Using Description Logics for Ontology Extraction

Amalia Todiraşcu (University Al.I.Cuza of Iasi, Romania and LIIA, ENSAIS, France) [1]
François de Beuvron (LIIA, ENSAIS, France),[2]
Dan Gâlea (Romanian Academy)[3]
François Rousselot (LIIA, ENSAIS, France)[2]

Abstract. The paper presents a prototype of a system for querying the Web in natural language (French) for a limited domain. The domain knowledge, represented in description logics (DL), is used for filtering the results of the search and is extended dynamically, (when new concepts are identified in the texts) as result of DL inference mechanisms. The conceptual hierarchy is built semi-automatically from the texts. Different small French corpora (heart surgery, newspaper articles, papers on natural language processing) have been used for experimenting the prototype. The system uses shallow natural language parsing techniques and DL reasoning mechanisms are used to handle incomplete or incorrect user queries.

1 Introduction

Web searching engines accept user query composed by a set of keywords, written in a command language or in natural language. These systems use index files for retrieving the documents. Indexes can be keywords, terms, syntactic or semantic structures.

User queries are transformed into semantic representations, which are matched to the index items. The semantic representation of the query could be a set of keywords or more complex semantic representations. The performances of these systems are evaluated by two parameters: *recall* (the number of retrieved documents/the number of documents) and *precision* (the number of relevant documents/the number of retrieved documents). Keyword-based searching engines provide bad recall(ignoring synonyms or generalization/specialization handling) and low precision (the answers contain a significant amount of irrelevant information).

Several modern IR (Information Retrieval) systems use semantic resources as filters for improving search results: keywords with multiple-word terms [9], their semantic variations [4], thesaurus (Corelex [2], EuroWordNet [13]) or lists of synonyms [7].Natural language querying needs linguistic knowledge and NLP tools, like conceptual sentence parsers, applying patterns with case constraints to extract information ([9]), or predefined case frames [8].

The design of these systems involves off-line, time-consuming building of resources and provides low flexibility and portability. New concepts are identified in the texts, but only human experts can extend domain model. The disadvantage of these systems is the use of predefined semantic resources (thesaurus, lists of concepts etc.) which are not modified at runtime. IR applications deal with incomplete and erroneous data which need robust methods for parsing. Deep semantic and syntactic parsing methods fail to handle erroneous input and need an important amount of linguistic knowledge.

Another approach used by IR and IE applications provides the use of data-driven acquisition of resources. The use of semantic issues (terms, summaries) as indexes or of the inference capabilities of knowledge representation formalisms are just a few examples of the data-driven acquisition paradigm.

Document Surrogater uses of phrasal terms as indexes, eliminating the ambiguity introduced by single words used for indexing. The algorithm uses a special module to produce a set of significant terms from a focus file prepared by a human expert. An application of this methodology is document summarization [15]. Another approach expands user queries to document summaries that are stored in index files. Summaries contain relevant concepts and relations for each document [11]. Special DL operators are defined for generating document summaries [8].

Other system uses terminological information acquired from texts like FASTER [4]. It identifies multi-words terms and uses them as indexes for the document base. A terminological base is used and is extended by new term candidates (generated by morphological transformations on the terms).

Other systems use DLs as representation formalism of the domain knowledge base. An example is CLASSIC [14], used for manually indexing documents (using the name of the author, the title, the subject) of a digital library, containing documents in XML (Extended Markup Language) format.

We design a prototype of a system for querying in natural language (French) a set of documents. We use semantic resources for filtering the search and we adopt a data-driven methodology for resource acquisition. The domain hierarchy is represented in description logic (DL), providing efficiency and fault tolerance to incomplete or erroneous data. Logic inference mechanisms provided by DL are used to extend dynamically the domain model, and to complete missing information identified from the user query. Building linguistic and domain knowledge requires minimal efforts from the human designer, while it integrates shallow natural language processing techniques.

The system could be easily ported for another domain, due to the dynamic maintenance of the domain knowledge base. The new concepts inferred from the new documents are validated and added to the existing hierarchy. The methodology is not appropriate for unrestricted domains, due to the limited size of the ontology supported by the system.

[1] University "Al.I.Cuza" of Iasi, Computer Science Department, 16, Berthelot Str., Iasi 6600, Romania, Phone: +4032201529, Fax:+4032201490, E-mail: amalia@infoiasi.ro
[2] Laboratoire d'Informatique et d'Intelligence Artificielle, ENSAIS, 24, Bd. de la Victoire, 67084 Strasbourg Cedex, France, Phone: +33388144753, Fax: +33388241490, E-mail:{amalia,beuvron,rousse}@liia.u-strasbg.fr
[3] Computer Science Institute, Romanian Academy - Iasi, 22, Carol Av., Iasi 6600, Romania, Phone: +40-32-146534, Email:dgalea@iit.iit.tuiasi.ro

2 Description Logics

Description logics (DL) are formalisms related to semantic networks and frame systems dedicated to knowledge representation ([1]).

DL structures the domain knowledge on two levels: a **terminological level** (T-Box), containing the axioms defining the classes of objects of the domain (named *concepts*), with their properties and relations (*roles*) with other objects, and an **assertional level**, (A-Box), containing objects of the abstract classes (*individuals*). The main reasoning service available in T-Box is *subsumption* between two concepts, determining which concept is more general. A-Box provides *instantiation* test, determining which concept or role has as individual a given instance.

Some of the basic logical operators which are used for creating complex conceptual descriptions are the following:

DL Operator	Logic expresion	DL Interpretation
C = (SOME Rel D)	$\exists$ Rel.D	there is at least one object belonging to D related by a relation Rel with the objects of C
C= (ALL Rel D)	$\forall$ Rel.D	restricts the co-domain of the relation Rel
C = (AND D1 D2)	D1 $\wedge$ D2	conjunction of conceptual descriptions
C= (OR D1 D2)	D1 $\vee$ D2	disjunction of conceptual descriptions
C = NOT D	$\neg$ D	the complement of a concept
C = $\geq nRel.D$	$\exists y1 \ldots yn$ $(1 \leq i \leq n, R(x, yi) \wedge D(yi))$	there are at least n objects of D in relation Rel with C

Example. The definition

```
(define-concept Mother (AND Woman (SOME hasChild
Child)(ALL hasAge Age)))
```

is interpreted as: a **Mother** is a **Woman** that have at least one child (relation **hasChild**) being an instance of the concept **Child**. For each instance of the concept **Mother**, all the instances related by **hasAge** must be an individual of the concept **Age**.

DLs provide powerful inference mechanisms. At the terminological level, the main reasoning mechanism is the subsumption relation between two concepts (detecting which concept is more general than the other one). A concept description can be checked for satisfiability. Classification is a partially ordering of the concepts in a hierarchy. The A-Box provides consistency test (i.e. if there is a contradiction in the set of statements). Instantiation test detects which conceptual description is instantiated by a given instance, and retrieval inference allows for retrieval the individuals which belongs to a given concept. It provides also the posibility of reasoning about the membership relation between pairs of individuals and relations.

DLs are appropriate for applications dealing with semi-structured or incomplete data, like IR systems.The concepts are defined by their roles and attributes. The instances do not contain all the values of the concept attributes. In some frame-based knowledge representation formalism, the missing values are not allowed, while DL accepts defining incomplete instances.

Example.

```
(define-primitive-concept Person (AND domain (SOME
hasAge Age)))
(define-concept Patient (AND Person (SOME
hasDisease Disease)))
(instance p1 Patient)
```

The last command [4] is not giving any particular value for the age or the disease of the **Patient**, even if **p1** is an instance of the concept **Person**.

It is difficult to handle semi-structured data because they have no precise schemas. An instance restricts the relations with other objects. These objects are not always identified explicitly.

```
(define-concept Patient (AND domain (SOME hasAge
Age)(SOME hasDisease Disease)))
(instance y0 (AND Patient (SOME hasAge 60)))
```

In the example, the instance **y0** of the concept **Patient** is related to "60" (an instance of **Age**) by the role **hasAge**, then there must be some instances of **Disease** related to **y0** by the role **hasDisease**, but we cannot identify them.

A DL standard was proposed [5] and several DLs have been developped and used for different purposes: BACK [6] for natural language processing, CLASSIC [14] for IR, FaCT [3] for designing medical terminologies. CICLOP[5] [10] provides similar expresivity with the other systems: it deals with role hierarchy, inverse roles, multiple hierarchies, transitive roles and features. It also allows disjunctions and SAME-AS for generic roles, unlike CLASSIC [14]. It accepts reasoning simultaneously in several hierarchies (multiple T-Boxes)[12], it implements an A-Box and an optimized tableau calculus algorithm. The CICLOP DL is used for representing the domain knowledge. CICLOP is developed in Java.

3 System Architecture

The prototype of the system integrates several natural language processing modules as well as some logical inference modules. For testing the prototype we use a few small experimental French corpora on heart surgery (70000 words), newspaper articles (300000 words) and NLP articles (250000 words). Most of the examples presented are extracted from the heart surgery corpus. The prototype is partially implemented in Java and in Perl. The system uses common representation formalism for domain knowledge and sense (in DL), which provides powerful inference mechanisms, capable of dealing with incomplete, erroneous data. It integrates shallow natural language processing techniques for text documents (Fig. 1).

The NLP modules use domain knowledge and shallow semantic parsing, in the aim of designing a robust, fault-tolerant querying system. NLP modules are used for extending domain hierarchy, as well as interpreting user queries.

The modules identifies relevant semantic issues (*semantic chunks*), using minimal syntactic knowledge. Complex concepts are inferred by DL mechanisms.

[4] The syntax is DL standard which can be found in [5]

[5] Customizable Inference and Concept Language for Object Processing, developed at LIIA(Laboratoire d'Informatique et d'Intelligence Artificielle), ENSAIS, Strasbourg, France

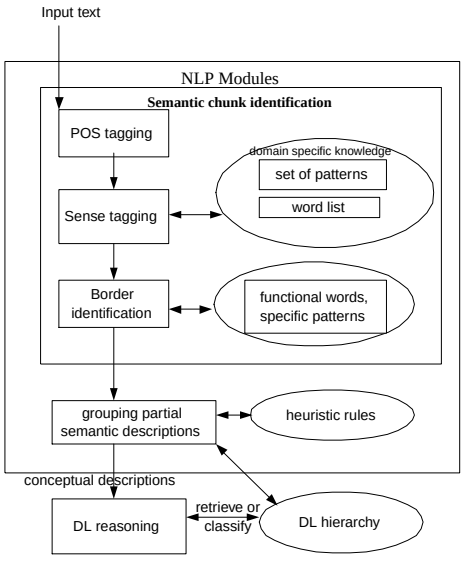

Figure 1. System architecture

The semantic representation of a document is a set of *primitive concepts* (the most frequent of its words and syntagms). Specific concepts are identified by processing the contexts of the instances of primitive concepts.

Example. "le patient, hospitalisé avec infarctus"
[the pacient, hospitalized with heart_attack]
The primitive concept **Patient** has an instance : the phrase "le patient". The phrase is an instance of the more specific **Patient_with_heart_attack**.

3.1 The Modules

I. **Semantic chunks identification**. The main goal of this module is to identify the word sequences corresponding to the most significant domain concepts (semantic chunks). A *semantic chunk* contains a noun and it is delimited by two border words. Border words are functional words, auxiliaries, some prepositional syntagms.

Example. "le patient, ayant eu infarctus"
[the patient, having heart attack]

In this example, "le patient" and "infarctus" are semantic chunks, containing relevant information. The example contains syntactic errors (missing the determiner "un"), but the semantic information is sufficient to understand the query and to return a correct answer.

This module uses several tools: a POS tagger, a sense tagger, a border identifier. The identification of the semantic chunks is based on lexical information, provided by the POS tagger.

a) *The POS tagging* (using WinBrill, trained for French with a set of data provided by INALF - Institut National pour la Langue Française) identifies the content words (nouns, adjectives, verbs) and functional words (prepositions, conjunctions etc.). Brill's tagger uses a set of contextual and lexical rules (based on prefixes and suffixes identification) learned from annotated texts, for guessing the POS for the unknown words.

Example.
`"le/FW degré/NN de/FW sévérité/NN de/FW cette/NN athérosclérose,/NN sa/VB diffusion/NN et/FW sa/VB répartition;/NN"`

`"the degree of this arteriosclerosis, its spread and its repartition;"`

where the labels have the meaning: NN - simple noun; FW - functional word; VB - adverbe or pronominal adjective.

b) *The sense tagger* contains a pattern matcher (implemented in Java), consulting a set of patterns (words, lexical categories and syntagms) and their sense assigned by a human expert. The sense is represented by DL concepts. The set of conceptual descriptions was established by a human expert from a list of the most frequent repeated segments and words extracted from a set of representative texts. The pattern matcher annotates each sequence of words matching the pattern with its semantic description.

Example.
`"il garde [un angor d'effort]/ANGOR_EFFORT en rapport avec [une dysfonction]/TROUBLE du [pont IVA]/BRIDGE alors que [le pont coronaire droit]/BRIDGE est occlus/OCCLUSION mais implanté/hasPlace sur [une artère]/ARTERY [non sténosée]/(NOT STENOSE)"`

`"he presented an effort angor due to a disfunction of the IVA bridge, while the right coronary bridge is ocluded but implanted on a non-stenosed artery."`

We tagged the known syntagms and the words with a set of primitive concepts: **Angor_effort**, **Trouble** (a subconcept of **Disease**), **Bridge**, **Artery**, **Occlusion** and **(Not Stenose)** (subconcepts of **Symptom**), and the role **hasPlace** .

We made some studies for different texts: a set of medical texts, of journal texts and of NLP articles in French. The nouns and the adjectives represent concepts themselves (87 % of the total number of concepts), while prepositions are just delimiters of chunks. Most of the instances of the concepts are complex noun phrases.

c) *A module for border identification.* It identifies the words and the syntactic constructions delimiting the semantic chunks. This module uses the output of POS tagger (identifying the functional words), as well as a set of cue phrases (syntactic phrases containing auxiliaries, composed prepositions etc.). The set of cue phrases is manually built as a result of studies on experimental corpora. The borders of noun and prepositional phrases (determiners, prepositions) are best candidates for chunk border. The same pattern matcher uses the set of cue phrases for border identification.

Example. The label BD means border word:
`"une/BD dysfonction du/BD pont IVA [alors que]/BD le/BD pont coronaire droit est/BD occlus"`

We have an example of cue phrase "alors que" being a border of semantic chunks.

The following lexical categories and syntactic constituents represent the borders of the semantic chunks:

Lexical categories	Percent
Phrase separators	37.73
Prepositions	42.52
Conjunctions	6.57
Auxiliaries	4.2
Verbs	4.16
Adverbs	1.88
Other phrases	2.90

II. **Combining partial semantic representations**. This module applies DL inference mechanisms, as well as syntactic heuristic rules in order to combine conceptual descriptions associated to each semantic chunk. The heuristic rules are established manually by a human expert on a list of patterns ($\langle Chunk1\rangle$ $?x/\langle Border\rangle$ $\langle Chunk2\rangle$) with probabilities, provided by a Perl module, for each test corpus. The test corpora was POS tagged and manually annotated with conceptual descriptions.

Examples of syntactic heuristic rules:

1) if a preposition is a delimiter between two semantic chunks and the preposition relates the noun to its modifier, then we can combine the conceptual descriptions of the two chunks:

```
if (⟨Chunk1⟩ ⟨Border⟩ ⟨Chunk2⟩)
and (Noun in Chunk1)
and (Modifier in Chunk2)
then new concept(and sem(Chunk1)(SOME relation
sem(Chunk2)))
```

where **relation** is the most general role in the role hierarchy. We use this **relation** for combining the chunks.

2) if a conjunction relates two semantic chunks, then we combine the two associated descriptions.

```
if (⟨Chunk1⟩ ⟨conjunction⟩ ⟨Chunk2⟩)
then new concept(and sem(Chunk1) sem(Chunk2))
```

Rules are represented by patterns with probabilities. Each pattern represents the condition of the rule application and it has attached the action and the probability.

Each pattern is indexed by a trigger word that identifies the conditions for rule application. Prepositions, past participle verbs, are some examples of triggers of heuristic rules. Triggers are parts of the associated patterns.

The rules are represented as

(1) ($\langle triggerword\rangle$, $\langle probability\rangle$, $\langle pattern\rangle$, $\langle action\rangle$)

An example of rule (2) is:

($\langle$Border$\rangle$/Conj, 0.76, $\langle$Chunk1$\rangle$ $\langle$Border$\rangle$ $\langle$Chunk2$\rangle$, new concept(and sem(Chunk1) sem(Chunk2)))

The reason for assigning different probabilities is that some border words are more frequent than the other, and the rules triggered by frequent words will be assigned a greater priority. Prepositions relating a noun and its modifier are triggers for heuristic rules more frequent than auxiliaries.

The steps for applying a rule are:

a) identification of a trigger word;

b) selection of the highest probability pattern;

c) check the pattern in the text;

d) executing the action.

Example. For the text above, we apply rule 1 to "une disfonction du pont IVA" and we obtain a description `(AND Trouble (SOME relation Bridge))`. Rule 2 is triggered by the conjunction "mais" and the representation for the phrase "le pont coronaire droit est occlus mais implante sur une artére non sténosée": `(AND Bridge (SOME relation Occlusion)(SOME relation (AND Artery (ALL relation (NOT Stenose)))))`

We applied rule 1 once and another rule triggered by an auxiliary verb. We can specify more the conceptual description using the role hierarchy and the constraints imposed by the concept. For example, using DL hierarchies, we obtain `(AND Trouble (SOME hasPlace Bridge))`

III. **Wordcounter.** This module (developped in Java) is used to preprocess the documents when we modify at runtime the content of the domain model. The module extracts the list of most frequents words from the new document. From this list, it deletes the functional words (prepositions, conjunctions, determiners). For each content word (noun, adjective, verb), it stores the left and right contexts. Primitive concepts are instantiated by content words. The specific concepts are identified from the contexts (0-10 words) of the instances of primitive concepts.

3.2 Functionality

The NLP modules described above process user queries or documents to be included in the base.

The content of the hierarchy is extended dynamically, while the Web is modified every moment, new pages appear, others become not available. New documents are added to the base of documents. User queries are first processed like a set of keywords. The system retrieves a number of documents containing these keywords. These documents are then processed by NLP modules for refining the search results and for identifying new concepts.While a new document is indexed, the system identifies the concepts in the document and extends accordingly the domain model:

a) The document is first processed by **wordcounter**. The context of the content words provided by this module are processed by NLP modules for new concept identification.

b) Lexical information is assigned to each word by the POS tagger.

c) The sense tagger labels the syntagms and words with conceptual descriptions.

d) The semantic chunks are identified in the input text. Each chunk is assigned a conceptual description. The heuristic rules will be used for combining partial semantic descriptions.

e) we identify a set of new conceptual descriptions (as a result of heuristic rules), checked by the DL module. The new conceptual definitions are validated by the DL module and they are added to the domain hierarchy.

If we process user input, then the instances of the concepts are retrieved from the domain hierarchy. If new documents were processed, then the domain hierarchy could be extended with new concepts.

4 DL Conceptual Hierarchy

CICLOP, as DL system, provides powerful inference mechanisms for handling incomplete and semi-structured data, as well as validity tests for new inferred facts. On the other hand, IR systems handle incomplete or erroneous input data as well

as fuzzy domain knowledge. For these reasons we choose CICLOP as a domain knowledge representation formalism for our IR system.

The DL hierarchy is used for filtering the results of keyword based searching. New documents identified by keyword-based search are parsed by NLP modules, new concepts are identified in the text, they are validated by CICLOP module and then the hierarchy is updated. We try to automate the process of creating domain hierarchy, but we need a small set of primitive concepts defined by a human expert.

4.1 Initializing the ontology

The DL hierarchy of the domain has as its core a manually built initial hierarchy. The expert defines a set of representative keywords for the limited domain. A set of initial texts were selected manually from the results of keyword search, returned by a search engine.

The initial concepts are identified by a human expert in the list of repeated segments extracted from a set of initial texts. The expert defines also the relations between the concepts. The initial hierarchy was extracted from a set of 900 repeated segments. We defined a final set of 76 concepts. It is difficult to decide which concepts and the degree of specificity of the concept to be included in the hierarchy. The main criteria for inclusion in the hierarchy is frequency of repeated segments. Another criteria is induced by the subsumption relation, we keep the most general concepts in the initial hierarchy. The definitions of the concepts have been tested in CICLOP. Relation definitions and testing took a few days.

Example. From the segments "coronaire droite bien revascularisée", "coronaire droite dominante", "coronaire droite moyenne" et "coronaire droite occluse", we derive the concept **Right_coronary**. These segments had low frequency in the document (2-6 occurences).

Some examples of concepts and roles from the hierarchy:

```
(define-primitive-concept medtop)
(define-concept Symptom (AND medtop (SOME
hasIdentified Diagnostic) (SOME hasPlaced Anatomy)))
(define-concept Lesion (AND Symptom (ALL hasType
"lesion")))
(define-concept LesionATHCor (AND Lesion (SOME
hasLesType "athéromateuse")(SOME hasPlace Coronary))
```

Symptom is identified by a **Diagnostic** and an anatomic part where is located **Anatomy**. A **Lesion** is a **Symptom** of a given type and a coronary lesion **LesionATHCor** is a subtype of **Lesion** identified by the place **Coronary** and its type.

4.2 Extending the hierarchy

New documents found on the Web or prepared by a human expert, as well as user queries, contain instances of unknown concepts. The goal of the system is to acquire new concepts, and to place them into the existing domain hierarchies.

When a new document is added to the index base, it is processed by the POS module. Then the document is preprocessed by the **wordcounter** module, extracting the most frequent content words (noun, adjectives, etc.) and their contexts.

As a result of the preprocessing phrase, we obtain an ordered list of the most frequent content words. Their left and right contexts (up to 10 words) are used to derive other concepts. The content words and their contexts represent input for NLP modules. For each context, a concept description is built, it is checked if it exists in the domain hierarchy. If it does not exist, it is classified in the existing hierarchy.

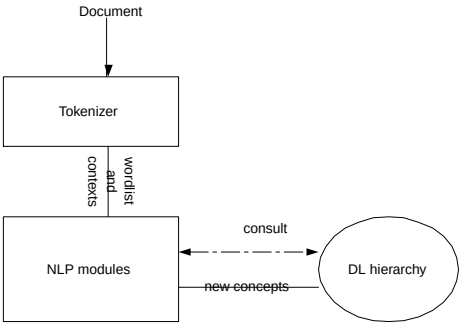

Figure 2. Document preparing for indexing

Sense tagging assigns words and syntagms with their DL descriptions. Partial semantic descriptions are combined by heuristic rules application (encoding syntactic knowledge), as explained in section 2.

Example. "infarctus" is a content word that is frequent in the heart surgery corpus. Its left and right contexts are "les patients avec" and "mais sans angioplastie":

"Les patients avec **un infarctus** mais sans angioplastie"
[The patients with a heart attack but without angioplasty.]

The system will extract the primitive concepts: **Patient**, **Heart_Attack** and **(not Angioplasty)** and it combines them obtaining more complex descriptions. We will combine **Patient** and **Heart_Attack** (while "avec un infarctus" modifies the noun phrase "les patients") and also **Patient** and **(not Angioplasty)**, using DL reasoning module (there is a role relating the two concepts). We obtain:

```
(AND Patient (SOME hasDisease Heart_Attack)(ALL
hasTreatment (not Angioplasty)))
```

In the context of Web querying, we have to limit the size of the hierarchy, in order to have acceptable answer time. The criteria of selecting the concepts is the frequency of instances in the document. Another criteria is generality. We order new concepts (with the subsumption relation) and we keep only the most general ones. A few new concepts are added for each documents (up to 10 concepts were added for the medical corpus, from small documents of 1500-2000 words).

Another problem is raised by the inconsistent conceptual descriptions. The rejected concepts have to be examined by a human expert which decides if the answer of the system was correct. If it is the case, the expert can take the decision of modifying some elements of the domain knowledge.

We need to test the prototype on real corpora and to find better criteria for limiting hierarchy size. As further work, summaries of documents will be used as index. The summaries will contain the set of most frequent general concepts from the document.

4.3 User queries

This section illustrates the use of DL concept hierarchy for handling erroneous or incomplete data.

User queries are interpreted by the NLP modules in order to extract a semantic representation. The concepts identified in the user query are used to retrieve the instances.

Example. The user asks

`"Donner le patient ayant eu un infarctus mais pas une angioplastie."`

`"Give me the patients having a heart attack but not an angioplasty".`

The sense tagging module will identify the concepts **Patient**, **Heart_attack**, **(NOT Angioplasty)**. The semantic chunks identified in this query are: "le patient", "un infarctus","pas une angioplastie". The borders phrases are : "ayant eu" et "mais".

From **Patient**, and **Heart_attack** we can create a more complex conceptual representation while in the domain model there is a role relating them (**hasDisease**) and because "un infarctus" is a modifier of "the patients". **Patient** has a **Treatment**, so we have a relation between **Patient** and **(not Angioplasty)** which is sub-concept of **Treatment**. We obtain the description:

`(AND Patient (SOME hasDisease Heart_Attack)(ALL hasTreatment (not Angioplasty)))`

The system return the set of documents containing instances of this concept.

Another reason for using DL as representation formalism of the domain knowledge and shallow NLP methods is the ability to handle incomplete or erroneous data, based on the following features:

- identification of the relevant concepts, without rigorous syntactic structures identification;
- reasoning on domain data for completing information.

Input containing errors is correctly handled by the system. Let's suppose that a syntax error occurred in the query:

"les patient eu un infarctus"

[the patient past_part a heart attack]

The semantic chunks are : "les patient" and "un infarctus". The concepts associated to these chunks are **Patient** and **Heart_Attack**. The agreement error and the missing word do not influence query understanding.

Semantic inconsistencies are detected due to DL validity check:

`"les patients avec infarctus mais sans douleurs pectorales gauches"`

`[the patients with heart attack but without pains at the left side of the breast]`

A symptom of a heart attack is the pain in the left side of the breast. The system detects the inconsistencies when combining the three semantic chunks.

The conceptual description assigned to this phrase is:

`(AND patient (SOME hasDisease (AND HeartAttack (ALL hasSymptom (NOT PainLeftBreast))))`

But the representation for the concept **HeartAttack** is:

`(define-concept HeartAttack (AND Disease (SOME hasSymptom PainLeftBreast)))`

and it is not consistent with the query.

5 Conclusion and further work

The paper presents a semantic-based approach for retrieving information from a base of documents. The system integrates shallow natural language processing for extracting the most relevant semantic chunks. It uses a domain hierarchy maintained and extended with the help of DL reasoning, as well as of shallow syntactic knowledge, used for computing semantic representation for texts and queries. The domain hierarchy is built semi-automatically, in a data-driven manner. We intend to automate the acquisition of the heuristic rules for combining the chunks by implementing a learning algorithm. Further directions of development are the use of document summaries as indexes and the integration of CICLOP commands in XML documents.

REFERENCES

[1] **F.Baader, B.Hollunder** - *A terminological Knowledge Representation systems with Complete Inference Algorithms*, Workshop on Processing Declarative Knowledge, PDK'91.

[2] **P.Buitelaar** - *CORELEX: Systematic Polysemy and Underspecification*, Ph.D. thesis, Brandeis University, Department of Computer Science, 1998

[3] **I.Horrocks** - *Optimising Tableaux Decision Procedures for Description Logics*, Ph.Thesis, University of Manchester, 1997

[4] **C.Jacquemin** - *Improving Automatic Indexing through Concept Combination and Term Enrichment*in *Proceedings of the 17th International Conference on Computational Linguistics (COLING'98)*, pages 595-599, Montréal.

[5] **P.Patel-Schneider, B.Swartout** - *Description-Logic Knowledge Representation System Specification from the KRSS Group of the ARPA Knowledge Sharing Effort* , Technical Report, DARPA Knowledge Representation System Specification (KRSS) group of the Knowledge Sharing Initiative, 1993.

[6] **C.Peltason** - *The BACK System - an overview*, SIGART Bulletin, 2(3):114-119, 1991

[7] **T.Read, E.Barcena** - *JaBot:a multilingual Java-based inteligent agent for Web sites*, in COLING'98, Montreal, Canada, 10-14 August 1998

[8] **U.Reimer, U.Hahn** - *A Formal Model of Text Summarization Based on Condensation Operators of a Terminological Logic*, in Proceedings of the Workshop on *Intelligent Scalable Text Summarization*, ed. I.Mani, M.Maybury, 11 July 1997, Madrid, pp. 97 - 104

[9] **E. Riloff, J.Lorenzen** - *Extraction-based Text Categorization Generating Domain-Specific Role Relationships Automatically*, in ed. **T.Strzalkowski**, *Natural Language Information Retrieval*, Kluwer Academic Publishers, 1999

[10] **F. de Bertrand de Beuvron, F. Rousselot , M. Grathwohl, D. Rudloff, M. Schlick** - *CICLOP*, Proceedings of the International Workshop on Description Logics - DL-99, pp 127-129, 1999.

[11] **T.Strzalkowski, J.Wang, B.Wise** - *Summarization-based Query Expansion in Information Retrieval*, in COLING'98, Montreal, Canada, 10-14 August 1998

[12] **A.Todiraşcu, F. de Beuvron, F.Rousselot** - *Using Description Logics for Document Indexing*, IAR 14th Annual Meeting, Strasbourg, 18-19 November 1999.

[13] **P.Vossen** - *EuroWordNet - A Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publishers, 1998

[14] **C. Welty, N. Ide** - Using the right tools: enhancing retrieval from marked-up documents, in J. Computers and the Humanities, Kluwer, 33(10):59-84. April, 1999.

[15] **J.Zhou** - Phrasal Terms in Real-Word IR Applications, in **T.Strzalkowski** - Natural Language Information Retrieval, Kluwer Academic Publishers, 1999