

Semantic Indexing for Domain-Specific Search

Amalia Todiraşcu*,**, François de Beuvron*, François Rousset*

*LIIA, ENSAIS, 24, bd.de la Victoire, 67084 Strasbourg, France
email:{amalia, beuvron, rousse}@liia.u-strasbg.fr

**Faculty of Computer Science, University "Al.I.Cuza" of Iasi
16, Berthelot Str., Iasi 6600, Romania, email: amalia@infoiasi.ro

Abstract:The paper presents an implemented IR system, integrating semantic knowledge, for a specific domain. The paper focuses on the extraction of the knowledge from texts, using shallow natural language processing (NLP) techniques. The domain knowledge, represented in description logics (DL), is updated dynamically, as result of DL inferences.

1 Introduction

Statistical IR systems [SB98] match document and query representations as vectors of word weights. They provide bad recall (the ratio between the number of retrieved documents and the number of relevant documents) and low precision (the ratio between the number of relevant retrieved documents and the number of retrieved documents), due to natural language ambiguity (synonyms) and polymorphism (hyponyms/hyperonyms). To solve these problems, linguistically-motivated indexing methods propose, as document representations, sets of multiple-word terms [RL98], their semantic variations [Ja98] or concepts [AW98], extracted by robust NLP tools [AC97]. General-purpose thesauri (Corelex [Bu98], EuroWordnet [Vo98]) identifies hyponyms/hyperonyms or synonyms, for free texts searching, but their drawbacks are redundancy, incompleteness and low availability. Users expect precise answers when searching a limited domain; domain knowledge is required.

For these reasons, we adopted a semantic-based method for searching a set of documents from a limited domain (French texts from medicine and news). The information extracted from texts is used for building semi-automatically a domain ontology (a set of concepts and relations between concepts, formalizing domain knowledge), represented in description logic (DL) (providing fault tolerance and handling incomplete or erroneous data). The domain ontology is used to build document and query representations, to handle polymorphism or ambiguities.

2 Description Logics

Description logics (DL) are frame formalisms dedicated to knowledge representation ([BII91]). DL structures the domain knowledge on two levels: a **terminological level** (T-Box), containing abstract classes (*concepts*), with their properties (*roles*) and an **assertional level**, (A-Box), containing individuals (*instances*).

Dls provide various expressiveness: concept definitions, inverse or transitive roles, role hierarchies. For certain expressiveness, decidability algorithms implement T-Box inference mechanisms (concept satisfiability test, subsumption relation between two concepts, *classification* - ordering the concepts in the hierarchy), or A-Box tests (*consistency*, *instantiation* - i.e. concept subsuming the instance, or *retrieval inference* - retrieving for a given concept all its individuals).

DLs are suitable for IR applications because they handle erroneous or incomplete data. The concepts do not define the exclusive list of roles for their individuals. Intentional definitions are accepted.

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

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

The instance **y0** of **Alpinist** defines only the role **hasAge**, while the concept has also other roles, and also **30** is not explicitly defined as instance of **Age**.

Various DLs ontologies (built manually) have been used by IR systems: for modelling a digital library [W199] or indexing images [MSS99]. No system integrates a DL ontology updated dynamically. Hierarchical organization of the knowledge provides solutions for hyperonymy handling. The expressiveness required for IR is concept definition, inverse role and role hierarchy. CICLOP¹ [RBS98], a specific DL, provides the expressiveness required for IR. It also provides reasoning with multiple hierarchies, transitive roles and it implements an A-Box.

3 Using Ontologies for Indexing

Our IR system uses a DL ontology to check the consistency of new concepts identified in the texts and to handle hyperonymy/hyponymy cases. Concepts are used as indexes, as document and query representations. Our ontology is extracted from texts in two steps:

- term identification (instances of domain concepts);
- identification of relations between terms.

3.1 Building the DL hierarchy

Manual Building. The DL hierarchy has as its core a set of initial concepts identified by a human expert in the list of repeated segments (word sequences occurring at least twice) [FOR00] extracted from a set of representative texts. The expert defines also the relations between the concepts. The medical hierarchy contains 137 concepts, selected from 748 repeated segments. The news hierarchy contains 98 concepts.

Modifying the Hierarchy. The new concepts are extracted from input texts (documents or user queries) by the NLP modules presented below and they are

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

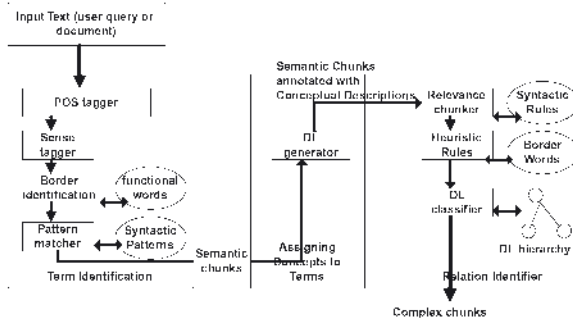


Figure 1: System architecture

combined into complex concepts. The results are checked by the DL classifier. If they are consistent, the existing definitions are updated.

Concepts are identified around the most frequent content words and their contexts. A criterion for limiting hierarchy size is to add the most frequent concepts in the document to the hierarchy. Another criterion is the subsumption test - in the hierarchy we keep only most general concepts. Specific concepts, which are important to select relevant documents, might be left out from the hierarchy, by both methods.

3.2 Tools for Term Extraction

The prototype integrates several NLP modules (fig. 1) as well as logical inference modules, for term identification and for creating complex concepts. They are used to extract a representation from queries and documents, as indexes.

1. Semantic chunks identifier detects the word sequences corresponding to the most significant domain terms (*semantic chunks*) and annotates them. A *semantic chunk* contains a simple syntactic pattern (simple noun phrase, verb) and it is delimited by two boundary words (functional words, auxiliaries, prepositional phrases) and it might contain syntactic or typing errors.

Examples of semantic chunks are "la victime" and "l'avalanche":

"la victime, emportée par l'avalanche"

[the victim, taken from the avalanche]

First, the *POS tagger* (using WinBrill, and French data provided by Institut National pour la Langue Française [Le98]) identifies the content words (nouns, adjectives, verbs) and functional words (prepositions, conjunctions etc.). The *sense tagger* assigns to some words or idiomatic phrases (their sense could not be composed from the component parts) their sense (domain concepts). The link between the concepts and the words is established by a human expert, from the list of the most frequent repeated segments extracted from texts. Then, *the boundary identifier* annotates delimiters of the semantic chunks: functional words (determiners,

prepositions), or cue phrases (syntactic phrases containing auxiliaries, composed prepositions etc.), extracted from test corpora. Then, *the pattern matcher* identifies the core of the semantic chunks (between two consecutive delimiters), which is formed by simple noun phrases (*N, N ADJ*) and verb phrases.

11. Relations between terms. This module identifies syntactic relations and creates complex concepts, validated by the DL classifier. First, *the Relevance chunker* annotates the chunks as **main** (corresponds to the notion of head proposed by classical linguistic theories) or **secondary** chunks (play the role of a modifier, adding more information to the sense of the head). In our system, the annotations corresponds to the order of the chunks in the sentences: the **secondary chunks** follow after a gerund verb or a preposition; finite verbs are always **main** chunks. This information is used by heuristic rules.

Heuristic rules were established by a human expert(43 CLIPS rules), to identify links between domain terms. The output is a set of complex concepts, to be checked by the DL reasoner. Delimiters (prepositions, past participle verbs) correspond to potential syntactic relations between the semantic chunks.

Example: if a preposition is between two semantic chunks and it relates a noun to its modifier, then we can combine the conceptual descriptions of these chunks.

```
if ((MainChunk1) {Border} {SecChunk2})
and (Noun in MainChunk1) and (Modifier in SecChunk2)
then combine(sem(MainChunk1), sem(SecChunk2))
```

A boundary is associated with several weighted rules (extracted from corpora) and only the highest weighted rule is applied.

3.3 Evaluation

We use for tests small French corpora on heart surgery (70000 words), newspaper articles (300000 words). The system is implemented in Java, in Perl and in CLIPS. The NLP tools used for identifying terms and relations between terms were evaluated for a set of 80 documents about mountain accidents (a subset of news corpus, because the domain ontology is smaller and easier to be evaluated).

The **semantic chunk identifier** provides 61% of right annotations. The errors are due mainly to POS tagging or to input errors (missing full stop, comma). The **RelevanceChunker** module provides a set of 69% of correct annotations, because it uses only some information about the term position in the sentence. This result could be improved by adding more annotation rules. **Heuristic rules** propose new combinations of primitive concepts, triggered by delimiters. Only 32% of the resulting conceptual descriptions are found consistent with the domain ontology by the DL reasoner. This modest result is due mainly to gaps of the ontology. The output of heuristic rules, depends highly on the content of the domain ontology. If there are no role paths between the concepts, the new definition is rejected. The DL reasoner identifies the correct definitions as well as wrong ones (97% of right answers). CICLOP failures are due to some constraints in defining concepts (no cyclic definitions).

4 Conclusion and Further Work

The paper presents a semantic-based approach for retrieving information from a limited domain. The system integrates shallow NLP tools for extracting the most relevant terms. It uses a domain hierarchy (dynamically maintained with the help of the DL reasoner), as well as of shallow syntactic knowledge, for building representations of texts and queries. The evaluation of the system for large corpora, the impact of granularity ontology to the precision are still open problems.

References

- [AC97] Ait-Mokhtar, S.; Chanod, J.-P.: Incremental Finite-State Parsing. In Proc. of the 5th ACL Conference on Application of Natural Language Processing, March 1997, Washington, pp. 72-79.
- [AW98] Ambroziak, J., Woods, W.: Natural Language Technology in Precision Content Retrieval. In Proc. of the Conference on Natural Language Processing and Industrial Applications, August 18-21, 1998, Moncton, Canada.
- [BH91] Baader, F., Hollunder, B.: A Terminological Knowledge Representation Systems with Complete Inference Algorithms. In Proc. of the Workshop on Processing Declarative Knowledge, PDK'91.
- [Bu98] Buitelaar, P.: CORELEX: Systematic Polysemy and Underspecification, Ph.D. thesis, Brandeis University, Dept. of Computer Science, 1998.
- [FOR00] Frath, P.; Oueslati, R.; Rousselot, P.: Identification de relations sémantiques par repérage et analyse de cooccurrences de signes linguistiques. In (Charlet, J.; Zacklad, M.; Kassel, G.; Bourigault, D. Eds.): Ingénierie des connaissances - Evolutions récentes et nouveaux défis, Eyrolles Publishing House, 2000, pp. 291-304.
- [Ja98] Jacquemin, C.: Improving Automatic Indexing through Concept Combination and Term Enrichment. In Proc. of COLING'98, pp. 595-599, Montréal.
- [Lc98] Lecomte, J.: Le Catégoriseur BRILL14-JL5/WINBRILL-0.3, InaLF, InaLF/CNRS report, December 1998.
- [MSS99] Meghini, C.; Sebastiani, F.; Straccia, U.: A System for the Fast Prototyping of Multidimensional Image Retrieval. In Proc. of ICMCS'99, Firenze, vol 11, pp. 603-609.
- [RI98] Riloff, E.; Lorenzen, J.: Extraction-based Text Categorization Generating Domain-Specific Role Relationships Automatically. In (Strzalkowski, T. ed.): Natural Language Information Retrieval, Kluwer Academic Publishers, 1999
- [RBS98] Rudloff, D.; de Beuvron, F.; Schliek, M.: Extending Tableaux Calculus with Limited Regular Expression for Role Path: an Application to Natural Language Processing. In Proc. of DL'98, Trento, Italy, 1998
- [SB98] Salton, G.; Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. In Information Processing and Management, vol.24, pp.513-523, 1998.
- [Vo98] Vossen, P.: EuroWordNet - A Multilingual Database with Lexical Semantic Networks, Kluwer Academic Publishers, 1998
- [W199] Welty, C.; Ide, N.: Using the right tools: enhancing retrieval from marked-up documents. In J. Computers and the Humanities, Kluwer, 33(10):59-84. April, 1999.