Ontologies for Information Retrieval

Amalia Todiraşcu (1) and François Rousselot (2)
(1) Faculty of Computer Science, University ”Al.I.Cuza” of Iasi, 16, Berthelot Str., Iasi 6600, Romania
Phone: +40 32 201529, Fax: +40 32 201490
E-mail: amalia@infoiasi.ro
(2) LIIA, ENSAIS, 24, Bd. de la Victoire, 67084 Strasbourg Cedex, France
Phone: +33 3 88 14 47 53, Fax: +33 3 88 24 14 90
E-mail: rousse@liia.u-strasbg.fr

Abstract

The paper presents a system for querying (in natural language) a set of text documents from a limited domain. The domain knowledge, represented in description logics (DL), is used for filtering the documents returned as answer and it 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. Concept instances are identified using shallow natural language parsing techniques.

Keywords: dynamically modified ontologies, description logics, NLP for IR systems

Résumé

L’article présente un système destiné à interroger en langue naturelle une base de texte sur un domaine limité. Les connaissances du domaine, représentées en logique de description, sont utilisées pour filtrer les documents retournés comme réponse. L’ontologie du domaine est extraite automatiquement à partir des textes et elle est modifiée dynamiquement avec des faits déduits par les mécanismes de logique de description. Les références aux concepts dans les textes sont identifiées par des techniques d’analyse du langage naturel.

1 Introduction

The increasing amount of texts available in electronic format (Web pages, CD-ROMs) from limited domains requires precise answers from Information Retrieval (IR) systems. Users have various expectations about the result provided by IR systems for various corpora: medical corpus contains information about precise diagnostic or treatment, while the news corpora contain information about persons, places, names.

The main goal of this project is to investigate several approaches of integrating semantic information into an Information Retrieval system, for querying in natural language a set of documents from a limited domain. Purely statistical search engines provide bad recall and low precision (the answers contain an important amount of irrelevant information), because they ignore hyponyms, hyperonyms and synonyms. We consider that the use of semantics is a good solution for improving IR systems performances. We integrate a dynamically-modified ontology extracted from texts into an IR system, for investigating the possible improvements of recall or precision. The information extracted from the text is used for building semi-automatically a hierarchy of domain concepts. The domain hierarchy is represented in description logic (DL), providing efficiency and fault tolerance when incomplete or erroneous data are processed. Logic inference mechanisms provided by DL reasoner are used to extend dynamically the domain model, and to complete and to correct missing information extracted from the user query. The system could be easily ported for another domain, due to the dynamic maintenance of the domain knowledge base. Other systems consulting DL ontologies used fixed, manually build domain hierarchies (Welty, Ide, 1999), and they accept searches on a few fields (author, title, book).

NLP techniques have been applied for eliminating some of the drawbacks of IR systems. NLP tools are used to extract index terms, or to develop linguistic resources dedicated to IR systems. Terms are identified by robust methods: finite state automata, POS taggers, shallow syntactic parsers (S.Ait-Mokhtar, J.-P.Chanod, 1997).

Another solution for extracting a better interpretation of the user query and documents is based on the use of semantic resources (as filters or for identifying index terms) for improving search results: multiple-word terms ((Riloff, Lorenzen, 1998), (Zhou, 1999)), their semantic variations (Jacquemin, 1998), thesauri (Corelex (Buitelaar, 1998), EuroWordnet (Vossen, 1998)) or lists of synonyms (Read, Barcena, 1998), concepts (Ambroziak, Woods, 1998). (Jacquemin, 1998) proposes a small grammar for term identification.

Existing semantic resources (WordNet, MRD, CoreLex) contain redundant information, but are often incomplete. These resources are useful for general search, on unrestricted texts, but must be reorganised to be integrated into an IR system. Searching a text base for a given domain requires domain-dependent resources. Domain-specific ontologies might help in understanding user queries and documents. Words might have specific senses, not available in general-purpose resources. For these reasons, we proposed a method for extracting the ontology from texts and several possibilities of integrating it into an IR system.

2 Description Logics

Description logics (DL) are formalisms dedicated to knowledge representation (Baader, Hollunder, 1991), more flexible than frame systems, but providing rigurous semantics and syntax. DL structures the domain knowledge on two levels: a **terminological level** (T-Box), containing the classes of domain objects (*concepts*), with their properties (*roles*) and an **assertional level**, (A-Box), containing individuals of the abstract objects (*instances*).

The terminological level provides some powerful inference tests: the satisfiability of a concept definition (there is an interpretation of the concept which is true in the set of domain facts), the detection of the subsumption relation between two concepts (detecting which concept is more general than the other one) and *classification* (ordering the new concepts in the hierarchy).

The A-Box provides *consistency* test (i.e. contradiction-free) or *instantiation* test (i.e. concept subsuming the instance) for the individual descriptions, or *retrieval inference* (retrieving for a given concept all its individuals).

Among knowledge representation formalisms, Description Logics (DL) are suitable for IR applications because they handle erroneous or incomplete data, together with the possibility of organising hierarchically the knowledge (used for). CLASSIC has been used for indexing a digital library (Welty, Ide, 1999), but it requires manual indexation and limited search on some fields. The ontology was fixed and it was built off-line. Some methods based on DL mechanisms were applied to extract terms or relations between terms (Capponi and Toussaint, 2000), but no IR system integrates a DL ontology, updated dynamically.

DLs handle semi-structured data, they do not define the exclusive list of roles for their individuals (they accept implicit definitions). DL accepts also incomplete definitions.

```
(define-concept Alpinist (AND Person (SOME hasAge Age)
(SOME hasIdeal Climbing)))

(instance y0 (AND Alpinist (SOME hasAge 30)))
```

In the example, the instance **y0** of the concept **Alpinist** defines only the role **hasAge**, while the concept has some other roles, like **hasIdeal**. The concepts are defined by **define-concept** and the logical operators AND, OR, NOT and the DL operators SOME (existential quantifier) and ALL (the universal quantifier).

A specific DL, CICLOP[1] (Rudloff et all, 1998), was chosen for representing the domain knowledge because it deals with role hierarchy, inverse roles and transitive roles. It accepts reasoning simultaneously in several hierarchies (multiple T-Boxes) and implements an A-Box. Some of the expressiveness (the features defining concepts and roles, role hierarchy, transitive roles) are necessary for representing domain knowledge into an IR system. Some tests like are useful for checking inferred facts.

3 System Architecture

The prototype integrates several natural language processing modules (Figure 1) as well as the DL classifier. We implement the prototype and we use for tests small experimental French corpora on heart surgery - MENELAS (83600 words), newspaper articles (300000 words) and NLP articles (250000 words). The system is partially implemented in Java, in Perl and in CLIPS (the rules combining domain-specific terms). The aim of these modules is to identify domain-specific terms and eventually relations between terms. From the set of terms and relations, the system updates a domain-specific hierarchy.

The automatic extraction of ontologies was the object of the TAI group (Terminology for Artificial Intelligence), defining principles for ontologies (Bouaud and all, 2000), and developping tools for identifying terms and relations. A bottom-up approach for building a hierarchy was adopted by several systems: identifying terms and assigning them to concepts (Bachimont,

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

2000), generalising predicate structures to DL concepts (Capponi and Toussaint, 2000). Our ontology definition is a simplified hierarchical model of the terminological knowledge from a given domain. Our approach starts also from texts in order to identify terms. Cue phrases (functional words, clause markers) are used to identify potential relations between terms (and also between concepts). A link between candidate terms and DL representations is proposed, and the conceptual representations are validated by DL tests.

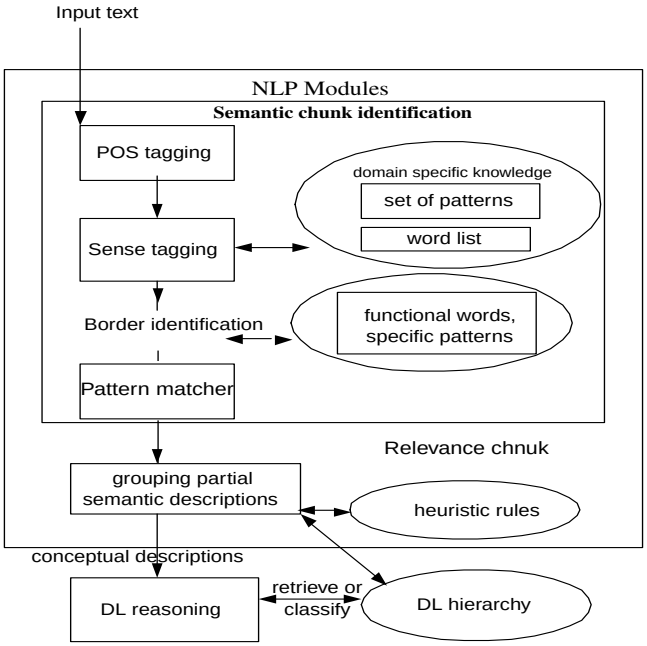

Figure 1: System architecture

We adopt a shallow, fault-tolerant approach for identifying term candidates; we need to recognize only the most relevant pieces of information in the text, even if some small errors occur in the input.

3.1 Semantic chunk 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 simple syntactic pattern (simple noun phrase, verb) and it is delimited by two border words. Border words are functional words, auxiliaries, some prepositional syntagms. The semantic chunk, as we defined in our system, might contain some errors, which are ignored by the processing modules.

Example. ”la victime, emportée par l’avalanche”

[the victim, taken from the avalanche]

In this example, ”la victime” and ”l’avalanche” are semantic chunks, containing the relevant information.

This module uses several tools: a POS tagger, a sense tagger, a border identifier and a pattern matcher. 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 Institut National pour la Langue Française (Lecomte, 1998)) 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. It was chosen because it provides good results (95 %) and it was available, with the French language model.

b) *The sense tagger* contains a pattern matcher, consulting a set of idiomatic phrases (their sense could not be composed from the component parts) and their conceptual descriptions assigned by a human expert. The sense is represented by DL conceptual descriptions. The pattern matcher annotates each known word or idiomatic phrase with its semantic description.

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). The set of cue phrases is built as a result of studies on experimental corpora. The determiners and prepositions are best candidates for chunk border.

d) *Pattern matcher.* The goal of this module is to identify the core of the semantic chunks, which is represented by simple noun phrases and verb phrases, between two consecutive borders.

Examples. A simple noun phrase is identified by the following rules: $N - \rangle NP$, $N\ \ ADJ - \rangle NP$, $N\ \ *\ \ ADJ - \rangle NP$

Errors might occur between two consecutive borders, and these cases are handled by patterns like the last one. '*' replaces anything in the input text, so errors might be ignored.

e) *DLgen* interprets the information provided by the POS tagger and generates automatically a concept definition. A human expert must check the output of this module. A few examples of rules proposed for generating simple DL descriptions:

- S1/N S2/ADJ is associated to the definition `(define-concept S1_S2 (AND S1 (SOME hasAtr "S2")))`
- S1/N S2/NNP is associated to the definition `(define-concept S1 (SOME hasName "S2"))`
- S1/ADJ S2/N is associated to the definition `(define-concept S2_S1 (AND S2 (SOME hasAtr "S1")))`
- The verbs are translated into role names: S1/VB is associated to the role **hasS1**.

There are also some simple patterns for describing phenomena like negation, in particular, even if it is impossible to enumerate all the possibilities, to detect correctly the scope of the negation:

- sans/ADV S1/N is associated to the definition `(define-concept not_S1 (NOT S1))`
- aucun/ADV S1/N is associated to the same definition `(define-concept not_S1 (NOT S1))`

Of course, not all the negations might be handled. The output of **DLgen** module provides only 61 % right annotations. Its output is checked by the human expert with the use of the DL classifier to see if the concept definitions generated automatically are correct or not, due to input errors: determiners starting the sentence, some patterns which are not coded by the system.

3.2 Relations between terms

This module applies DL inference mechanisms, as well as syntax-based rules, in order to combine the conceptual descriptions associated to each semantic chunk.

1) *Relevance chunker.* We consider two classes of chunks: **main** chunks and **secondary** chunks. Main chunks are similar to the notion of heads proposed by classical linguistic theories. Secondary chunks play the role of a modifier, which just add more information to the sense of the head, which might be absent. We interpreted the order and the position of the chunks in the sentence, used for combining concepts. Some examples of rules defines various chunks:

- semantic chunks following after a gerund verb, an auxiliary plus a past participle verb or a preposition are *secondary chunks*;
- verbs are always *Main* chunks;
- chunks following after a conjunction are annotated by the same tag as the previous one.

Example.

'[Main Les contrebandiers Main] [Main ont commencé Main] [Second á utiliser ces " monstres " Second]'

'The trafficants have started to use monsters'

The main chunks identified here are "les contrebandiers" (the first chunk in the sentence), and the "ont commencé" (comme verb principal). The last chunk is marked as secondary because it follows after the preposition "à".

2) *Heuristic rules.* The rules are established by a human expert as a result of corpora studies. The corpora were POS tagged and manually annotated with conceptual descriptions. The set of heuristic rules is established after a list of patterns extracted from corpora.

Example of syntactic heuristic rules: if a preposition is a delimiter between two semantic chunks and the preposition is relating a noun to its modifier, then we can combine the conceptual descriptions of the two chunks into a more complex semantic description.

```
if (⟨MainChunk1⟩ ⟨Border⟩ ⟨SecChunk2⟩)
and (Noun in MainChunk1)
and (Modifier in SecChunk2)
then combine(sem(MainChunk1), sem(SecChunk2))
```

Each pattern is an activation condition for the rule. Prepositions, past participle verbs, are some examples of border words, included in the activation conditions of the heuristic rules. Patterns associated with a given trigger word has associated a weight, extracted from corpora. A number of 43 rules has been described in CLIPS. Some rules (containing prepositions or gerunds) are more frequent than those triggered by conjunctions, so various saliences are assigned to these rules.

3) *CICLOP classifier.* We use the expressiveness and the logical inferences for checking the validity of the new inferred facts. The output of heuristic rules is refined and classified and

invalid concepts are rejected by the system. The heuristic rules associate to each trigger word or cue phrase a generic relation between the two concepts. Specifying this relation when the rules are defined means to have a set of domain-dependent rules. CICLOP inference mechanisms are used to refine the role definition, checking the concepts and their subsumees related by the roles.

Among the results provided by the heuristics rules, only 32 % are accepted by the DL classifier, because some of the inferred concepts are not consistent with the existing definitions, or because the DL hierarchy is still incomplete. The resulting concepts are validated by the existing knowledge only if the ontology is complete.

3.3 Functionality

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

The document is first processed by a tokenizer, identifying the list of the most frequent words. For each word from the list, we extract the left and the right context (a limited number of words - 5 to 10). From this list, we extract the most frequent content words (nouns, adjectives, and verbs). The contexts of these content words are processed by the NLP modules for new concept identification. The DL module validates the new conceptual definitions and they are added to the domain hierarchy.

First, the pattern matcher identifies the specific phrases in the input text, and the sense tagger labels the known words and the identified phrases with the conceptual descriptions. Lexical information is assigned to each word by the POS tagger.

Next, the semantic chunks are identified in the input text. For each chunk, we have a conceptual description. The heuristic rules will be used for combining partial semantic descriptions. We identify a set of new conceptual descriptions (as a result of heuristic rules), checked by the DL classifier. If we are processing user input, then the instances of the concepts are retrieved. If new documents were processed to be added to the text base, the domain hierarchy is extended with new concepts.

4 DL for Indexing

DL 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, DL is chosen as a domain knowledge representation formalism for our IR system. The indexing method is a Latent Semantic Indexing one (Dumais, 1993), where terms have been replaced by concepts. The matrix of documents and concepts is reduced computing subsumption relations between concepts and the concept weights (local or global frequencies).

Corpus	Concepts	Repeated segments
Menelas	137	748
News	98	350

Table 1: Various hierarchy size

4.1 Building the DL hierarchy

We try to automate as much as possible the process of creating domain hierarchy, but we need a small set of primitive concepts to start extending it.

4.1.1 Manual building

The DL hierarchy of the domain has as its core a manually built initial hierarchy. The initial concepts are identified by a human expert in the list of repeated segments extracted from a set of initial texts. The initial texts were selected manually from a set of results of keyword search (proposed by the experts to be representative for the domain) returned by Google. The expert defines also the relations between the concepts. The following table presents various hierarchy size for various corpus:

4.1.2 Modifying the hierarchy

When a new document is added to the index base, first it is preprocessed by a module extracting the most frequent content words (noun, adjectives, verbs) and their contexts. The NLP modules parse the contexts (0-10 words) and extract new concepts. For each concept, a DL description is built and it is added to the existing hierarchy.

Sense tagging assigns words and idiomatic phrases with their DL descriptions. Partial semantic descriptions are combined by applying rules encoding syntactic knowledge. For example, it is possible to combine conceptual descriptions for a noun and its modifier.

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**, **Infarct** and **(not Angioplasty)** and it combines them obtaining more complex descriptions. We will combine **Patient** and **Infarct** (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).

Due to DL capabilities of dealing with incomplete and semi-structured data, the new concepts identified in the text are combined and dynamically added to the existing hierarchy. If the new definitions are not consistent with the existing ones, then the older definition is replaced by a disjunction of two definitions:

```
C1 = (AND Heart_attack (ALL hasSympthom LeftPainBreast))
C2 = (AND Heart_attack (SOME hasSympthom (NOT LeftPainBreast)))
```

The two definitions are contradictory so the new concept `D = (OR C1 C2)` replaces C1.

Anyway, we have to limit the hierarchy size, applying two criteria. A natural method is to identify the most frequent concepts in the document, but they might be contained in all the documents. Another criterion is the subsumption test - in the hierarchy we keep only most specific concepts. Some general concepts are in this case eliminated from the index and hyponymy or hyperonymy.

4.2 Evaluation

To have a real image of the efficiency, we need to test the system on large corpora. The NLP tools used for identifying terms and relations between terms were evaluated for a set of 80 documents about accidents (mainly mountain accidents). Only 61% annotations provided by the semantic chunk identifier are correct. The errors are due mainly to POS tagging or to errors in the input (missing full stop, comma). The **RelevanceChunker** module provides a set of 69% of right annotations, because it proposes a set of ad-hoc rules. The result might be improved if complex rules must be described. 32 % of the resulting conceptual descriptions (provided by the heuristic rule module) are validated by the DL classifier as coherent with the other concept definitions. This modest result is due mainly to the granularity ontology (98 concepts extracted from this small set of documents). Even if the results are not very impressive, the LSI method improved with concepts provided better or similar precision and recall to the keyword-base searching, for 17 questions (from 30 tested).

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, by a bottom-up approach.

References

S.Ait-Mokhtar, J.-P.Chanod - *Incremental Finite-State Parsing*, in *Proceedings of the 5th ACL Conference on Application of Natural Language Processing*, March 1997, Washington, pp. 72-79.

J.Ambroziak, W.Woods - *Natural Language Technology in Precision Content Retrieval*, in *Proceedings of the International Conference on Natural Language Processing and Industrial Applications (NLP+IA 98)*, August 18-21, 1998, Moncton, New Brunswick, CANADA

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

B.Bachimont - *Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en ingénierie des connaissances*, in **J.Charlet, M.Zacklad, G.Kassel, D.Bourigault (eds.)** - *Ingénierie des connaissances - Evolutions récentes et nouveaux défis*, Eyrolles Publishing House, 2000, pp. 305-324.

J.Bouaud, B.Habert, A.Nazarenko, P.Zweigenbaum - *Regroupements issus de dépendances syntaxiques sur un corpus de spécialité: catégorisation et confrontation à deux conceptualisations du domaine*, in **J.Charlet, M.Zacklad, G.Kassel, D.Bourigault (eds.)** - *Ingénierie des connaissances - Evolutions récentes et nouveaux défis*, Eyrolles Publishing House, 2000, pp. 275-290.

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

N.Capponi, Y.Toussaint - *Interprétation de classes de termes par généralisation de structures prédicat-argument*, in **J.Charlet, M.Zacklad, G.Kassel, D.Bourigault (eds.)** - *Ingénierie des connaissances - Evolutions récentes et nouveaux défis*, Eyrolles Publishing House, 2000, pp. 337-356.

S.Dumais - *LSI meets TREC: A status report.* In **D.K.Harman, ed.** - *The First Text Retrieval Conference (TREC-1)*, 500-207, pp. 137-207, Gaithersburg, MD, March 1993. NIST Special publication.

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.

J.Lecomte - *Le Catégoriseur BRILL14-JL5/WINBRILL-0.3*, InaLF, InaLF/CNRS report, December 1998.

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

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

D.Rudloff, F. de Beuvron, M.Schlick - *Extending Tableaux Calculus with Limited Regular Expression for Role Path : an Application to Natural Language Processing*, DL'98, Trento, Italy, 1998

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

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.

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