Modelling and automatic extracting of contextual semantic annotations

Noureddine Mokhtari, Olivier Corby
INRIA
2004 route des lucioles - BP 93
FR-06902 Sophia Antipolis cedex France
{Noureddine.Mokhtari, Olivier.Corby}@sophia.inria.fr

Abstract: In order to reach the semantic Web, approaches to automatically extract semantic annotations from textual documents have been proposed. In this paper we propose an approach to automatically extract annotations by taking into account context in order to obtain a better representation of the document content. Our context is modelled by contextual relations built up from both the structure and the semantics of the text. Our approach requires text documents and a domain ontology as input. It automatically generates a set of contextual semantic annotations represented in RDF.
Keywords: semantic annotation, annotation extraction, rhetorical relation, context, ontology.

1 Introduction

These last years, many works have been performed to semi-automatically extract annotations from web resources in order to reach the semantic Web. In the field of textual semantic extraction, an important step forward has been realised through the availability of automatic natural language processing (NLP) tools. These tools are generally based on linguistic methods such as morpho-syntactic patterns matching [Aussenac-Gilles, 06] or on statistical methods such as frequency of terms co-occurrences. The approaches used so far are generally based on term extraction. Some of them enable also the extraction of relations between these terms. But in most cases, the context where these terms appear is ignored. This observed limitation of term extraction approaches was our main motivation to propose a new approach of modelling and extracting annotations, which takes into account the context in order to give a better representation of the document content. In this article, *semantics* is considered *as the representation of a linguistic unit (of text) in a knowledge representation formalism which allows inferring on this semantics*. From our particular point of view, the *semantic annotation (SA)* of a document is considered as *a snapshot of its content done by an annotator (human or program). This SA must be machine readable.* Our work was carried out in the framework of the SEVENPRO[1] European project whose objective was to develop a virtual environment annotated semantically to design products, in order to assist engineers in their new products design and to allow the exploitation of both textual document semantics and 3D representations.

This paper is organised as follows. First of all, in *section 2*, we present a brief overview of related work on textual annotation. Then in *section 3*, we propose a model of context and contextual relations. In *section 4*, we detail the contextual

[1] http://www.sevenpro.org/

semantic annotations extraction approach and we give a case study showing how the annotations are extracted. In *section 5*, we expose score results of the approach experiments. In *section 6*, our conclusions are presented.

2 Semantic annotation

This section focuses especially on both approaches, automatic and semi-automatic, for generating annotations taking into account the notion of context. However, more details are provided in [Uren, 06] for a more thorough understanding of semantic annotation and a state of the art of the current generation of semantic annotation systems and platforms. There are two ways to extract annotations. The first one relies on the content of the document itself and the second one relies on external information sources (context of redaction, external hyperlink…). For the first one, there are two techniques to annotate documents by their contents. The classical one usually consists in associating a set of keywords to each document. The second one, called semantic, annotation assigns to each document an annotation based on *concepts* defined in an ontology and possibly their relations [Guarino, 99][Khelif, 07]. In [Desmontils, 02], the author proposes a supervised approach to index Web resources. In [Vargas-Vera, 01], the author describes a semantic annotation tool for extraction of knowledge structures from web pages through the use of simple user-defined knowledge extraction patterns. Other extraction methods are dedicated to specific areas such as genomics [Nédellec, 04] for example. For the second way, the basic idea proposed by [Njmogue, 04] is that the indexation of a document depends on the activities of the organisation and not on the keywords present in the document. In [Abrouk, 06], the author proposes a semi-automatic annotation approach based on the references cited in a document and without prior knowledge of its content. The authors [Naing, 03] propose an ontology based method to automatically extract hyperlink information about relation instances from a web site.
The challenge is to propose an approach which models the context for the two ways of annotations extraction. In addition, the approach must give a concrete process to extract annotations for one of these ways.

3 Contextual semantic annotation modelling

We are interested in extracting contextual semantic annotations from texts. Therefore, the objects handled are of textual type. A "*textual object (TO)*" is defined as a text element (*word*, *sentence*, *title*, *text between bracket*, *paragraph*, *section*, *part of a sentence…*) which conveys semantics. A semantic conveyed by a TO is called *semantic annotation*. The SA is represented in RDF formalism and it can be *class(s)*, *triple(s)* and *named graph(s).* The use of any semantics is indeed tightly depending on the context it is located in.

McCarthy [McCarthy, 93] defines the context as the generalisation of a collection of assumptions. Contexts are thus represented as first order formal objects. McCarthy assumes that a proposition p is true in a context c, where c is supposed to capture everything that is not explicit in p but that is required to make p a significant statement, for what it is supposed to represent. Such a basic relation (between the proposition *p* and its context) is always given in another context. In [Brézillon, 03], the author proposes the following definition: "*The context is the information that characterises the interactions between humans, applications, and environments.*"

With regards to textual document processing, we define the context as (Fig. 1)*:*
" The context of a given TO is a tuple of sets <TOs,RCs> where: the TOs are the textual objects interacting with it and the RCs are the contextual relations (structural, temporal and others) implied in the different interactions".
" The context of a given SA is a tuple of sets <SAs,RCs> where: the SAs are the semantic annotations interacting with it and the RCs are the contextual relations (spatial, temporal and others) implied in the different interactions".
Consequently, "*the contextual SA is this SA with its context".*

In contrast to the relations between *concepts* which have been proposed to represent knowledge, the contextual relations, proposed here represent the relations between TO and between SA.

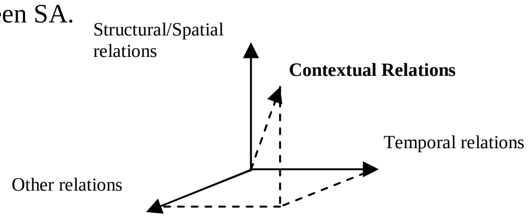

Figure 1 : Three dimension representation to allow modelling context

In addition, the notion of granularity allows us to reinforce our vision on the fact that contexts are nested within each other. The context notion exists at any granularity level. Therefore, it is possible to study the contextual relations not only between the TOs on the same detail level, but also between TOs belonging to different levels. Moreover, when the handled TOs are the documents themselves, we can regard relation among documents as contextual relations such as *cited references, hyperlink, etc.* In this paper, we focus on existing contextual relations within documents.

A temporal CR is a relation expressing the time notion among TOs (resp. SAs) within the same granularity level or not. For instance, with regards to SAs (*before, during....*) and with regards to TOs, we also consider verb tenses (*present, future…*).

A structural (resp. spatial) CR is a relation expressing the relative position of TO (resp. SA) among then within the same granularity level or not. With regards to TOs, examples of structural relations are *succession, belonging….* With regards to SAs, spatial discourse markers express a rank (*front, after...*) or location (*in, on, under...*).

Other CR, that is neither spatial/structural nor temporal, is a relation expressing a semantic notion among TOs (resp. SAs) within the same granularity level or not. With regards to TOs, an example of CR is the degree of importance between a paragraph and its title. With regards to SA, any discourse marker expressing either an addition (*moreover, in fact…*) or an illustration (*such as…*) for example is considered as a CR. We will detail the use of *discourse relations* in the next section.

So far, the context notion and contextual SAs are modelled. It remains to expose how it is possible to identify the TOs, the CRs and how to generate SAs of TOs?

4 Extraction of contextual semantic annotations

As said above, several levels of granularity exist with regards to SAs, depending on the TOs chosen (sentence, paragraph…). The difficulty is to choose the best granularity to produce the best *SA*. The proposed approach focuses on a particular level of granularity that is deemed semantically rich. The TOs handled at this level are delimited by "*discourse relations*". *Discourse relations* [Taboada, 06] in Rhetorical Structure Theory (RST) domain are also called coherence relations, conjunctive relations or rhetorical relations. Some works have already been done to identify these discourse relations in the text [Marcu, 02][Teufel, 00][Saito, 06]. Another approach is proposed by [Desclés, 06] to automatically annotate documents by using the "semantics of speech". This method relies on a rule engine enabling to identify the text segments that contain a *definition*, a *cause*....In our approach, we focus on the identification of rhetorical relations explicitly marked. In the example: [Jack failed the exam] *because* [he was lazy], the first argument in brackets is the *fact*, the second argument is the *cause* and *because* the explicit marker. Explicit markers existing between *arguments* (sentences or parts of text) represent the three dimensions of contextual relations: temporal, spatial/structural or semantic.

4.1 Contextual semantic annotation extraction process

Two main stages constitute our process: "*textual handling*" stages followed by the "*semantic handling*" stages (see Figure 2).

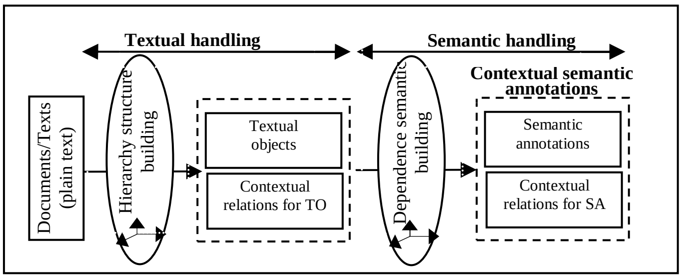

Figure 2: Contextual semantic annotation extraction process

3 Description of the mill internal elements
3.1 Inlet Headliners
<u>This new design is composed of 3 thicker bolted rings, compared to the original design of 2 rings.</u> The liners have a thickness of XXmm. except for the area of most wear (R/2-R/2+R/3), where the thickness is YYmm.

Figure 3: Text excerpt

Figure 3 shows an excerpt of document provided. It describes the components of an industrial machine (engineering mill), and the assembling procedure.

4.1.1 Text handling stage

The aim of this stage of our exploitation process is to identify the *textual objects* and the *contextual relations* between them.

The textual objects identification step consists in identifying *titles*, *phrases*, *discourse markers*, *words*… as well as the *arguments* of each discourse markers in the text. The engineering GATE platform [Cunningham, 02] has been used to identify TOs. It is based on successive application (pipeline) of transducers[2] to the texts. A library of contextual relations is collected from both the *discourse markers* and other *spatial, temporal and semantic contextual relations.* For each contextual relation of this library, a JAPE[3] rule is generated automatically to obtain their positions in the text. Other heuristics are considered and manually transformed into JAPE rules. For instance, a JAPE rule (Figure 4) allows identifying the *numerical indicators* preceding some sentences (titles) such as "7.5.3.".

```
Rule:TitleNumber
({Token.kind == number}(({SpaceToken.kind == control}
({SpaceToken.kind == space})*)|
(({Token.lemma == "."}{Token.kind == number})+
{SpaceToken.kind == control} ({Token.lemma == "-"})*
({SpaceToken.kind == space})*)))
:TitleNumber --> :TitleNumber.TitleNumber = {kind =
"titleNumber", rule=TitleNumber}
```

Figure 4. The JAPE rule to identify the numerical indicators

The JAPE rules are applied as transducers in the GATE pipeline. In addition, indications of position such as *start* and *end* in the text of *sentences*, *discourse markers*… are used to identify the *discourse markers arguments* in the text.

The contextual relations identification step requires to build the text hierarchical structure: *(a)* First, titles are identified by using *numerical indicators* preceding sentences and heuristics such as "*GATE transducer identifies titles as paragraphs. If a paragraph contains a single sentence which starts by a numerical indicator, then this sentence is a title*". It has to be noted that the default transducer provided by GATE to identify titles is not very accurate; *(b)* then the scope of the detected titles and in their hierarchy (i.e. a paragraph or a subtitle belongs to a title) are deduced; *(c)* finally, the nesting between paragraphs, sentences and arguments is built by using position indicators in the text. Once the hierarchical structure of the text is built, contextual relations are deduced. For example, the sentence 's1' *belongs to* the paragraph "p1".

4.1.2 Semantic handling stage

This stage aims at identifying SAs and contextual semantic relations.

The semantic annotations generation step aims at representing the semantics of TOs within a knowledge representation formalism. The chosen formalism is RDF(S). RDF is based upon the notion of triples (*resource*, *property*, *value*). To associate RDF triples to TOs by referring to the ontology, we propose to identify *resources* (or

[2] A transducer is a finite state machine which enables to produce for each visited state, one or several information.

[3] JAPE (Java Annotation Patterns Engine) is the language for expressing grammars offered by the platform GATE (an example is given in 4.2.3 section).

concepts or *classes*) and *properties* in the text. Therefore, we propose to build automatically a set of JAPE rules. Indeed, the main idea is to use the value of the "*rdfs:label*" property in a RDFS schema to build JAPE rules.
The *ClassesJapeRulesBuilding algorithm* explains how associated JAPE rules to "classes" are generated:

```
ClassJapeRulesBuilding_Algorithm {
  Classes ← ListOfClass(ontology); Rules ←"";
// ListOfClass request ontology by using SPARQL4
// query to get the Class list and their Labels from
// ontology
For each Class ∈ Classes do {
HeadRule←HeadRuleBuilding(GetNameOf(Class));
   EndRule←EndRuleBuilding(GetNameOf(Class));
   BodyRule ←"";
  For each label ∈ GetLabelsFrom(Class) do {
      BodyRule←BodyRule +
"("+BodyRuleBuilding(Label)+")";
      If (ExistNextLabel) then {BodyRule ←
BodyRule + "|"};
      //in Jape language "|" is the operator "or"
  }//endFor
   Rules ← Rules +HeadRule + BodyRule + EndRule;
} //endFor
} //End of ClassJapeRulesBuilding_Algorithm
```

```
String HeadRuleBuilding(nameOfRule){
   Return "Rule: JRule"+ nameOfRule ;
}

String EndRuleBuilding(nameOfRule){
   Return "):nameOfRule--
>:nameOfRule.Class={kind= "nameOfRule",
rule=JRulenameOfRule }";
}

String BodyRuleBuilding(label){
    Body ←""; Words ←GetWordsOf(label);
    For each w ∈ Words do {
       wlemma ← lemmaOf(w);
       Body←Body + "{Token.lemma=="wlemma"}
({SpaceToken})?";
    }//endFor
   Return Body;
}
```

For each class, *head rule* and *end rule* are built by *HeadRuleBuilding and EndRuleBuilding* functions. The body rule of class is built by using BodyRuleBuilding function with lemmatised words of labels. If there are more than one label for class the operator "|" is added among parts of body rules.

The algorithm of properties rules building is the same as for classes. In this case, the SPARQL query returns properties list and their labels.

```
<rdfs:Class
rdf:ID="RingOfDiaphragmMill">
<rdfs:subClassOf rdf:resource="#Item"/>
<rdfs:label xml:lang="en">Ring of mill
diaphragm </rdfs:label>
<rdfs:label xml:lang="en">Ring of
diaphragm </rdfs:label>
<rdfs:label
xml:lang="en">Ring</rdfs:label>
<rdfs:comment xml:lang="en">denotes a
part of a diaphragm mill (i.e.ring).
 </rdfs:comment>
</rdfs:Class>
```

```
Rule:JRuleRingOfDiaphragmMill (
({Token.lemma=="Ring"}({SpaceToken})?
 {Token.lemma=="of"}({SpaceToken})?
 {Token.lemma    =="mill"}({SpaceToken})?
{Token.lemma=="diaphragm"}({SpaceToken})
?) |
({Token.lemma=="Ring"}({SpaceToken})?{To
ken.lemma=="of"}({SpaceToken})?{Token.le
mma=="diaphragm"}({SpaceToken})?)
|
 ({Token.lemma =="Ring"}({SpaceToken})?)
):RingOfDiaphragmMill-->:
RingOfDiaphragmMill.Class={kind="RingOfD
iaphragmMill",
rule=JRuleRingOfDiaphragmMill }
```

Figure 5: Domain ontology excerpt and its associated JAPE rules automatically generated

Figure 5 shows an example of RDFS description of the class "*RingOfDiaphragmMill*" in the domain ontology. It also shows the associated "JAPE rule". The "*Token.lemma*" refers to all the possible variants for a word. Afterwards, JAPE rules are built to detect *candidate values* of properties such as numbers in the

[4] http://www.w3.org/TR/rdf-sparql-query/

text. Thereafter, all JAPE rules are introduced in the *pipeline* to locate *instances* of classes, *properties* and *candidate values* of properties in the text. Then it is possible to build RDF triples. However, ambiguity issues may arise. Indeed, different *classes, properties or values* may be detected in a same TO. To solve this problem, we check the *range* and the *domain* constraints in order to associate the right class to each property. However, the opposed problem may arise, i.e. no existing class to associate to properties of TO. To solve this issue, the granularity level is increased. These solutions are implemented in the *semantic annotations generation algorithm*:

```
Semantic annotations generation algorithm
TO: textual object (argument) deduced from hierarchical structure; TOs : set of TO;
TO_max: The biggest TO considered (for example: all of the document, section, paragraph,…);
INPUT : TOs, TO_max;
SA: semantic annotation that we want to associate to TO;
For each TO ∈ TOs do {
TO'←TO;P←set_of_properties_in(TO');
SA←Null;
 While ((TO' ≤ TO_max) and (P ≠ ∅)) {
   For each pj ∈ P do {
     RDFTriplesGenerating_Algorithm (pj,TO',SA);
     If(used_for_RDF_generation(pj))then {P←P-{pj};}
   } //endFor
   If (P ≠ ∅) then {TO' ← TO'.IncreaseContext()
// Increase the granularity level: from the argument,
// to the sentence, to the //paragraph, then add the
// scope of subtitle, then add the scope of title...
   } //endIf
 } //endWhile
 If(notNull(SA))then{AssociateObjects(TO,SA)}
}//endFor
```

```
RDFTriplesGenerating_ Algorithm(p, TO, SA) {
C ← set_of_classes_in(TO) // set of classes in TO
c1 ∈ C(domain(p)) //c1 is a class which verifies the
// domain constraint of the property "p" and if
// several classes exist, then select in the ontology
// the class which is the most specific class and
//which has the shortest distance with "p" in
// the text
c2 ∈ C;  I ← set_of_values_in(TO)
inst ∈ I // Select from values inst which has the
// shortest distance with "p" in the text and
// preferably after "p"
If (notNull(c1)) then {
 If (range(p)="literal") then{
SA←SA+GenRDFtriples(c1,p,GetLabelInText(p));
// The label corresponds to the 'string' value of the
//properties found in the text
 }
 If (range(p)="float") and (notNull(inst)) then {
       SA←SA+GenRDFtriples(c1,p, inst)}
 If(range(p)=c2)then{
SA←SA+GenRDFValueOfClass(c2,GetStringInText(c2)) ;
  SA←SA+GenRDFtriples(c1,p,
GetStringInText(c2)) ;
 }
 }}end_of_RDFTriplesGenerating_ Algorithm
```

This algorithm takes as input the TO extracted at the lowest granularity level considered (*argument*). For each TO, it identifies properties occurring in the text, and subsequently, an attempt is made to match each property with the class it is a property of and its value. If for some properties in the text, the algorithm fails to create the triple, a larger context is sought. Figure 6 shows generated contextual SAs corresponding to the underlined sentence in the text excerpt of figure 3.

The SAs are represented in RDF (XML syntax). In addition, figure 6 shows the contextual relation "*compareTo*" identified between the two SAs.

The contextual semantic relations identification step of the semantic handling stage of our extraction process aims at assigning semantic roles to the discourse markers already detected. In [Marcu, 02][Sporleder, 06], authors propose to automatically identify these roles (*contrast, continuation, explanations…*). However, some problems persist in complex ambiguous "discourse markers". In addition, no automatic tool to identify semantic role exists in "RST" domain. The scope of this work is limited to the identification of discourse markers locations in the text.

Figure 6: Contextual relations dependency between semantic annotations

5 Experiment result

Experiments have been carried out on texts written in English. The proposed approach has been experimented on a corpus of 313 sentences (3768 words and 1862 other linguistic units such as: numbers, commas, brackets…) written by the industrial partners of the European project SEVENPRO. 114 rules are thus automatically generated corresponding to 64 classes and 50 properties in the domain ontology and 80 other rules to identify discourse markers.

	Identifications		Total existing	Precision (%)	Recall (%)
	Right	Wrong			
Titles and hierarchy	26	0	26	-	-
Sentences/ Paragraphs	285/ 205	101/0	313/ 205	-	-
Arguments	497	-	-	-	-
Discourse markers	113	16	127	87,59	88,87
Properties	174	30	195	85,29	89,23
Classes	436	126	458	77,58	95,19
Candidate values properties	413	15	413	-	-

Table 1: Evaluation of the extraction results

Table 1 lists the score results of all elements extracted from the text. The validation (right/wrong) of all elements extracted is verified manually. Two parameters are computed to identify the quality of the generated SAs: the precision and the recall. Getting the best scores, involves a trade-off between these two parameters. All titles and their hierarchy are identified correctly. The *sentences* and *paragraphs* are identified by default transducer of GATE. The score of identification *argument* relies on *sentences* identification score. The wrong scores (see *Table 1*) of *discourse markers, properties* and *classes* identifications is due to duplicate identification of the same words. For instance, the rhetorical relation "*as well as*" is identified as three rhetorical relations: two "*as*" and "*as well as*".
Furthermore, the proposed approach has been experimented on large corpus of 2422 sentences. The "Table 2" compares the RDF triples extraction result within various levels of granularity. The granularity is an important factor. Indeed, it impacts greatly the generated SAs. The existing classes within a sentence of title are used to generate RDF triples in TOs to get better result. We observe that the precision of generated RDF triples is satisfactory when TO_max is an *argument.* On the other side, the recall is unsatisfactory. When the TO_max is increased, the precision decreases a little and the recall increases. When the TO_max is increased to section, the precision has a significant decrease. Indeed, the precision of generated triple relies on the proximity among property, class and value. The trade-off between precision and recall gives the best RDF triples score when the TO_max is paragraph.

		Identifications		Total existing	Precision (%)	Recall (%)
		Right	Wrong			
Generated RDF triples when TO_max is :	Argument without title phrase	191	29	756	86,82	25,26
	Argument + title sentence	539	113	756	82,67	71,30
	Sentence + title sentence	546	118	756	82,23	72,22
	Paragraph + title sentence	593	130	756	82,02	78,44
	Section + title sentence	641	208	756	75,50	84,79

Table 2: Evaluation of semantic annotations generation algorithm: generated RDF triples

6 Conclusions

In this paper, we proposed an approach to model and to extract semantic annotation by taking into account the context of textual sources. The main steps of the proposed approach are summarised as follows: *i)* identification of textual objects; *ii)* identification of contextual relations corresponding to TOs; *iii)* generation of semantics annotations represented by RDF triples; *iv)* identification of contextual semantic relations. All proposed steps are automated, and a prototype is implemented to assess the various steps of this contextual extraction approach. The evaluation results are very satisfactory. This approach of extraction and use of contextual semantic annotations offers promising results in the domain of knowledge extraction from texts. However, two challenges remain to be overcome:

The first one is the exploitation of inference on semantic discourse relations by using semantic Web technologies. Therefore, we propose on one hand to represent discourse relations semantics with inference rules and on the other hand to use the proposals discussed about *Source Declaration*[5] to assign a IRI (Internationalized Resource Identifiers) to a set of triples (named graph) and [Corby, 07] to represent interweaving between *semantic annotations* by related named graphs.

The second challenge deals with the difficulty of inferring on spatial and temporal contextual relations. Therefore, we draw inspiration from *Geographic Information Systems* works. More specifically, we are looking at Allen relations [Allen, 84] to specify temporal contextual relations and Egenhofer relations [Egenhofer, 91] to specify spatial contextual relations.

Acknowledgements

Our thoughts are for *Rose Dieng-Kuntz* who has supervised this work and ensured its completion.

References

[Aussenac-Gilles, 06] Aussenac-Gilles N. et Marie-Paule J. : Designing and Evaluating Patterns for Ontology Enrichment from Texts, EKAW'2006, LNAI 4248, pp. 158–165, Tchèque (2006)

[Uren, 06] Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. Journal of Web Semantics, 4(1):14–28, (2006)

[Guarino, 99] Guarino, N., Masolo, C.,Vetere, G., Ontoseek: Content-based access to the web, IEEE Intelligent Systems, May-June 4-5:70–80, (1999)

[Khelif, 07] Khelif, K., Dieng-Kuntz R., Barbry P.: An ontology-based approach to support text mining and information retrieval in the biological domain, Journal of Universal Computer Science (JUCS), Vol. 13, No. 12, pp. 1881-1907 (2007)

[Desmontils, 02] Desmontils, E., Jacquin, C.: Indexing a web site with a terminology oriented ontology, The Emerging Semantic Web, IOS Press, 181-197, (2002)

[Vargas-Vera, 01] Vargas-Vera, M., E. Motta, J. Domingue, S. Buckingham Shum, and M. Lanzoni,: Knowledge Extraction by using an Ontology-based Annotation Tool. In K-CAP 2001 workshop on Knowledge Markup and Semantic Annotation, Canada, October, (2001)

[Nédellec, 04] Nédellec, C.: Machine Learning for Information Extraction in Genomics - State of the Art and Perspectives, Studies In Fuzziness And Soft Computing, Vol. 138, pp. 99-118, (2004)

[Njmogue, 04] Njmogue, W., Fontaine D., Fontaine P.: Identification des thèmes d'un document relativement à un référentiel métier, Proc. of MAJECSTIC'04, Calais, France, (2004)

[Abrouk, 06] Abrouk, L. : Annotation de documents par le contexte de citation basée sur une ontologie, Phd in Computer Science, Montpellier (2006)

[Naing, 03] Naing, M. M., Lim, E.-P., and Goh, D. H.-L.: On Extracting Link Information of Relationship Instances from a Web Site. ICWS-Europe, pp. 213–226, Germany, (2003).

[McCarthy, 93] McCarthy, J.: Notes on formalizing context, Proc. 13th IJCAI, pp.555–560, California, (1993)

[Brézillon, 03] Brézillon, P. : Représentation de pratiques dans le formalisme des graphes contextuels. In: J.M.C. Bastien (Ed.), EPIQUE'2003, pp. 3-14, Inria, France, (2003)

[Taboada, 06] Taboada, M., Mann, W.C.: Rhetorical Structure Theory: Looking Back and Moving Ahead. Discourse Studies 8(3): 423-459, (2006)

[5] http://www.w3.org/Submission/rdfsource/

[Marcu, 02] Marcu, D., Echihabi, A.: An Unsupervised Approach to Recognizing Discourse Relations, Proc. of the 40th Annual Meeting of the ACL, pp. 368-375, Philadelphia, (2002)
[Teufel, 00] Teufel S., Moens, M.: What's yours and what's mine: Determining Intellectual Attribution in Scientific Text, SIGDAT (EMNLP/VLC-2000), 9 – 17, Hong Kong, (2000)
[Saito, 06] Saito, M., Yamamoto, K., Sekine, S.: Using Phrasal Patterns to Identify Discourse Relations, Proc. of the HLTCNA Chapter of the ACL, pp. 133–136, New York, (2006)
[Desclés, 06] Desclés, J. P.: Contextual Exploration Processing for Discourse Automatic Annotation for Texts, FLAIRS2006, AAAI Press, Florida, pp. 281-284, (2006)
[Cunningham, 02] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, ACL, pp. 235-238, Budapest, Hungary, (2002)
[Sporleder, 06] Sporleder, C., Lascarides, A.: Using automatically labelled examples to classify rhetorical relations: An assessment, Natural Language Engineering, (2006)
[Corby, 07] Corby, O., Faron-Zucker, C.: RDF/SPARQL Design Pattern for Contextual Metadata, In Proc. of IEEE/WIC/ACM Inter. Conference on Web Intelligence, USA, (2007)
[Allen, 84] Allen, J.F.: Towards a general theory of action and time, AI (1984)
[Egenhofer, 91] Egenhofer, M., Franzosa, R.: Point-Set Topological Spatial Relations, International Journal of Geographical Information Systems, vol. 5, no. 2, pp. 161-174, (1991)