

# How can document structure improve ontology learning?

Mouna Kamel  
IRIT

Université Paul Sabatier de Toulouse  
Toulouse, France  
kamel@irit.fr

Nathalie Aussenac-Gilles  
IRIT

Université Paul Sabatier de Toulouse  
Toulouse, France  
aussenac@irit.fr

## ABSTRACT

Most existing methods for ontology learning from textual documents rely on the natural language analysis of the text itself. We extend these approaches by taking into account the document structure which bears additional knowledge. The documents that we deal with are database specifications. Not only do they convey classical linguistic clues but the structural organization of such documents also contributes to their semantics. Our method is a two steps process to learn ontologies from text. The first step consists in applying structural patterns to automatically create a kernel of ontology. The second step aims at enriching this ontology with the results of text analysis with lexico-syntactic patterns. Ontology learning rules and patterns are implemented in the Gate platform.

## Categories and Subject Descriptors

I.2.7 : Natural Language Processing –*Text Analysis*.

## General Terms

Algorithms, Design, Experimentation

## Keywords

Semantic relation extraction, ontology learning from texts, ontology learning from document structure.

## 1. INTRODUCTION

Ontology learning from text has been investigated since around 2000, with early works like the *Terminae* (Aussenac-Gilles, Despres and Szulman, 2008) and the *Text-to-Onto* methods and tools, or several reference books like (Buitelaar, Cimiano and Magnini, 2005 ; Maedche, 2002). These methods define how to select and combine relevant natural language processing (NLP) tools to find out linguistic clues for ontology items, or to automatically learn and enrich an ontology. High level tasks, like term or relation extraction (Bourigault, 2002), combine several basic text processing steps like tokenization, lemmatization, POS tagging, syntactic dependency analyses, etc. (Jacquemin, 1997). Relation extraction plays a major role to structure the ontology with hierarchical and other semantic relations, to assign properties to concepts and also to identify concepts. Relation extraction techniques (Grefenstette, 1994) include statistical methods (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; Hearst, 1992), learning (to learn new patterns from tagged corpora) (Nédellec and Nazarenko, 2003) or text mining techniques (Grcar, Klein and Novak, 2007).

A recent survey on pattern-based relation extraction from text shows that many tools implement variants of this approach (Auger and Barriere, 2008). Tuning efficient patterns is a complex task that identifies few but precise relations. Patterns characterize how a semantic relation may be expressed in a given language and corpora. Nevertheless a major assumption is that each pattern occurrence should appear within one sentence. A text is much richer than a list of sentences (Charolles, 1997). When producing a document, a writer may use his linguistic skills but also his ability to structure logically and physically his text. In this regard, (Virbel and Luc, 2001) consider that the materiality of a text is part of its meaning. Finally, background knowledge of the reader takes a large part in his interpretation process. For all these reasons, we assume that document structural and material features also contribute to knowledge identification and should be taken into account in the ontology learning process.

Apart from a methodology which exploits the layout of a text for building a taxonomy (Laurens, 2006) (Abadie and Mustière, 2008), we do not know any ontology learning approach that takes advantage of document layout or structure. In this article we illustrate how exploiting not only the content of a document but also its logical structure and its layout can improve ontology learning from text.

This paper is organized as follows. Section 2 recalls how the structure of a document contributes to its semantics. Section 3 describes how we combine two approaches to exploit both the document structure and its natural language content. Section 4 presents an application of this method within the GEONTO project, in which ontologies are built from geographical database specification documents. Section 5 presents an implementation of our approach and gives some evaluation clues. We conclude and list several future work required to extend this approach.

## 2. DOCUMENT STRUCTURE AND SEMANTICS

### 2.1 Role of a document structure

Studies like (Lemarié et al, 2008) have shown that a document material realization, and particularly its structure, is one of the features of a text that influences its interpretation by the reader.

The structure plays a role as important as the reader's intension, his back-ground knowledge and the written text itself. Given a document structure, several interpretations are possible. One of the parameters that guide this interpretation is the type of document in hand. Other guidelines are specific structural rules that give a specific meaning to some structural properties. Each reader has in mind some of these rules that connect semantic features and structure. Nevertheless, such correspondence rules are not universal (for instance, the relation between a title and subtitle will differ in a newspaper and in a scientific article). According to the kind of documents, making these rules explicit requires a careful analysis of both the domain and the document.

## 2.2 Structural elements

A structural element reflects hierarchical links between segments of text. Among the various existing structural elements, the most commonly used are titles and sub-titles, enumerations and definitions.

Many studies show that enumerations, titles/sub-titles and definitions are knowledge rich contexts where relations between domain concepts are more or less explicitly expressed. As such, they may be used to identify ontological relations. For example, an enumerative structure consists in a preliminary term and a set of items (Luc, 2001). When these items share the same discourse structure, a semantic relation can be established between the preliminary term and each of the items. Furthermore, nested or parallel sub-section titles of a given section denote subordination or juxtaposition relations between these sections or their labels (Jacques, 2005). Again, when sub-titles of a given title have the same discourse structure, a semantic relation can be established between the title and each sub-title or between sub-titles. As for definitions, they are privileged places where semantic relations such as hierarchical relations, metonymy relations or even some property definitions are expressed (Rebeyrolle and Tanguy, 2000).

## 2.3 Representation of Structural Elements

A reader may perceive a document organization thanks to visual clues like captions, titles, paragraph indentation, enumerations etc. Nevertheless, an automatic identification of a document structure only based on lay-out features is restricted and misleading. In the scope of an automatic interpretation of structural elements, the document structure has to be made explicit. SGML languages have been designed to represent such features with tags that mark text fragments. With the notion of XML schema, a collection of documents can refer to a similar set of structural tags and share a similar structure. Large sets of textual documents are now available in XML format, which makes it possible to process automatically not only the natural language they contain, but also their structure for a more precise semantic interpretation.

In this paper, we propose to process the structure and the natural language of documents for ontology learning.

## 3. DOCUMENT ANALYSIS FOR ONTOLOGY LEARNING

We assume that combining the processing of a document structure and its linguistic content will help to build richer ontologies than just natural language processing. To start with, we include the three most commonly used structural elements (enumerations, titles and definitions) in an ontology learning process from text.

This approach requires that documents are in a SGML format type, so that the hierarchical structure of documents is explicit and allows an automatic processing. A systematic study of XML document shows that tags and their dependencies can be interpreted as indices of concepts, semantic relations and concept properties (Rebeyrolle and Perry-Woodley, 1998). Many tagged terms in titles are indices of domain concepts, and most of the relations between tags reveal hierarchical relations. In many favourable contexts where the document XML schema is semantically rich, the document structure provides pieces of a domain taxonomy. For this reason, we decided to process the document structure before processing in detail the natural language in these documents.

The idea is (1) to build a first ontology kernel by exploiting structural tags with the help of structural patterns, (2) to enrich this kernel by exploiting the text marked by tags with the help of lexico-syntactic patterns.

## 3.1 Structure Processing

The document structure analysis is carried out in three steps. First, XML documents are annotated with tags that make explicit the scope and attributes of each original XML tag. Then, instances of a structural generic pattern are defined according to the meaning of these tags. Then the document can be parsed with these patterns to get ontology fragments.

### 3.1.1 Additional Document Annotation

To make structure processing more efficient, we have developed a module that re-annotates the tags themselves. Each tag is renamed *Bal* with the semantic features *name* (tag name), *path* (tag localization) and *att<sub>1</sub>* (first tag attribute), *att<sub>2</sub>* (second tag attribute), and so on (the initial attributes of the tag are maintained). The *path* feature makes explicit the hierarchical position of each tag. The *att<sub>i</sub>* and *name* features are used for tag selection. In the example below, the XML document in fig. 1.a is translated into a new XML document (fig. 1.b):

```
<Book ISBN="9782212090819">
<Title> Hamlet </Title>
<Author> Shakespeare </Author>
<Details>
  <detail_name> Character </detail_name>
  <value_name> Hamlet </value_name>
  <value_name> Gertrude </value_name>
  <value_name> Claudius </value_name>
</Details>
</Book>
```

Figure 1.a: Input XML document

```
<Bal name="Book" path="/" ISBN="9782212090819">
<Bal name="Title" path="/Book"> Hamlet </Bal>
<Bal name="Author" path="/Book"> Shakespeare </Bal>
<Bal name="Details" path="/Book">
  <Bal name="detail_name" path="/Book/Details"> Character </Bal>
  <Bal name="value_name" path="/Book/Details"> Hamlet </Bal>
  ...
</Bal>
```

Figure 1.b: Re-annotated XML document

This new XML document is also enriched with additional annotations made by NLP tools: a tokenizer (*Token* tags), a term extractor (*Term* tags), a paragraph identifier (*Par* tags).

### 3.1.2 Structural Generic Patterns

We define two structural generic patterns (SGP1 and SGP2, fig. 2a and 2b) as a means of making explicit the semantics revealed by the combination of several structural tags. Each pattern tests the occurrence of a given sequence of terms and tags and concludes on the definition of an ontology fragment (new concepts, concept labels or semantic relation).

```
SGP1 (P1[, {(atti, vali)}], P2, P3[, {(attj, valj)}], P4)
({Bal.name=P1, Bal.path=P2, [Bal.atti= vali [, ...]], Term}):T1
({Token})*
({Bal.name=P3, Bal.path=P2{P4}, [Bal.attj= valj [, ...]], Term):T2
→ Oi (Rel (T1, T2))
```

**Figure 2a. Structural Generic Pattern for relation identification**

The pattern in fig. 2a applies when the term annotated T1 is followed by the term annotated T2 with 0 or several tokens in between. T1 (resp. T2) must be marked with a *Bal* tag named P1 (resp. P3) within path P2 (resp. P2{P4} which specifies that P4 is under the scope of P2), with a possible list of pairs <attribute, value> (att<sub>i</sub>, val<sub>i</sub>) (resp. (att<sub>j</sub>, val<sub>j</sub>)). If this sequence matches the XML document, Rel(T1, T2), a fragment of ontology, is defined: T1 and T2 will label concepts related by the Rel relation.

```
SGP2 (P1[, {(atti, vali)}], P2[, {(attj, valj)}], P3)
({Bal.name=P1, Bal.path=P3, [Bal.atti= vali [, ...]], Term}):T1
({Token})*
({Bal.name=P2, Bal.path=P3, [Bal.attj= valj [, ...]], Par): Par1
→ Oi (Def (T1, Par1))
```

**Figure 2b. Structural Generic Pattern for definition identification**

When the pattern in fig. 2b is matched, the definition paragraph Par1 of the concept labeled T1 is stored for further natural language processing.

### 3.1.3 Instances of Structural Patterns

Each XML schema has its own tags with a specific semantics. Only some of them mark term labels. Then the hierarchical structure and the document realization may reveal various semantic relations. For instance, structural elements like item lists neither have a conventional typography nor a stable tag name: enumerated items can be introduced by bullets, scores, separated by commas or not, they can be marked either as <item> or as <enum> etc. On the opposite, the same tag may have various meanings even within a single document.

Given an XML document (or schema), the SGPs are instantiated as many times as required, by fixing their parameter value according to the tags and their meaning in this schema. The instantiation produces document specific patterns. This work must be carried out by a person aware of the tag semantics.

Then these document specific patterns are matched on the XML document. The patterns for relation identification generate ontology fragments that are connected to get an ontology kernel. The definition patterns enrich concepts with natural language

definitions. During the next stage, this text is parsed by a pattern-based relation extractor.

## 3.2 Natural Language Processing

The body of the 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) and Hearst (1992), 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.

Some relations like hierarchical relations between classes (is-a) and meronymy (part-of) can be found in many domains. We defined several patterns adapted from Hearst (1992) and (Rebeyrolle and Tanguy, 2000), so that patterns may match on text where related terms appear in text fragments marked with different tags. For instance, a term and its definition could be marked with different tags.

With the objective of discovering domain specific relations, properties and attributes, specific patterns can be defined. For a full exploitation of the natural language paragraphs, a set of applicable patterns has to be adapted to each domain and corpus.

## 4. THE GEONTO APPLICATION

Within the GEONTO<sup>1</sup> project, the COGIT<sup>2</sup> lab (one of the project's partners) has built several heterogeneous geographical databases and aims at reaching interoperability among them. Each of these databases models geographic data according to a certain point of view. For instance, the BDTopo database is the reference one for localizing information related to urbanism, environment and territorial organization, while BDCarto refers to information on map data. To reach interoperability, a solution is to build one ontology per database which should reflect its content as much as possible, and then to map these ontologies.

To each database corresponds a specification document that describes the DB entities, their relationships, their definitions and possible values. It provides guidelines for feeding the databases with new data. These documents are an interesting input for the ontology learning process. Their rich semantics is expressed through their layout, their explicit XML structure and the natural language that they contain. Furthermore, all COGIT's specification documents are consistent with an XML schema according to the INSPIRE<sup>3</sup> standard. The semantics conveyed by this schema (tags and their hierarchical relations) will be explored to define an ontology.

Unlike previous works starting from databases to get an ontology, the ontology will be produced from specifications and not from the database schema (Bizer, 2003), (Tirmizi, Sequeda and Miranker, 2008), (Gardarin, Bedini and Nguyen, 2008).

Now let us describe the structure and the content of a specification document, and how the ontology is progressively built.

<sup>1</sup> GEONTO: <http://geonto.lri.fr/>

<sup>2</sup> COGIT: Object Oriented Topographical Information Management

<sup>3</sup> INSPIRE: <http://inspire.jrc.ec.europa.eu/>

## 4.1 GEONTO Document Features

The excerpt of the XML document shown in Figure 5 concerns the BDTopo database specifications. Object classes `<className>` are distributed over 9 information areas `<packageName>`. The objects of a particular class listed in the `<description type="ExtensionalDefinition">` markup share a single definition `<description type="definition">`, and the same list of attributes `<attributes>`. These attributes may refer either to qualitative information (list of object labels) when they cover `<enumerated Values>` markup, otherwise to quantities. In turn, each attribute value has its own definition and can list object names.

## 4.2 Preliminary Layout Processing

At this stage, we focus on typographical elements that introduce enumerations. In GEONTO specification documents, enumeration marks can be slash or pipe, or the text "Voir les attributs ..." (Cf. attributes ...) which refers to other objects in the text. A parser analyzes the text and tags each element of the list with a specific markup. For example, the following description extracted from Figure 5

```
<description type="extensionalDefinition">
  Allée(carrossable)|Piste|Route empierrée
</description>
```

is parsed into :

```
<description type="extensionalDefinition">
  <listTerm> Allée (carrossable) </listTerm>
  <listTerm> Piste </listTerm>
  <listTerm> Route empierrée </listTerm>
</description>
```

## 4.3 Instantiation of SGPs for GEONTO

The SGPs instances defined for the GEONTO project identify ontological elements in any geographical database specification document compliant with the INSPIRE standard. To exploit the structural elements presented in section 2.2, we instantiate SGP1 and SGP2 according to the semantics present in these documents.

### 4.3.1 Titles and Enumerations

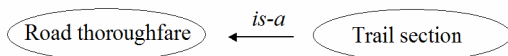
Although title/sub-titles and enumeration elements are different logical structures, they both bear hierarchical relations between a preliminary term and its subordinates. For these two cases, we use SGP1. We show below how SGP1 is instantiated for title/sub-titles present in the document of the Figure5.

```
SGP1("/Document/package","packageName",
      "/Document/package/class", "className") :
```

→

```
((Bal.name="packageName", Bal.path="/Document/package", Term)):T1
({Token})*
((Bal.name="className", Bal.path="/Document/package/class", Term)):T2
({Token})* +
→ Oi = { is-a(T1, T2) }
```

Matching this instantiated pattern on text gives the following fragment of ontology:



```
<package name="A - Road thoroughfare">
```

Figure 3. A first fragment of ontology O<sub>i</sub> obtained with SGP1

### 4.3.2 Definitions

To associate a definition to its concept (for a further linguistic exploitation), we instantiate SGP2 :

```
SGP2 ("className", "description", ("type", "definition"),
      "/Document/package/class") :
```

→

```
((Bal.name="className", Bal.path="/Document/package/class", Term)):T1
({Token})*
((Bal.name="description", Bal.path="/Document/package/class",
  Bal.type="definition", Par)): Par1
→ Oi (Def(T1, Par1))
```

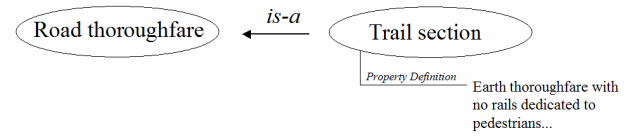


Figure 4. Storage of a definition obtained from SGP2

As a result, the set of all O<sub>i</sub> forms an ontology kernel structured as a taxonomy.

## 4.4 Text Analysis with Lexico-Syntactic Patterns

In the case of GEONTO documents, defined terms have already been recognized as concepts of the ontology (and then annotated as *Concept* in the document). Previously, documents have been annotated by different tools: terms have been annotated by a term extractor (*Term*), properties (*Property*) by linguistic markers that we have defined as rules. We consider here a property any adjective or noun adjunct. The following lexico-syntactic patterns explore the paragraphs identified as concept definitions:

- When a concept is defined by a term :

```
{Concept} {Term}
```

The *Term* is associated to the *Concept* as a label (synonymy relation) (figure 6.(1))

- When a concept is defined by a term with properties :

```
{Concept} ({Property})* {Term} ({Property})*
```

The *Term* will label T, a more generic concept than the one labelled *Concept*: *Concept* is defined as son of T, and it is associated the *Properties* thanks to a "has-property" relation (figure 6.(3)).

- When a concept is defined by a linguistic marker, a term and possibly properties

```
{Concept} {Part-of} {Term} ({Property})*
```

The *Term* is considered as a new concept label and the *Concept* is linked to this new concept by the *part-of* relation (figure 6.(2)).

```

<packageName> A - Road thoroughfare </packageName>
<class name="Trail section">
  <className> Trail section </className>
  <description type="definition"> Earth thoroughfare with no rails dedicated to
pedestrians...</description>
  <description type="extensionalDefinition"> Cf. the different values of the &lt;nature&gt;
attribute </description>
  <attributes>
    <attribute name="Nature">
      <attributeName> Nature </attributeName>
      <valueType> Enumerated </valueType>
      <description type="definition"> Makes it possible to differentiate several kinds of earth
thoroughfares </description>
      <enumeratedValues>
        <value name="Stone path">
          <valueName> Stone path </valueName>
          <description type="definition"> Briefly surfaced road or stone path ... </description>
          <description type="extensionalDefinition"> Lane carriageway | Track| Gravel road
</description>
        </value>
        ...
      </enumeratedValues>
    </attribute>
    ...
    <attribute name="Name ">
      <attributeName> Name </attributeName>
      <valueType> Character </valueType>
      <description type="definition"> path name </description>
      <enumeratedValues/>
    </attribute>
  </attributes>
</class>
...
</packageName>
</package>

```

Figure 5. Translated excerpt of the XML document

Figure 6.(4) shows a selection of all these cases.

(1)	[Cascade]Concept : [Chute d'eau]Term
	[Cascade]Concept : [Waterfall]Term
(2)	[Tronçon de cours d'eau] Concept : [Portion de]LMarker Concept [cours d'eau]Term
	[Stretch of river] Concept : [Section of]LMarker Concept [river]Term
(3)	[Terrain de sport]Concept : [équipement sportif]Term [de plein air]Property
	[Sports field]Concept : [outdoor]Property [sports equipment]Term
(4)	[Surface de route]Concept : [Partie de]LMarker [la chaussée d'une route]Term [caractérisée par une largeur exceptionnelle]Property
	[Road surface]Concept : [Part of]LMarker [carriageway]Term [characterized by an exceptional road width]Property

Figure 6. Definition examples

Before creating a new concept labelled by the Term, we check that this concept does not already exist in the ontology. If not, when possible, we insert this concept using a lexical inclusion algorithm.

As an illustration, let's consider the *Road surface* (*Surface de route*) definition concept of the Figure 6. To refer to Figure 7, we give into brackets each term translation.

The *Road surface* concept will be linked to the *Road carriageway* (*Chaussée d'une route*) concept by the part-of relation. If the *Road carriageway* concept is not defined yet in the ontology, we look for a more generic concept (*Carriageway* (*Chaussée*)) which would be the father concept. If it does not exist, the *Road carriageway* concept is linked to the *Top* concept. The *exceptional road width* (*largeur de route exceptionnelle*) property is associated to the *Road surface* concept by the DataProperty *is-characterized-by*. If *Carriageway* concept exists, *Road carriageway* is set as one of its children.

To sum up, in this section, we have shown that rules browsing XML tags and their relationships can help to get a conceptual taxonomy as an ontology kernel. Then rules implementing more classical lexico-syntactic patterns can extract new concepts, relations, terms and properties from the annotated natural language paragraphs and enrich this ontology kernel.

## 5. IMPLEMENTATION

### 5.1 Tools

A first idea for the manipulation of XML document structure is to use XSLT language for instance. We rather chose to implement all our system using the GATE platform since it supports both structure manipulation, annotation devices and NLP tools.

The GATE<sup>4</sup> NLP platform allows developing pipeline processes including all the ontology learning stages. In fact, GATE can read any well-formed XML document and markups are converted into native GATE format. The tag names constitute annotation types and all the tag attributes will be materialized in the annotation features. Moreover, in GATE, a process is defined as a pipeline that runs text processing tools (like parsers, tokenizers, etc.) or refers to linguistic resources (like ontologies, gazetteers, etc.). Each step adds new annotations to the input corpus, each resource leaning on annotations obtained from the resources previously applied. As long as GATE makes available an *Ontology API*, it is easy to build up an ontology by parsing these annotations with JAPE Rules or Java programs. Hence, the GATE platform allows the definition of a unified process that supports ontology learning from XML documents:

- XML tags are exploited as annotations, and tag dependencies as overlapping annotations to get a first ontology kernel
- NLP tools produce morpho-syntactic and semantic tags, that are later processed to enrich the ontology.

We show an excerpt of the ontology resulting from processing the specifications given in Figure 5.

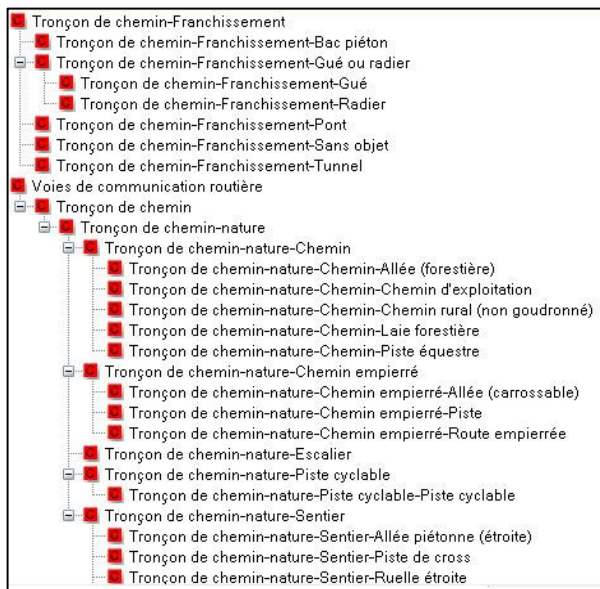


Figure 7. Excerpt of the ontology

The *Trail section* (*Tronçon de chemin*) (term tagged as a <className>) concept is a child of *Road thoroughfare* (*Voies de Communication Routière*) (<PackageName>) and has *Trail* (*Chemin*), *Dirt road* (*Chemin empierré*), *Stairs* (*Escalier*), *Cycling path* (*Piste cyclable*) and *Dirt road* (*Sentier*) (all these terms were <values>) as children concepts. The three children concepts of *Dirt road* (*Chemin empierré*) are those corresponding to the terms tagged as <listTerm>: *Carriage lane* (*Allée carrossable*), *Track* (*Piste*), *Dirt road* (*Route empierrée*).

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

The *Trail section* (*Tronçon de chemin*) has an attribute (represented as *DataTypeProperty*) labelled *has\_name* (*a-pour-Nom*) and is of type *String*. *Trail section* is related to the concept *Crossing* (*Franchissement*) by the semantic relation *has\_Crossing* (*a-pour-Franchissement*) modelled as *ObjectTypeProperty*.

Moreover, according to the *Trail section* definition ("Earth thoroughfare with no rails dedicated to pedestrians..."), the concept *Thoroughfare* is created and related to the concept *Top* and has the concept *Road thoroughfare* as child. Then the properties *earthly*, *with no rail*, *dedicated to pedestrians* are linked to the concept *Trail section*.

As the concepts are created, they are documented: the "comment" property indicates whether this concept definition comes from the structure or from the text processing.

## 5.2 Evaluation

Due to the formulation of the specifications, no statistical method can provide significant results: natural language paragraphs are very short with few redundancies, each term has a very few occurrences, many terms are 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). Therefore, in order to estimate the gain brought by our method, we compare it with other approaches.

### 5.2.1 Comparison with other approaches

We have compared the effectiveness of three methods by comparing the richness of the resulting ontologies. *Onto\_SV* results from the text layout (Laurens, 2006), *Onto\_SR* from the Relational Database schema (Abadie, 2009) and *Onto\_ST* is the one obtained with our approach. These three ontologies were created from the same database specifications. Table 1 compares several features of these ontologies.

Table 1: Features of the three ontologies

	Onto_SV	Onto_SR	Onto_ST
Number of concepts	615	?	1251
Depth	6	3	6
Hierarchical IS_A relations	yes	yes	yes
Terms	No	No	yes
Properties	No	No	yes
Meronymy relation	No	No	yes
Other semantic relations	No	No	yes
Learning process	Supervised	Unsupervised	Unsupervised

Abadie's approach is perhaps the most automatic one but provides a very flat ontology with very poor semantics and a lot of wrong concepts. For *Onto\_SV* and *Onto\_ST*, setting up the document analysis process requires a manual interpretation. Human

interpretation was also required at several other stages of the Onto\_SV development process: (I) to select/validate geographical terms, (II) to clean up the XML hierarchy before the automatic generation of the concept taxonomy, (III) to reorganise and improve the OWL representation of this taxonomy. An opposite option was chosen to build the Onto\_ST ontology: the ontologist is supposed to modify the ontology (only once it is automatically generated) in order to correct inconsistencies due to errors in the specification.

The high number of concepts in Onto\_ST is due to the fact that in specifications the same term may occur in different sections of the document. Then, each occurrence refers to a different concept (they differ in their definition and properties). For instance, the term *Cove* appears twice in the ontology: once it labels a child concept of *Section of Watermark*, another time, it is represented as a *kind-of Bay*, which is in turn a kind-of *Hydronym*. A first solution could consist in creating a single *Cove* concept, with several father concepts. This solution could not reflect the diversity of definitions and properties that the document provides for each of these concepts. To follow the document structure, we decided to concatenate the name of a given concept to the ones of its fathers (Figure 7). This option allows differentiating the two *Cove* concepts. Moreover, it provides a trace from the ontology back to the source documents used to build it.

While Onto\_ST is not the best domain ontology regarding concept definitions, relations or its hierarchical structure, it is however the closest one to the domain knowledge as expressed in the specification document. In addition, Onto\_ST is built up automatically, concepts are documented to help consistency checking, and numerous terms label each concept, which will facilitate the alignment phase.

### 5.2.2 Advantages and Limitations 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 such as these specifications. Whatever is the effort made by their authors, meaning variations (lexical, syntactical or related to the text layout) are one of the features of natural language in text. While processing the document, we note some inconsistencies.

#### a) problem occurring in the concept hierarchy

Enumerations often describe sibling concepts rather than a hierarchy. For instance, a field that should list children concepts of a kind of road sets *Streets* and *Pedestrian streets* at the same level where as *Pedestrian Streets* could be considered as a kind of *Street*. Many other similar cases have been found in the enumerations

#### b) inconsistencies in semantic relations

The *Road portion* concept is defined from the *Road thoroughfare* domain. The rule that interprets tags leads to define these two concepts as well as a hierarchical *is-a* relation between them.

On the other hand, a linguistic pattern for meronymy matches the definition field of *Road portion*. This definition says that a *Road*

*portion* is a part-of *thoroughfare* (dedicated to cars). According to the meaning of these two hierarchical relations, it is not relevant that they exist together between the same concepts. Only a human intervention (or additional domain knowledge) could decide which one to keep.

## 6. CONCLUSION AND FUTURE WORKS

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 proves efficient for ontology learning. This chain, implemented with the GATE platform, includes rules that browse the XML tags and their relations to get an ontology kernel, and rules that extract new concepts, relations, terms and properties from the natural language text that enrich this kernel. The first set of rules extends the type of information usually taken into account by relation extraction from text. The ontology obtained with this automatic process results rich in concepts and relations, and each of its elements is precisely connected to the text from which it originates. This method is applicable to any XML French documents referring to database specifications and validated by the INSPIRE standard.

We are aware that, like the specification documents, this ontology contains some inconsistencies that should be manually corrected. In the scope of the GEONTO project, ontology manual cleaning is planned just after the alignment of several ontologies learned from several specification documents.

We plan various extensions to take into account additional document features and to apply this approach to any database specification. The document features that could be taken into account include the lay-out, data tables and graphics. To better explore natural language paragraphs, we must define new lexico-syntactic patterns that manage disjunctions and conjunctions. Provided new pattern instances are implemented, our tool can be easily adapted to geographical database specifications compliant with the same XML schema but written in a different language.

Another means to enrich the current ontology would be to import concepts and relations from external resources. In GEONTO, the LIUPPA, one of the partners, will provide an ontology of mountain itineraries, with terms, concepts and relations extracted from journey notes, journals or novels.

## REFERENCES

- [1] Abadie N., 2009, Schema Matching Based on Attribute Values and Background Ontology, *12th AGILE International Conference on Geographic Information Science*, 2-5 June, Hannover, Germany.
- [2] Abadie, N., Mustière S., 2008. Constitution d'une taxonomie géographique à partir des spécifications de bases de données. In *SAGEO'08*, Montpellier
- [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. In *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. TerminWeb: 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] Bizer, C., 2003. D2R MAP – A Database to RDF Mapping language. In *Proceedings of the Twelfth International World Wide Web Conference (WWW)*
- [7] 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
- [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] Gardarin, G., Bedini, I., Nguyen, B., 2008. B2B Automatic Taxonomy Construction, *ICES (3-2) 2008*: 325-330.
- [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.
- [13] Grefenstette G. (1994), *Explorations in Automatic Thesaurus Discovery*. Boston, MA: Kluwer Academic Publisher.
- [14] Hearst, M.A., 1992. Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th conference on Computational linguistics*, Morristown, NJ, USA, ACL, 539-545.
- [15] 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.
- [16] Jacquemin, C., 1997. Présentation des travaux en analyse automatique pour la reconnaissance et l'acquisition terminologique. In *Séminaire du LIPN*, Université Paris 13, Villetaneuse.
- [17] Jacques, M-P., 2005. Structure matérielle et contenu sémantique du texte écrit. *Corela, Volume 3, Numéro 2*.
- [18] Laurens, F., 2006. Construction d'une Ontologie à partir de Textes en Langage Naturel. Rapport de Stage Master 1 en linguistique-Informatique, September 2006.
- [19] Lemarié J., Lorch L., Eyrolle H., Virbel J., 2008. SARA: A text-based and reader-based theory of text signaling. *Educational Psychologist*, Lawrence Erlbaum Associates, Mahwah - USA, Vol. 43, p. 1-23, février 2008.
- [20] Luc, C., 2001. Une typologie des énumérations basée sur les structures rhétoriques et architecturales du texte. *TALN2001*, Université de Tours, juillet 2001, p. 263-272.
- [21] Maedche, A., 2002. *Ontology learning for the Semantic Web*, vol 665. Kluwer Academic Publishing.
- [22] Marshman, E., 2007. *Lexical Knowledge Patterns for the Semi-automatic Extraction of Cause-effect and Association Relations from Medical Texts: A Comparative Analysis of English and French*. Ph.D. Thesis, Département de linguistique et de traduction, Université de Montréal.
- [23] Nédellec, C., Nazarenko, A., 2003. *Ontology and Information Extraction*. in S. Staab & R. Studer (eds.) *Handbook on Ontologies in Information Systems*, Springer.
- [24] Rebeyrolle, J., Perry-Woodley M.-P., 1998. Repérage d'objets textuels fonctionnels pour le filtrage d'information : le cas de la définition. In *Actes Rencontre Internationale sur l'Extraction, le Filtrage et le Résumé Automatiques*, pp. 19-30. Sfax, Tunisie, novembre 1998.
- [25] Rebeyrolle, J., Tanguy, L. 2000. Repérage automatique de structures linguistiques en corpus: le cas des énoncés définitoires. *Cahiers de Grammaire*, 25, 153-174
- [26] Tirmizi, S., Sequeda, S., Miranker, J.F, 2008. Translating SQL Applications to the Semantic Web. *Dexa 2008*, Turin , Italie, 450-464.
- [27] Virbel, J., Luc, C., 2001. Le modèle d'architecture textuelle: fondements et expérimentation. *Verbum*, Vol. XXIII, N. 1, p. 103-123.