short paragraphs to enrich the core ontology.

a) **<u>Semantic relation identification</u>**

Corpus analysis shows that the most recurrent lexical relation is the meronymic one. Meronymy is characterized by key words like *part of*, *portion of*, *section of*, etc.

To find the maximum number of ways to express this relation, we use dictionaries and WordNet synsets. We show below a part of the pattern (Rule 1) that matches all

[1] INSPIRE : http://inspire.jrc.ec.europa.eu/

these formulations (patterns are written according to the JAPE[2] syntax):

```
({Concept} ({Token.lemma== "portion"}|{Token.lemma== "part"}|...)
 ({Token.categoty== "PREP"}) ({Token.category==”DET:ART”})?
 ({Term}) :annot)
- -> annot.ANNOT1 = { rule="Rule1"}
```

Rule1 looks for an ontology concept (obtained by construction in 3.2.1) followed by a sequence of one of the *part*, *portion*, etc. lemmas, a preposition, eventually a determinant, then by a term annotated *Term* (recognized by a term extractor, here TermoStat[3]). If such a sequence is recognized, the term will be tagged as *ANNOT1*.

Then a parser will read those tags, define a concept C_2 from this term tagged as *ANNOT1*, and have to set a *part-of* relation between C_2 and the concept C_1 where this definition occurs. Two cases may arise:

- if C_2 already exists in the ontology, the relation is established between C_1 and C_2.

- If C_2 does not exist, we create it and apply one of these processes (Buitelaar, Olejnik and Sintek, 2004):

a) if C_3 is a concept of the ontology the label of which is included in that of C_2 (C_2 may be considered as more specific than C_3), C_2 becomes a child of C_3.

b) if C_3 is a concept of the ontology the label of which includes that of C_2 (C_3 may be considered as more specific than C_2), C_2 becomes a father of C_3.

c) otherwise C_2 becomes a child of the ontology Top concept.

Then the relation is stated between C_1 and C_2 .

For instance, take the following definition: "*Road section: part of thoroughfare dedicated to cars*". By construction, *Road section* is a concept of the ontology. Rule1 annotates *thoroughfare* as a term. A concept labelled *Thoroughfare* is created and linked to the *Top* concept as a child. The *part-of* relation is then established between the *Road section* and *Thoroughfare* concepts.

A further stage consists in making the concept *Thoroughfare* as the father concept of the concept *Earth thoroughfare* (which is yet under the concept *Top*) using the lexical inclusion principle.

b) **Property identification**

A property expression is characterized by an adjective or a noun adjunct when it is associated to a term. We give below the pattern that identify these cases:

```
 ({Concept} {Term}
 ((({Token.category==”VER:pper”}{Token.category==”PREP”}
{Terme}) : annot1) | ({Token.category==”ADJ”}):annot2)
) - - > annot1.ANNOTA = {rule="Rule2"},
        annot2.ANNOTB = {rule="Rule2"},
```

When applied to the definition of the *Road section*, this pattern identifies a new property *dedicated to cars* which will associated to the *Road section* concept.

Implementation

This first step of the ontology learning process has been implemented within the GATE[4] NLP platform. A process in GATE is defined as a pipeline that runs more basic text processing tools. Each step adds new annotations in the input corpus, and the result is an annotated corpus. In a further stage, these annotations can be read or processed by JAPE Rules or Java programs. As long as GATE makes available an *Ontology API*, it is easy to build up an ontology by processing text annotations. Moreover, GATE considers that the XML tags of an input document are annotations. Hence, the GATE platform makes it possible to define a unified process that supports the exploitation of (1) structural tags to get a first ontology kernel, and (2) annotations resulting from processing resources to enrich the ontology.

Figure 5 presents an extract of the resulting concept hierarchy in the ontology. The *Trail section* (*Tronçon de chemin*) concept is a child of *Road thoroughfare (Voies de Communication Routière*) and has *Trail* (*Chemin*), *Dirt road* (*Chemin empierre*), *Stairs* (*Escalier*), *Cycling_path* (*Piste cyclable*) *and Dirt_road* (*Sentier*) as children concepts. The three children concepts of *Dirt road* are *Carriage lane* (*Allée carossable*), *Track* (*Piste*), *Dirt road* (*Route empierrée*).

Furthermore, rules (1) and patterns (2) state new concepts, relations and properties. Let’s consider the concept *Trail section*.

(1) it has an attribute (represented as DataTypeProperty) labelled *has-name* and is of type *String*. *Trail section* is related to the concept *passage* (*Franchissement*) by the semantic relation *has-Crossing* modelled as ObjectType Property.

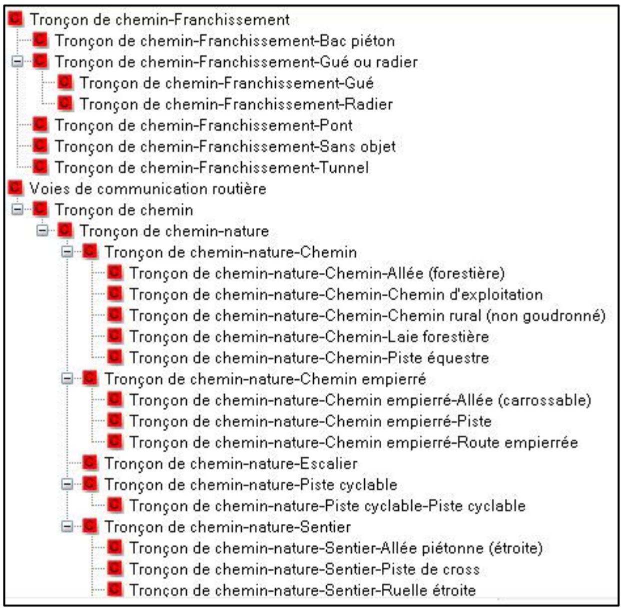
(2) The property *dedicated to pedestrians* is linked to the concept *Trail section*.

As a same term may appear in several parts of the text with different definitions or properties, we choose to differentiate these concepts by labelling a concept with its father concept labels and its own.

[2] JAPE : Java Annotation Pattern Engine

[3] TermoStat : developed at the Montreal University http://www.mapageweb.umontreal.ca/drouinp/index_en.html

[4] General Architecture for Text Engineering : Natural language processing platform developed at the Sheffield University (http://gate.ac.uk)



- Tronçon de chemin-Franchissement
  - Tronçon de chemin-Franchissement-Bac piéton
  - Tronçon de chemin-Franchissement-Gué ou radier
    - Tronçon de chemin-Franchissement-Gué
    - Tronçon de chemin-Franchissement-Radier
  - Tronçon de chemin-Franchissement-Pont
  - Tronçon de chemin-Franchissement-Sans objet
  - Tronçon de chemin-Franchissement-Tunnel
- Voies de communication routière
  - Tronçon de chemin
    - Tronçon de chemin-nature
      - Tronçon de chemin-nature-Chemin
        - Tronçon de chemin-nature-Chemin-Allée (forestière)
        - Tronçon de chemin-nature-Chemin-Chemin d'exploitation
        - Tronçon de chemin-nature-Chemin-Chemin rural (non goudronné)
        - Tronçon de chemin-nature-Chemin-Laie forestière
        - Tronçon de chemin-nature-Chemin-Piste équestre
      - Tronçon de chemin-nature-Chemin empierré
        - Tronçon de chemin-nature-Chemin empierré-Allée (carrossable)
        - Tronçon de chemin-nature-Chemin empierré-Piste
        - Tronçon de chemin-nature-Chemin empierré-Route empierrée
      - Tronçon de chemin-nature-Escalier
      - Tronçon de chemin-nature-Piste cyclable
        - Tronçon de chemin-nature-Piste cyclable-Piste cyclable
      - Tronçon de chemin-nature-Sentier
        - Tronçon de chemin-nature-Sentier-Allée piétonne (étroite)
        - Tronçon de chemin-nature-Sentier-Piste de cross
        - Tronçon de chemin-nature-Sentier-Ruelle étroite

Figure 5. Extract of the ontology resulting from processing the specifications given in Figure1.

EVALUATION

We have tried to estimate the gain brought by our method on this corpus compared with similar known approaches. Given the formulation of the specifications, any statistical method cannot provide significant results: very short natural language paragraphs with few redundancies, very few occurrences of each term, many terms used in list without meaningful linguistic context. For similar reasons, linguistic approaches are not efficient as long as there are quite few written paragraphs (most of the phrases marked by tags are terms that define concept labels or attributes).

A previous work had been carried out on the same specification document (BDTopo) and had led to a first ontology (Laurens, 2006). This work explores the text visual lay-out (paragraph style, character type and caption, frames, etc.) considered as bearing semantics. The overall process produces a taxonomy.

So we have compared the effectiveness of both methods by comparing the quality of the resulting ontologies.

Comparison of the two resulting ontologies

Onto_SV and Onto_ST, the two ontologies obtained respectively by Laurens using the visual structure and by us using our approach, result from the same specification document of the BDTopo database. In both cases, setting up the document analysis process requires a precise manual interpretation of the semantics of original tags and of the way they are entwined. Human interpretation was also required at several other stages of the Onto_SV development process: to select/validate geographical terms, to clean up the XML hierarchy before the automatic generation of the concept taxonomy, to reorganise and improve the OWL representation of this taxonomy. An opposite option has been selected to build the Onto_ST ontology: the ontologist is supposed to modify only the ontology once it is automatically generated, to correct inconsistencies due to errors in the specification. Table1 compares several features of both ontologies.

Table 1: Features of the two ontologies

	Onto_SV	Onto_ST
Number of concepts	615	1251
Depth	6	6
Hierarchical IS_A relations	yes	yes
Properties	No	yes
Meronymy relation	No	yes
Other semantic relations	No	yes
Learning process	Supervised	Unsupervised

Onto_ST is built up automatically, contains more concepts (because our method is able to differentiate concepts with the same label but with different properties) and more relations (non hierarchical relations are extracted) than Onto_SV. Onto_ST is not the best domain ontology regarding concept definitions, relations or its hierarchical structure, but it is the closer one to the domain knowledge as expressed in the specification document.

Limitations and Advantages of our Approach

The quality of the resulting ontology depends entirely on the quality of the specification document: when inconsistencies appear in the specification file, human interpretation is required to correct their consequences in the ontology. This is one of the advantages of formalization: it helps localize any fuzzy information or inconsistency within highly structured documents like these specifications. Whatever the effort made by their authors, meaning variations (whether lexical, syntactical or related to the text material presentation) are one of the features of natural language in text. While processing the document, several such cases occurred: either the semantics of the relation was not the expected one, or one of the items of an enumeration had a different status from others, etc. A detailed study is given in (Kamel and Aussenac, 2009).

CONCLUSION AND FUTURE WORK

We have shown that, in the very positive context where texts are structured with well-defined tags with a clear semantics, it is possible to define a text processing chain that results efficient for the automatic construction of an ontology. This chain, implemented with the GATE platform, includes rules that exploit together several features of the document: its explicit structure through available tags and its content in natural language. The ontology obtained with this automatic process results rich in concepts and relations, and each of its element is precisely connected to the text from which it originates. This method is applicable to all XML documents referring database specifications and validated by the INSPIRE standard.

We are aware that this ontology contains inconsistencies that should be manually corrected. In the scope of the GEONTO project, ontology manual cleaning is planned.

For the time being, we feel like enriching the ontology automatically built up, in particular thanks to a more systematic analysis of definitions (especially when they contain conjunctions or disjunctions) and the text material presentation (we have identified several kinds of typographic marks that were not considered yet).

REFERENCES

[1] Ahmad, K., Holmes-Higgin, P.R., 1995. SystemQuick : A unified approach to text and terminology. In *Terminology in Advanced Microcomputer Applications. Proceedings of the 3rd TermNet Symposium.*. 181-194. Vienna, Austria.

[2] Asher, N., Busquet, J., Vieu, L., 2001. La SDRT: une approche de la cohérence du discours dans la tradition de la sémantique dynamique. *Verbum 23, 73-101.*

[3] Auger, A., Barriere, C., 2008. Pattern based approaches to semantic relation extraction: a state-of-the-art. *Terminology,* John Benjamins*,* 14-1,1-19.

[4] Aussenac-Gilles, N., Despres, S., Szulman, S. 2008. The TERMINAE Method and Platform for Ontology Engineering from texts. Bridging the Gap between Text and Knowledge - *Selected Contributions to Ontology Learning and Population from Text.* P. Buitelaar, P. Cimiano (Eds.), IOS Press, p. 199-223.

[5] Barrière, C., Agbado, A. 2006. TerminoWeb: a software environment for term study in rich contexts. *International Conference on Terminology, Standardization and Technology Transfert (TSTT 2006),* Beijing (China), p. 103-113.

[6] Bourigault, D., 2002. UPERY: un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. *TALN 2002,* Nancy, 24-27 juin 2002

[7] Buitelaar, P., Olejnik, D., Sintek, M., 2004. A Protégé plug-in for ontology extraction from text based on linguistic analysis. *In Proceedings of the 1st European Semantic Web Symposium (ESWS)*, p. 31-44.

[8] Buitelaar, P., Cimiano, P., Magnini, B., 2005. Ontology Learning From Text: Methods, Evaluation and Applications. *IOS Press.*

[9] Charolles, M., 1997. L'encadrement du discours: Univers, Champs, Domaines et Espaces. *Cahier de Recherche Linguistique,* LANDISCO, URA-CNRS 1035, Univ. Nancy 2, n°6, 1-73.

[10] Daoust, F ., 1996. SATO (Système d'Analyse de Texte par Ordinateur). Version 4.0. *Manuel de référence, Service d'Analyse de Texte par Ordinateur (ATO).* Montréal : Université du Québec

[11] Giuliano, C., Lavelli, A., Romano, L., 2006. Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. *In Proc. EACL* 2006.

[12] Grcar, M., Klein, E., Novak, B., 2007. Using Term-Matching Algorithms for the Annotation of Geo-services. *Post-proceedings of the ECML-PKDD 2007* Workshops, Springer, Berlin – Heidelberg – New York.Boston, MA: Kluwer Academic Plubisher .

[13] Grefenstette G. (1994), Explorations in Automatic Thesaurus Discovery. Boston, MA: *Kluwer Academic Plubisher.*

[14] Hindle, D., 1990. Noun classification from predicate argument structures. *In proc. of the 28th Annual Meeting of the Association for Computational Linguistics (ACL'90),* Berkeley USA.

[15] Jacquemin, C., 1997. Présentation des travaux en analyse automatique pour la reconnaissance et l'acquisition terminologique. *In Séminaire du LIPN,* Université Paris 13, Villetaneuse.

[16] Kamel, M., Aussenac, N., 2009. Construction automatique d'ontologies à partir de spécifications de bases de données. *Ingénierie des Connaissances*, Hammamet Tunisie 2009.

[17] Laurens, F., 2006. Construction d'une Ontologie à partir de Textes en Langage Naturel. *Rapport de Stage Master 1 en linguistique-Informatique*, September 2006.

[18] Nédellec, C., Nazarenko, A., 2003. Ontology and Information Extraction. *in S. Staab & R. Studer (eds.) Handbook on Ontologies in Information Systems*, Springer.

[19] Rebeyrolle, J., Tanguy, L. 2000. Repérage automatique de structures linguistiques en corpus: le cas des énoncés définitoires. *Cahiers de Grammaire*, 25, 153-174

[20] Tirmizi, S., Sequeda, S., Miranker, J.F, 2008. Translating SQL Applications to the Semantic Web. *Dexa 2008*, Turin , Italie, 450-464.

[21] Virbel, J., Luc, C., 2001. Le modèle d'architecture textuelle: fondements et expérimentation. *Verbum,* Vol. XXIII, N. 1, p. 103-123.