Semantic Interpretation of Terminological Strings

Roberto Navigli* and Paola Velardi*

* Department of Computer Science

University of Roma "La Sapienza"
via Salaria, 113
Roma, Italy
{navigli, velardi}@dsi.uniroma1.it

Abstract

Terminology is the surface appearance of relevant domain concepts. Though many methods have been presented to extract from texts relevant domain terminology, a semantic interpretation of these terms is still left to ontology engineers. In this paper we present a method for term extraction and semantic interpretation, based on the use of corpora and existing lexical databases, such as WordNet.

1. Introduction

Automatic extraction of terminology from corpora is a critical task for several Information Technology applications, like Document Classification and Management, Information Retrieval, etc. In particular, terminology extraction is considered a useful step for creating *Domain Ontologies*.

There has been a growing awareness on the importance of ontologies in information systems. Despite the significant amount of work carried out in recent years, ontologies are still scarcely applied and used. Research has mainly addressed the basic principles, such as knowledge representation formalisms, but limited attention has been devoted to more practical issues, such as techniques and tools aimed at the actual construction of an ontology (i.e. its actual *content*).

A key issue is the task of identifying, defining, and entering concept definitions. In case of a large and complex application domain this task can be lengthy, costly, and controversial, since different persons may have different points of view about the same concept. To reduce time, cost (and, sometimes, harsh discussions) it is highly advisable to refer, in constructing or updating an ontology, to the documents available in the field. Term extraction tools may be of great help in this task.

Though recently a number of contributors proposed methods to extract terminology and word relations from domain data and web sites (Maedche and Staab, 2000) (Morin, 1999) (Vossen, 2001), what is learned from available documents is mainly a list of *terms* and term relations. The definition (i.e. the *semantic interpretation*) of these terms is still left to the ontology engineer.

In this paper we present a system, called *OntoLearn*, aimed at supporting ontology engineers in the time-consuming task of constructing a domain ontology. The system has been designed and experimented in the context of two European projects, Fetish and Harmonise[1], where it is used as the basis of a semantic interoperability platform for small and medium-sized enterprises, operating in the tourism domain.

OntoLearn is part of an ontology engineering platform including also ontology management (Missikoff, 2000) and validation (Missikoff and Wang, 2001) tools. The scope of the paper is however restricted to the description of the terminology extraction and semantic interpretation methods.

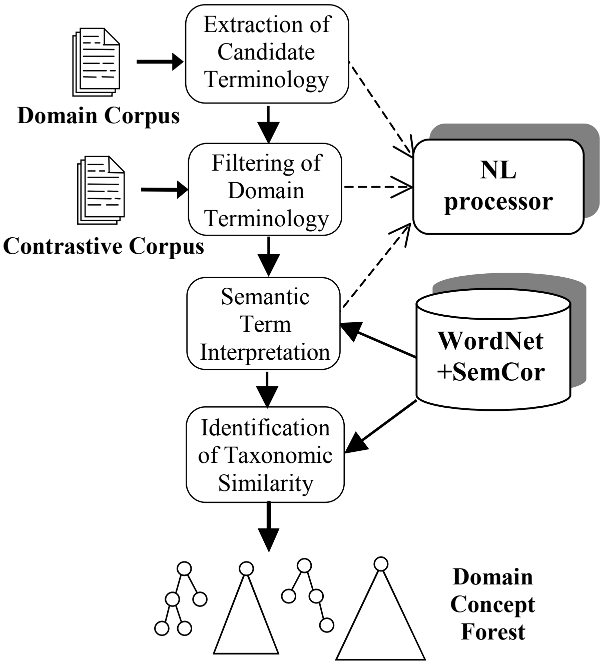
Extraction of Candidate Terminology
Domain Corpus
Filtering of Domain Terminology
Contrastive Corpus
NL processor
Semantic Term Interpretation
WordNet +SemCor
Identification of Taxonomic Similarity
Domain Concept Forest

Figure 1. Architecture of OntoLearn.

The rest of the paper is organized as follows: first, we provide a sketchy description of the system architecture. Then, we describe the term extraction methodology. We then present the semantic interpretation algorithm, and finally a performance evaluation.

[1] ITS-13015 (FETISH) and ITS-29329 (HARMONISE).

2. Architecture of OntoLearn

Figure 1 shows the architecture of the Ontolearn system. There are three main phases: first, a domain terminology is *extracted* from available texts in the application domain (specialized web sites or documents exchanged among members of a virtual community), and *filtered* using statistical techniques. Second, terms are *semantically interpreted* (in a sense that we clarify later) using WordNet 1.6 (Fellbaum, 1995) (4) and Semcor (Miller, 1993) (3), two lexical resources widely used within the computational lingustics community. Third, concepts are ordered according to *taxonomic relations*, generating a *Domain Concept Forest* (hereafter DCF).

The DCF is then validated and integrated with the domain upper ontology using the ontology validation and management tools previously referred.

2.1. Terminology extraction

Candidate terminological expressions are usually captured with more or less shallow techniques, ranging from stochastic methods (Church and Hanks, 1989) (Yamamoto and Church, 2000) to more sophisticated syntactic approaches (Jacquemin, 1997).

Obviously, richer syntactic information positively influences the quality of the result to be input to the statistical filtering. In our experiments we used the linguistic processor ARIOSTO (Basili et al., 1997) and the syntactic parser CHAOS (Basili et al., 1998). We parse the available documents in the application domain in order to extract a list T_c of syntactically *plausible* terminological candidates, e.g. compounds (*credit card*), adjective-noun (*public transport service*), prepositional phrases (*board of directors*).

OntoLearn uses a novel method for filtering "true" terminology extraction, described in detail in (Velardi et al., 2001). The method is based on two entropy related measures, called *Domain Relevance* and *Domain Consensus*, that we introduce hereafter.

High frequency in a corpus is a property observable for terminological as well as non-terminological expressions (e.g. *last week* or *real time*). We measure the specificity of a terminological candidate with respect to the target domain[2] via comparative analysis across different domains. A specific score, called *Domain Relevance* (*DR*), has been defined. A quantitative definition of the *Domain Relevance* can be given according to the amount of information captured within the target corpus wrt to the entire collection of corpora. More precisely, given a set of *n* domains { D_1, ..., D_n }, the domain relevance $DR_{t,k}$ of a term *t* in the domain D_k is computed as:

$$DR_{t,k} = \frac{P(t \mid D_k)}{\sum_{j=1}^{n} P(t \mid D_j)}$$

where $P(t \mid D_k)$ is estimated by:

$$E(P(t \mid D_k)) = \frac{f_{t,k}}{\sum_{t' \in D_k} f_{t',k}}$$

where $f_{t,k}$ is the frequency of term *t* in the domain D_k.

The second filter follows the idea that, in order for a term to be a "clue" for a domain D_k, it should appear in several documents, i.e. there must be some "consensus" on the use of that term in the domain D_k.

The *Domain Consensus* (*DC*) of term *t* in the domain D_k captures those terms that appear frequently across the documents of a given domain. *DC* is an entropy, defined as:

$$DC_{t,k} = \sum_{d \in D_k} \left(P_t(d) \log \frac{1}{P_t(d)} \right)$$

where $P_t(d)$ is the probability that document *d* includes *t*.

Terminology filtering is obtained through a linear combination of the two filters:

$$DW_{t,k} = \alpha DR_{t,k} + (1-\alpha) DC_{t,k}^{norm}$$

where $DC_{t,k}^{norm}$ is a normalized entropy and $\alpha \in (0,1)$.

2.2. Semantic interpretation of terminology

The set of validated terms are then hierarchically arranged in subtrees, according to simple string inclusion. Figure 2 is an example of what we call a lexicalized tree $\mathfrak{I}$. In absence of semantic interpretation, it is not possible to fully capture the conceptual relationships between concepts (for example, the *is-a* relation between *bus service* and *public transport service* in Figure 2).

To produce a semantic interpretation, two available lexical resources are used: WordNet and SemCor.

WordNet is a large lexical knowledge base whose popularity is recently growing even outside the computational linguistic community. In WordNet, a word sense is uniquely identified by a set of terms called *synset* (e.g., for the sense #3 of *transport*: { *transportation#4*, *shipping#1*, *transport#3* }), and a textual definition called *gloss* (e.g. "*the commercial enterprise of transporting goods and materials*"). Synsets are taxonomically structured in a lattice, with a number of "root" concepts called *unique beginners* (e.g., { *entity#1*, *something#1* }). WordNet includes over 120,000 words (and over 170,000 synsets), but very few domain terms: for example, *transport* and *company* are individually included, but not *transport company* as a unique term.

[2] With "domain" we intend a set of texts in a specific business or technical area of whatever granularity, e.g. tourism, ski packages, medicine, oncology, economy, company merges, etc.

SemCor is a corpus of semantically annotated sentences, i.e. every word is annotated with a sense tag, selected from the WordNet sense inventory for that word.

Let now $t = w_n \cdot \ldots \cdot w_2 \cdot w_1$ be a valid term belonging to a lexicalized tree $\mathfrak{I}$. The process of *semantic interpretation* is one that associates to each word w_k in t the appropriate WordNet synset S^k. The *sense* of t is hence defined as:

$$S(t) = \bigcup_{k=1}^{n} S^k,\ S^k \in Synset(w_k),\ w_k \in t.$$

Where $Synset(w_k)$ is the set of senses provided by WordNet for word w_k. For instance:

S("transport company") =
{ { transportation#4, shipping#1, transport#3 },
{ company#1 } }

corresponding to sense #1 of *company* ("*an institution created to conduct business*") and sense #3 of *transport* ("*the commercial enterprise of transporting goods and material*").

In order to disambiguate the words in a term we proceed as follows:

a) Disambiguation of the root

If t is the first analyzed element of $\mathfrak{I}$, manually disambiguate the *root* node of $\mathfrak{I}$.

b) Creation of semantic nets

For any $w_k \in t$ and any synset S_i^k of w_k (where S_i^k is the i-th sense of w_k in WordNet) create a *semantic net*. Semantic nets are automatically created using the following semantic relations: hyperonymy ($\rightarrow^{@}$), hyponymy ($\rightarrow^{\sim}$), meronymy ($\rightarrow^{\#}$), holonymy ($\rightarrow^{\%}$), pertainymy ($\rightarrow^{\backslash}$), attribute ($\rightarrow^{=}$), similarity ($\rightarrow^{\&}$), gloss ($\rightarrow^{gloss}$) and topic ($\rightarrow^{topic}$). The *gloss* and the *topic* relation are obtained parsing with ARIOSTO respectively the WordNet concept definitions and SemCor sentences including that sense. Every other relation is directly extracted from WordNet. To reduce the dimension of a SN, we consider only concepts at a distance not greater than 3 relations from S_i^k (the SN centre).

Figure 3 is an example of SN generated for sense #1 of *airplane*.

Let then $SN(S_i^k)$ be the semantic network for sense i of word w_k.

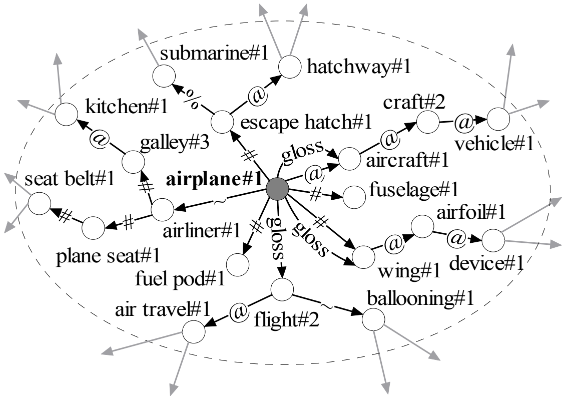

Figure 3. The semantic net for sense #1 of *airplane*.

c) Intersecting semantic nets

Starting from the "head" of t, w_1, and for any pair of words w_{k+1} and w_k (k=1,…,n) belonging to t, intersect alternative pairs of SNs. Let $I=SN(S_i^{k+1})\cap SN(S_j^k)$ be one such intersection for sense i of word w_{k+1} and sense j of word w_k. Notice that, in each step k, the word w_k is already disambiguated, either manually (for k=1) or as a result of step k-1.

d) Identifying common semantic patterns

For each alternative intersection I, identify common *semantic patterns* in order to select the sense pairs with the richest intersection.

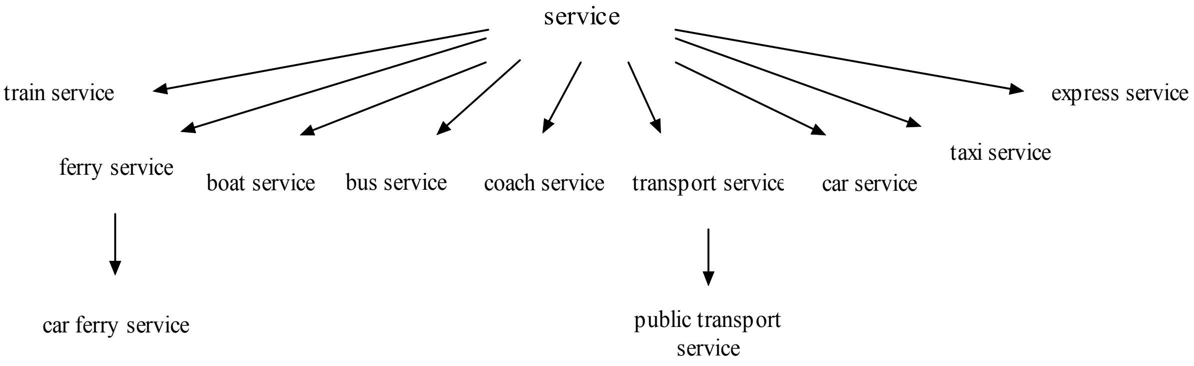

Figure 2. A lexicalized tree. The arrows simply stand for "string expansion" and provide no intended semantics.

To this end, given two arbitrary synsets S_1 and S_2, we use the following heuristics[3]:

1) *colour*, if S_1 is in the same adjectival cluster than *chromatic#3* and S_2 is a hyponym of concepts that can assume a colour like *physical object#1*, *food#1* etc. (e.g., $S_1 \equiv$*yellow#1* and $S_2 \equiv$*wall#1*);
2) *domain*, if the gloss of S_1 contains one or more domain labels and S_2 is a hyponym of those labels (for example, *white#3* is defined as "(of wine) almost colorless", therefore it is the best candidate for *wine#1* in order to disambiguate the term *white wine*);
3) *synonymy*, if

   (a) $S_1 \equiv S_2$ or (b) $\exists N \in Synset_{WN} : S_1 \overset{\backslash}{\rightarrow} N \equiv S_2$

   (for example, in the term *open air* both the words belong to synset { *open#8*, *air#2*, …, *outdoors#1* });
4) *hyperonymy/meronymy path*, if

   $$\exists M \in Synset_{WN} : S_1 \overset{@,\#}{\rightarrow^{\le 3}} M \overset{\sim,\%}{{}^{\le 3}\leftarrow} S_2$$

   (for instance, *mountain#1* $\rightarrow^{\#}$ *mountain peak#1* $\rightarrow^{@}$ *top#3* provides the right sense for each word of *mountain peak*);
5) *hyponymy/holonymy path*, if

   $$\exists M \in Synset_{WN} : S_1 \overset{\sim,\%}{\rightarrow^{\le 3}} M \overset{@,\#}{{}^{\le 3}\leftarrow} S_2$$

   (for example, in *sand beach*, *sand#1* $\rightarrow^{\%}$ *beach#1*);
6) *parallelism*, if

   $$\exists M \in Synset_{WN} : S_1 \overset{@}{\rightarrow^{\le 3}} M \overset{@}{{}^{\le 3}\leftarrow} S_2$$

   (for instance, in *enterprise company*, *organization#1* is a common ancestor of both *enterprise#2* and *company#1*);
7) *gloss*, if (a) $S_1 \overset{gloss}{\rightarrow} S_2$ or (b) $S_1 \overset{gloss}{\leftarrow} S_2$

   (for example, WordNet provides the example "a *picturesque village*" for sense 1 of *picturesque*; in *web site*, the gloss of *web#5* contains the word *site*; in *waiter service*, the gloss of *restaurant attendant#1*, hyperonym of *waiter#1*, contains the word *service*);
8) *topic*, if $S_1 \overset{topic}{\rightarrow} S_2$ (like for the term *archeological site,* where both words are tagged with sense 1 in a SemCor file; notice that WordNet provides no mutual information about them);
9) *gloss+hyperonymy/meronymy path*, if

   $$\exists G, M \in Synset_{WN} : S_1 \overset{gloss}{\rightarrow} G \overset{@,\#}{\rightarrow^{\le 3}} M \overset{\sim,\%}{{}^{\le 3}\leftarrow} S_2$$

   $$\vee S_1 \overset{gloss}{\rightarrow} G \overset{\sim,\%}{\rightarrow^{\le 3}} M \overset{@,\#}{{}^{\le 3}\leftarrow} S_2$$

   (for instance, in *railways company*, the gloss of *railway#1* contains the word *organization* and *company#1* $\rightarrow^{@}$ *institution#1* $\rightarrow^{@}$ *organization#1*);
10) *gloss+parallelism*, if

    $$\exists G, M \in Synset_{WN} : S_1 \overset{gloss}{\rightarrow} G \overset{@}{\rightarrow^{\le 3}} M \overset{@}{{}^{\le 3}\leftarrow} S_2$$

    (for instance, in *transport company*, the gloss of *transport#3* contains the word *enterprise* and *organization#1* is a common ancestor of *enterprise#2* and *company#1*);
11) *gloss+gloss*, if $\exists G \in Synset_{WN} : S_1 \overset{gloss}{\rightarrow} G \overset{gloss}{\leftarrow} S_2$

    (for example, in *mountain range*, *mountain#1* and *range#5* both contain the word *hill* so that the right senses can be chosen)
12) *hyperonymy/meronymy+gloss path*, if

    $$\exists G, M \in Synset_{WN} : S_1 \overset{@,\#}{\rightarrow^{\le 3}} M \overset{\sim,\%}{{}^{\le 3}\leftarrow} G \overset{gloss}{\leftarrow} S_2$$

    $$\vee S_1 \overset{\sim,\%}{\rightarrow^{\le 3}} M \overset{@,\#}{{}^{\le 3}\leftarrow} G \overset{gloss}{\leftarrow} S_2;$$
13) *parallelism+gloss*, if

    $$\exists G, M \in Synset_{WN} : S_1 \overset{@}{\rightarrow^{\le 3}} M \overset{@}{\leftarrow^{\le 3}} G \overset{gloss}{\leftarrow} S_2 .$$

Figure 4 is an example of one such intersection for *air-conditioned#1* and *room#1*.

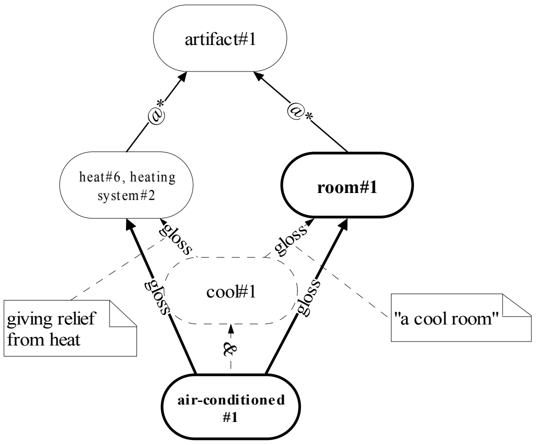

Figure 4. Intersection between the semantic nets of *air-conditioned*#1 and *room*#1 (rules (7) and (10) match).

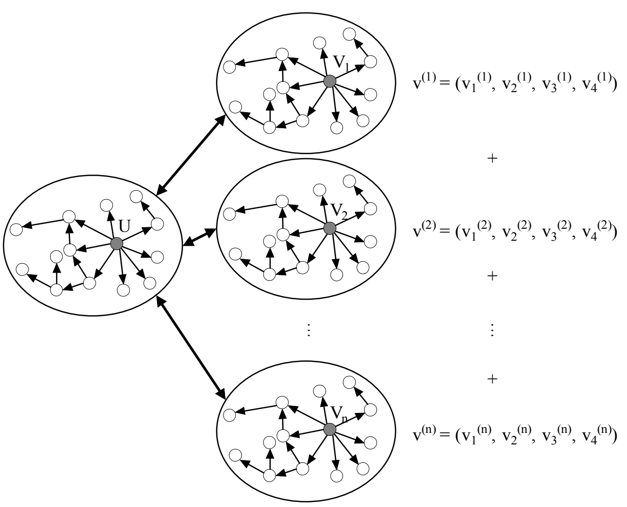
U
V1
V2
Vn
v(1) = (v1(1), v2(1), v3(1), v4(1))
+
v(2) = (v1(2), v2(2), v3(2), v4(2))
+
⋮
⋮
+
v(n) = (v1(n), v2(n), v3(n), v4(n))

Figure 5. The evaluation of a sense U of u wrt all possible senses of v (V_1, …, V_n).

In Figure 4 rules (7) and (10) are matched:

$$air-conditioned\#1 \overset{gloss}{\rightarrow} heat\#6 \overset{@^3}{\rightarrow} artifact\#1 \overset{{}^3@}{\leftarrow} room\#1$$

$$air-conditioned\#1 \overset{gloss}{\rightarrow} room\#1$$

[3] Some of these heuristics have been inspired by the work presented in (Mihalcea and Moldovan, 2001) and (Harabagiu and Moldovan, 1999)

e) Evaluate intersections

For each intersection *I*, a vector is created measuring the number and weight of matching semantic patterns, as sketched in Figure 5. That is, while disambiguating the subterm $w_{k+1}\cdot w_{k,}$ given the sense S_j^{k+1} for word w_{k+1} and all possible *n* senses of w_k, each intersection $SN(S_j^{k+1}) \cap SN(S_1^k)$, ..., $SN(S_j^{k+1}) \cap SN(S_n^k)$ is evaluated as a vector, and the sum represents the "score" vector for S_j^{k+1}. If no mutual information is retrieved (that is, the sum is $\underline{0}$), the process is repeated between w_{k+1} and w_i (i=k-1, ..., 1) until a positive score is calculated.

The best "score" vector (according to a lexicographic ordering) determines the sense for w_{k+1}. The process does not take into account the sense chosen for w_k in the previous iteration, because of a well acknowledged polysemy of words coded in WordNet (Krovetz, 1997) (in fact other senses may bring important information to the semantic interpretation process).

f) Find taxonomic relations

Initially, all the terms in a tree $\mathfrak{I}$ are independently disambiguated. In a subsequent step, the algorithm detects taxonomic relations between *concepts*, e.g. *ferry service*$\rightarrow^{@}$ *boat service*.

In this phase, since all the elements in $\mathfrak{I}$ are jointly considered, some interpretation error of the previous disambiguation step is corrected. In addition, certain concepts are *fused* in a unique "semantic domain", on the basis of pertainymy, adjectival similarity and synonymy relations (e.g. respectively: *manor house* and *manorial house*, *expert guide* and *skilled guide*, *bus service* and *coach service*). Notice again that we detect semantic relations between *concepts*, not words. For example, *bus#1* and *coach#5* are synonyms, but this relation does not hold for other senses of these two words. Each lexicalized tree $\mathfrak{I}$ is finally transformed in a *domain concept* tree Υ. Figure 6 shows the concept tree obtained from the lexicalized tree of Figure 2. Numbers for concepts are shown only when more than one semantic interpretation holds for a term, as for *coach service* and *bus service* (e.g. sense 3 of "bus" refers to "old cars").

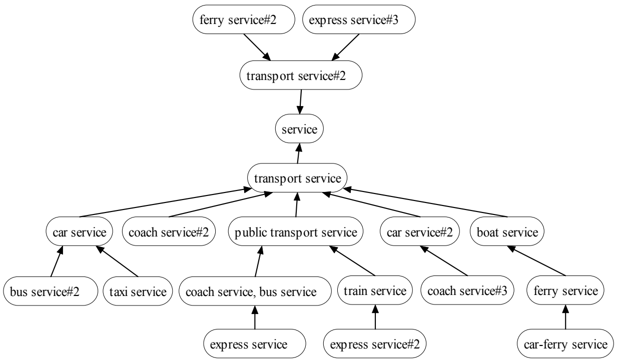

Figure 6. A domain concept tree.

3. Evaluation and Future Work

Starting from a 1 million-word corpus of travel descriptions found on web sites, a terminology of 3,840 terms was automatically extracted and manually evaluated by domain experts participating in the Harmonise project, obtaining a precision ranging from 72.9% to about 80% and a recall of 52.74%. The precision shift is due to the well-known fact that experts may have different intuitions. The recall estimate was produced by manually looking at 6,000 out of 14,383 candidate syntactic patterns (Section 2.1), marking all the terms judged as good domain concepts, and comparing the obtained list with the list of terms automatically filtered using Domain Relevance and Domain Consensus.

The authors personally evaluated the semantic disambiguation algorithm described in previous Section using a test bed of about 650 extracted terms, which have been manually assigned to the appropriate WordNet concepts. These terms contributed to the creation of 90 syntactic trees.

An extensive evaluation of the whole semantic disambiguation process led to interesting results. The analysis highlighted that some heuristics contribute more than others. In particular, rules making use of glosses ((7), (9), ..., (13)) bring precise semantic information for term disambiguation. In fact, as shown in Figure 7, while the inclusion of those heuristics gives a precision of 84.56%, their exclusion decreases precision close to the "first sense heuristic" threshold (about 79%). The precision grows to about 89% for highly structured sub-trees, as those in Figure 6, showing the importance of using rich resources like WordNet.

Variations on the structure of semantic nets have also been considered, both including the information conveyed by certain kinds of relations (pertainymy, attribute, similarity) and applying some cuts on the quantity of hyponyms and on the higher part of WordNet's name hierarchy. The best result was reached when including all kinds of semantic relations and applying reasonable cuts.

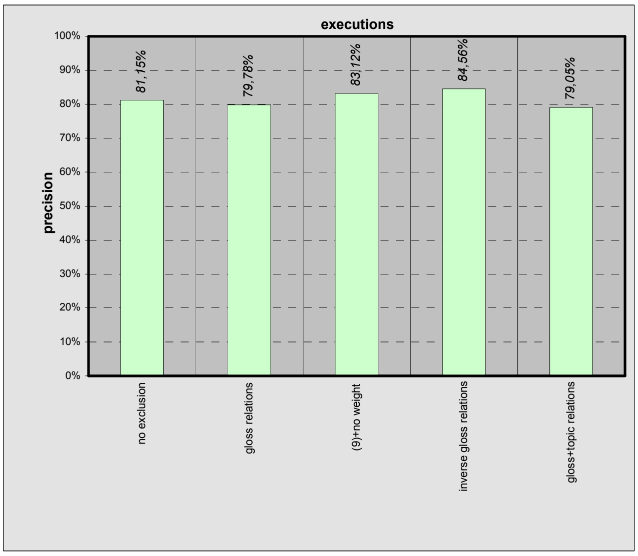

Figure 7. Precision obtained excluding some of the heuristics discussed in Section 2.2.

The obvious drawback of our algorithm is the need for manual disambiguation of the “root” node of a syntactic tree. This is a very delicate matter because choosing the wrong sense, that is the wrong collocation for the root in the hierarchy, would also affect all its descendants. However, as long as an automatic root disambiguation procedure shows a good precision, domain experts can check the results in order to make the necessary adjustments.

We are currently experimenting an algorithm for automatic disambiguation that uses as context of the root node the set of terms in its syntactic tree and all other roots in the Domain Concept Forest. Preliminary results with different settings of the algorithm give us performances ranging from 70% to 83% precision, although more experiments are needed to assess these results.

A second on-going research objective is to extend the scope of semantic interpretation beyond representing taxonomic relations. We are experimenting an inductive learning program to determine the semantic relation between concepts in a term. For example[4], "airport transfer" produces the interpretation: [transfer,transference]→(location)→[airport,aereodrome, airdrome].

4. References

Basili R., Pazienza M.T. and Velardi P., 1996. *An Empirical Symbolic Approach to Natural Language Processing*, Artificial Intelligence, n. 85.

Basili R., M.T. Pazienza, and F.M. Zanzotto, 1998. *A Robust Parser for Information Extraction*, Proceedings of the European Conference on Artificial Intelligence (ECAI '98), Brighton (UK), August 1998.

Church, K., and Hanks, P., 1989. *Word Association Norms, Mutual Information and Lexicography*. In Association for Computational Linguistics, Vancouver, Canada.

Fellbaum, C., 1995. *WordNet: an electronic lexical database*. Cambridge, MIT press.

Harabagiu, S., and Moldovan D., 1999. *Enriching the WordNet Taxonomy with Contextual Knowledge Acquired from Text*. AAAI/MIT Press.

Jacquemin, C., 1997. *Variation terminologique*. Memoire d'Habilitation Diriger des Recherces and Informatique Fondamentale. Université de Nantes, Nantes, France.

Krovetz, R., 1997. *Homonymy and polysemy in Information Retrieval*. In Proceedings of ACL/EACL'97.

Maedche A. and Staab S., 2000. *Semi-automatic Engineering of Ontologies from Text*. Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering.

Milhalcea, R. and Moldovan. D., 2001. *eXtended WordNet: progress report*. NAACL 2001 Workshop on WordNet and Other Lexical Resources, Pittsbourgh, June 2001.

Miller, G.A., Leacock, C., Tengi, R., and Bunker R. T., 1993. *A Semantic Concordance*. Proceedings of the ARPA WorkShop on Human Language Technology. San Francisco, Morgan Kaufman.

Missikoff, M. and Wang, X.F., 2001. *Consys - A Group Decision-Making Support System For Collaborative Ontology Building*, in Proc. of Group Decision & Negotiation 2001 Conference, La Rochelle, France.

Missikoff M., 2000. *OPAL - A Knolwedge-Based Approach for the Analysis of Complex Business Systems*, LEKS, IASI-CNR, Rome.

Morin E., 1999. *Automatic Acquisition of semantic relations between terms from technical corpora*, Proc. of 5th International Congress on Terminology and Knowledge extraction,TKE-99.

Velardi P., Missikoff M. and Basili R., 2001. *Identification of relevant terms to support the construction of Domain Ontologies*. ACL-EACL Workshop on Human Language Technologies, Toulouse, France, July 2001.

Vossen P., 2001. *Extending, Trimming and Fusing WordNet for technical Documents*, NAACL 2001 workshop on WordNet and Other Lexical Resources, Pittsbourgh, July 2001.

Yamamoto M. and K. Church, 2000. *Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus*, Computational Linguistics.

Web Sites

1. Fetish EC project ITS-13015 http://fetish.singladura.com/index.php

2. Harmonise EC project IST-2000-29329 http://dbs.cordis.lu

3. SemCor, *The semantic concordance corpus* http://mind.princeton.edu/wordnet/doc/man/semcor.htm

4. WordNet 1.6 http://www.cogsci.princeton.edu/~wn/w3wn.html

[4] We use Sowa's conceptual graph formalism and conceptual relation catalogue