Journal of Machine Learning Research 4 (2003) 493-525 Submitted 5/01; Published 8/03

Learning Semantic Lexicons from a Part-of-Speech and Semantically Tagged Corpus Using Inductive Logic Programming

Vincent Claveau VINCENT.CLAVEAU@IRISA.FR
Pascale Sébillot PASCALE.SEBILLOT@IRISA.FR
IRISA
Campus de Beaulieu
35042 Rennes cedex, France

Cécile Fabre CÉCILE.FABRE@UNIV-TLSE2.FR
ERSS
University of Toulouse II
5 allées A. Machado
31058 Toulouse cedex, France

Pierrette Bouillon PIERRETTE.BOUILLON@ISSCO.UNIGE.CH
TIM/ISSCO - ETI
University of Geneva
40 Bvd du Pont-d'Arve
CH-1205 Geneva, Switzerland

Editors: James Cussens and Alan M. Frisch

Abstract

This paper describes an inductive logic programming learning method designed to acquire from a corpus specific Noun-Verb (N-V) pairs—relevant in information retrieval applications to perform index expansion—in order to build up semantic lexicons based on Pustejovsky's generative lexicon (GL) principles (Pustejovsky, 1995). In one of the components of this lexical model, called the *qualia structure*, words are described in terms of semantic roles. For example, the *telic* role indicates the purpose or function of an item (*cut* for *knife*), the agentive role its creation mode (*build* for *house*), etc. The qualia structure of a noun is mainly made up of verbal associations, encoding relational information. The learning method enables us to automatically extract, from a morpho-syntactically and semantically tagged corpus, N-V pairs whose elements are linked by one of the semantic relations defined in the qualia structure in GL. It also infers rules explaining what in the surrounding context distinguishes such pairs from others also found in sentences of the corpus but which are not relevant. Stress is put here on the learning efficiency that is required to be able to deal with all the available contextual information, and to produce linguistically meaningful rules.

Keywords: corpus-based acquisition, lexicon learning, generative lexicon, inductive logic programming, subsumption under object identity, private properties

1. Introduction

The aim of information retrieval (IR) is to develop systems able to provide a user who questions a document database with the most relevant texts. In order to achieve this goal, a representation of the contents of the documents and/or the query is needed, and one commonly used technique is to associate those elements with a collection of some of the words that they contain, called in-

©2003 Vincent Claveau, Pascale Sébillot, Cécile Fabre and Pierrette Bouillon.

dex terms. For example, the most frequent (simple or compound) common nouns (N), verbs (V) and/or adjectives (A) can be chosen as indexing terms. See Salton (1989), Spärck Jones (1999) and Strzalkowski (1995) for other possibilities. The solutions proposed to the user are the texts whose indexes better match the query index. The quality of IR systems therefore depends highly on the indexing language that has been chosen. Their performance can be improved by offering more extended possibilities of matching between indexes. This can be achieved through index expansion, that is, the extension of index words with other words that are close to them in order to get more matching chances. Morpho-syntactic expansion is quite usual: for example, the same index words in plural and singular forms can be matched. Systems with linguistic knowledge databases at their disposal can also deal with one type of semantic similarity, usually limited to specific intra-category reformulations (especially N-to-N ones), following synonymy or hyperonymy links: for example, the index word *car* can be expanded into *vehicle*.

Here we deal with a new kind of expansion that has been proven to be particularly useful (Grefenstette, 1997; Fabre and Sébillot, 1999) for document database questioning. It concerns N-V links and aims at allowing matching between nominal and verbal formulations that are semantically close. Our objective is to permit a matching, for example, between a query index *disk store* and the text formulation *to sell disks*, related by the semantic affinity between an entity (store) and its typical function (sell). N-V index expansion however has to be controlled in order to ensure that the same concept is involved in the two formulations. We have chosen Pustejovsky's generative lexicon (GL) framework (Pustejovsky, 1995; Bouillon and Busa, 2001) to define what a relevant N-V link is, that is, an N-V pair in which the N and the V are related by a semantic link that is prominent enough to be used to expand index terms.

In the GL formalism, lexical entries consist of structured sets of predicates that define a word. In one of the components of this lexical model, called the *qualia structure*, words are described in terms of semantic roles. The *telic* role indicates the purpose or function of an item (for example, *cut* for *knife*), the agentive role its creation mode (*build* for *house*), the constitutive role its constitutive parts (*handle* for *handcup*) and the formal role its semantic category (*contain (information)* for *book*). The qualia structure of a noun is mainly made up of verbal associations, encoding relational information. Such N-V links are especially relevant for index expansion in IR systems (Fabre and Sébillot, 1999; Bouillon et al., 2000b). In what follows, we will thus consider as a relevant N-V pair a pair composed of an N and a V related by one of the four semantic relations defined in the qualia structure in GL.

However, GL is currently no more than a formalism; no generative lexicons exist that are precise enough for every domain and application (for example IR), and the manual construction cost of a lexicon based on GL principles is prohibitive. Moreover, the real N-V links that are the keypoint of the GL formalism vary from one corpus to another and cannot therefore be defined *a priori*. A way of building such lexicons—that is, such N-V pairs in which V plays one of the qualia roles of N—is required. The aim of this paper is to present a machine learning method, developed in the inductive logic programming framework, that enables us to automatically extract from a corpus N-V pairs whose elements are linked by one of the semantic relations defined in the GL qualia structure (called *qualia pairs* hereafter), and to distinguish them, in terms of surrounding categorial (Part-of-Speech, POS) and semantic context, from N-V pairs also found in sentences of the corpus but not relevant. Our method must respect two kinds of properties: firstly it must be robust, that is, it must infer rules explaining the concept of qualia pair that can be used on a corpus to automatically acquire GL semantic lexicons. Secondly it has to be efficient in producing generalizations from a large amount of possible contextual information found in very large corpora. This work has also a linguistic motivation: linguists do not currently know all the patterns that are likely to convey qualia relations in texts and cannot therefore verbalize rules that describe them; the generalizations inferred by our learning method have thus a linguistic interest. The paper will be divided into four parts. Section 2 briefly presents a little more information about GL and motivates using N-V index expansion based on this formalism in information retrieval applications. Section 3 describes the corpus that we have used in order to build and test our learning method, and the POS and semantic tagging that we have associated with its words to be able to characterize the context of N-V qualia pairs. Section 4 explains the machine learning method that we have developed and in particular raises questions of expressiveness and efficiency. Section 5 is dedicated to its theoretical and empirical validation, when applied to our technical corpus, and ends with a discussion about the linguistic relevance of the generalized clauses that we have learnt in order to explain the concept of qualia pairs.

2. The Generative Lexicon and Information Retrieval

In this section, we first describe the structure of a lexical entry in the GL formalism. We then argue for the use of N-V index expansion based on GL qualia structure in information retrieval.

2.1 Lexical Entries in the Generative Lexicon

As mentioned above, lexical entries in GL consist of structured sets of typed predicates that define a word. Lexical representations can thus be considered as reserves of types on which different interpretative strategies operate; these representations are responsible for word meaning in context. This generative theory of the lexicon includes three levels of representation for a lexical entry: the argument structure (`argstr`), the event structure (`eventstr`), and the qualia structure (`qs`) as illustrated in Figure 1 for word *W*.

$$\begin{bmatrix} \textbf{W} & \\ \text{ARGSTR} = & \begin{bmatrix} \text{ARG1} = \ldots \\ \text{D-ARG1} = \ldots \end{bmatrix} \\ \text{EVENTSTR} = & \begin{bmatrix} \text{E1} = \ldots \\ \text{E2} = \ldots \\ \text{RESTR} = \text{temporal relation between events} \\ \text{HEAD} = \text{relation of prominence} \end{bmatrix} \\ \text{QS} = & \begin{bmatrix} \text{W-lcp} \\ \text{FORMAL} = \ldots \\ \text{CONST} = \ldots \\ \text{TELIC} = \ldots \\ \text{AGENTIVE} = \ldots \end{bmatrix} \end{bmatrix}$$

Figure 1: Lexical entry in GL

All the syntactic categories receive the same levels of description. Argument and event structures contain the arguments and the events that occur in the definitions of the words. These elements can be either necessarily expressed syntactically or not—in this last case, they are called default arguments (D-ARG) or default events (D-E). The qualia structure links these arguments and events and defines the way they take part in the semantics of the word.

In the qualia structure, the four semantic roles correspond to interpreted features that form the basic vocabulary for the lexical description of a word, and determine the structure of the information associated with it (that is, its lexical conceptual paradigm (*lcp*)). Their meanings have already been given in Section 1. Figure 2 presents the lexical representation of *book* as mentioned by Pustejovsky (1995), in which the item both appears as a physical object, and as an object that contains information (denoted by *info.physobj-lcp*).

$$
\left[\begin{array}{ll}
\textbf{book} & \\
\text{ARGSTR} = & \left[\begin{array}{l}\text{ARG1} = \text{y : info}\\ \text{ARG2} = \text{x : physobj}\end{array}\right]\\
\text{EVENTSTR} = & \left[\begin{array}{l}\text{D-E1} = \text{e}_1\\ \text{D-E2} = \text{e}_2\end{array}\right]\\
\text{QS} = & \left[\begin{array}{l}\text{info.physobj-lcp}\\ \text{FORMAL} = \text{contain(x, y)}\\ \text{CONST} = \text{part-of(x.y, z : cover, pages, ...)}\\ \text{TELIC} = \text{read(e}_1\text{, w, x.y)}\\ \text{AGENTIVE} = \text{write(e}_2\text{, v, x.y)}\end{array}\right]
\end{array}\right]
$$

Figure 2: Lexical representation of *book*

This representation can be interpreted as: $\lambda x.y[book(x\ :\mathtt{physobj}.y\ :\mathtt{info}) \wedge contain(x,y) \wedge \lambda w \lambda e_1[read(e_1,w,x.y)] \wedge \exists e_2 \exists v[write(e_2,v,x.y)]]$.

A network of relations is thus defined for each noun, for example *book-write*, *book-read*, *book-contain* for *book*. These relations are not empirically defined but are linguistically motivated: they are the relations that are necessary to explain the semantic behaviour of the noun. These are the kinds of relations we want to use in information retrieval (IR) applications to expand index terms and deal with intercategorial semantic paraphrases of users' requests.

2.2 N-V Qualia Relations for Information Retrieval

Arguments for using GL theory to define N-V pairs relevant for index reformulation have already been reported by Bouillon et al. (2000b). We only point out here the main reasons for that option.

Many authors agree on the fact that index reformulation must not be limited to N-N relations. For example, Grefenstette (1997) suggests the importance of syntagmatic N-V links to explicit and disambiguate nouns contained in short requests in an IR application. One way to semantically characterize *research* is to extract verbs that co-occur with it to know what *research* can do (*research shows*, *research reveals*, etc.), or what is done for *research* (*do research*, *support research*, etc.). Our work within GL framework is a way to systemize such a proposition.

From the theoretical point of view, GL is a theory of words in context: it defines under-specified lexical representations that acquire their specifications within corpora. For example (see Figure 2), *book* in a given corpus can receive the agentive predicate *publish*, the telic predicate *teach*, etc. Those representations can be considered as a way to structure information in a corpus and, in that sense, the relations that are defined in GL are privileged information for IR. Moreover, in this perspective, GL has been preferred to existing lexical resources such as WordNet (Fellbaum, 1998) for two main reasons: the lexical relations we want to exhibit—namely N-V links—are unavailable in WordNet, which focuses on paradigmatic lexical relations; WordNet is a domain-independent, static resource, which, as such, cannot be used to describe lexical associations in specific texts, considering the great variability of semantic associations from one domain to another (Voorhees, 1994; Smeaton, 1999).

Concerning practical issues, the validity of using GL theory to define N-V couples relevant for reformulation has already been partly tested. First, Fabre (1996) has shown that N-V qualia pairs can be used to calculate the semantic representations of binominal sequences (*NN* compounds in English and *N preposition N* sequences in French), and thus offer extended possibilities for reformulations of compound index terms. Fabre and Sébillot (1999) have then used those relations in an experiment conducted on a French telematic service system. They have shown that the context of binominal sequences can be used to disambiguate nouns, provided that syntagmatic links exist or are developed within the thesaurus of the retrieval system, and that these syntagmatic relations can be used to discover semantic paraphrase links between a user's question and the texts of an indexed database. A second test has also been carried out in the documentation service of a Belgian bank (Vandenbroucke, 2000). Its documentalists traditionally use boolean questions with nominal terms. They were asked to evaluate the relevance of proposed qualia verbs associated with nouns of their questions to specify their requests or to access documents they had not thought of. Those first results were quite promising.

However, in order to be able to make the most of N-V qualia pairs and deeply evaluate their relevance for information retrieval applications, a method to automatically acquire these pairs from a corpus is necessary. Our goal is thus to learn GL-based semantic lexicons from corpora (more precisely N-V qualia pairs). Before describing the learning method we have developed to achieve this goal, we first present the corpus we have used, and the information we have associated with its words to be able to characterize the context of N-V qualia pairs.

3. The MATRA-CCR Corpus and its Tagging

In this section, the technical corpus that we have used to learn semantic lexicons based on GL principles is described. This corpus has first undergone part-of-speech (POS) tagging which aims at providing each word of the text with an unambiguous categorial tag (singular common noun, infinitive verb, etc.); categorial tagging is presented in Section 3.2. Secondly, in order to permit learning of what distinguishes qualia pairs from non-qualia ones that appear in exactly the same syntactic structures, semantic tags have been added. For example in structures like Verbinf det N1 prep N2, the pair N2 Verbinf is sometimes non-qualia (for example (*corrosion, vérifier*) (corrosion, check) in *vérifier l'absence de corrosion* (check the absence of corrosion)) but sometimes qualia (for example (*réservoir, vider*) (tank, empty) in *vider le fond du réservoir* (empty the bottom of the tank)) when N1 indicates for example a part of an object. A simple POS-tagging of those two sentences does not display any difference between them. Section 3.3 is dedicated to the description of the semantic tagging of the corpus, that is to the addition of tags unambiguously describing the semantic class of each of its words.

3.1 The MATRA-CCR Corpus

The French corpus used in this project is a 700 KBytes handbook of helicopter maintenance, provided by MATRA-CCR Aérospatiale, which contains more than 104,000 word occurrences. The MATRA-CCR corpus has some special characteristics that are especially well suited for our task: it is coherent, that is, its vocabulary and syntactic structures are homogeneous; it contains many concrete terms (*screw, door, etc.*) that are frequently used in sentences together with verbs indicating their telic (*screws must be tightened*, etc.) or agentive roles (*execute a setting*, etc.).

3.2 Part-of-Speech Tagging

This corpus has been POS-tagged with the help of annotation tools developed in the MULTEXT project (Ide and Véronis, 1994; Armstrong, 1996); sentences and words are first segmented with MTSEG; words are analyzed and lemmatized with MMORPH (Petitpierre and Russell, 1998; Bouillon et al., 1998), and finally disambiguated by the TATOO tool, a hidden Markov model tagger (Armstrong et al., 1995). Each word therefore only receives one POS tag which indicates its morpho-syntactic category (and its gender, number, etc.) with high precision: less than 2% of errors have been detected when compared to a manually tagged 4,000-word test-sample of the corpus. Those POS tags are one of the elements used by our learning method to characterize the context in which qualia pairs can be found.

3.3 Semantic Tagging

The semantic tagging is done on the already POS-tagged MATRA-CCR corpus; we therefore benefit from the disambiguation of polyfunctional words (that is, words that have different syntactic categories, such as *règle* in French which can be the indicative of the verb *to regulate* and the common noun *rule*) (Wilks and Stevenson, 1996). We have first built the semantic classification which we used as tagset for the semantic tagging. This tagging process is then carried out with the help of the same probabilistic tagger as for POS-tagging and, as shown here, the majority of the semantic ambiguities are solved.

More precisely, a lexicon containing every word (the lexicon entries) of the MATRA-CCR corpus is created; it associates with each word all its possible semantic tags. The most relevant tagset for each category must be chosen. We only describe here the semantic classification of the main POS categories of the MATRA-CCR corpus. We also give the results of its semantic tagging using the hidden Markov model tagger. A more detailed presentation can be found in (Bouillon et al., 2001).

WordNet's (Fellbaum, 1998) most generic classes have initially been selected to systematically classify the nouns. However, irrelevant classes (for our corpus) have been withdrawn and, for large classes, a more precise granularity has been chosen (for example the class *artefact* has been split into more precise categories). This has led to 33 classes, hierarchically organized as shown in Figure 3 (WordNet classes not used for tagging are in italics and semantic tags are bracketed). Only 8.7% of the entries of the common noun lexicon are ambiguous. Most of those ambiguities correspond to complementary polysemy (for example, *enfoncement* can both indicate a process (pushing in) or its result (hollow); it is therefore classified as both **pro** and **sta**).

Concerning verbs, WordNet classification was judged too specific. A minimal partition into 7 classes has been selected. Only 7 verbs (among about 570) are ambiguous. Adjectives and prepositions, etc. have also been classified and have led to the creation of lexicons in which very few entries are ambiguous.

Those various lexicons are then used to carry out the semantic tagging of the POS-tagged MATRA-CCR corpus by projecting the semantic tags on the corresponding words. Ambiguities are solved with the help of the probabilistic tagger, following principles described in (Bouillon et al., 2000a). A 6,000-word sample of the corpus has been chosen to evaluate the semantic tagging pre-

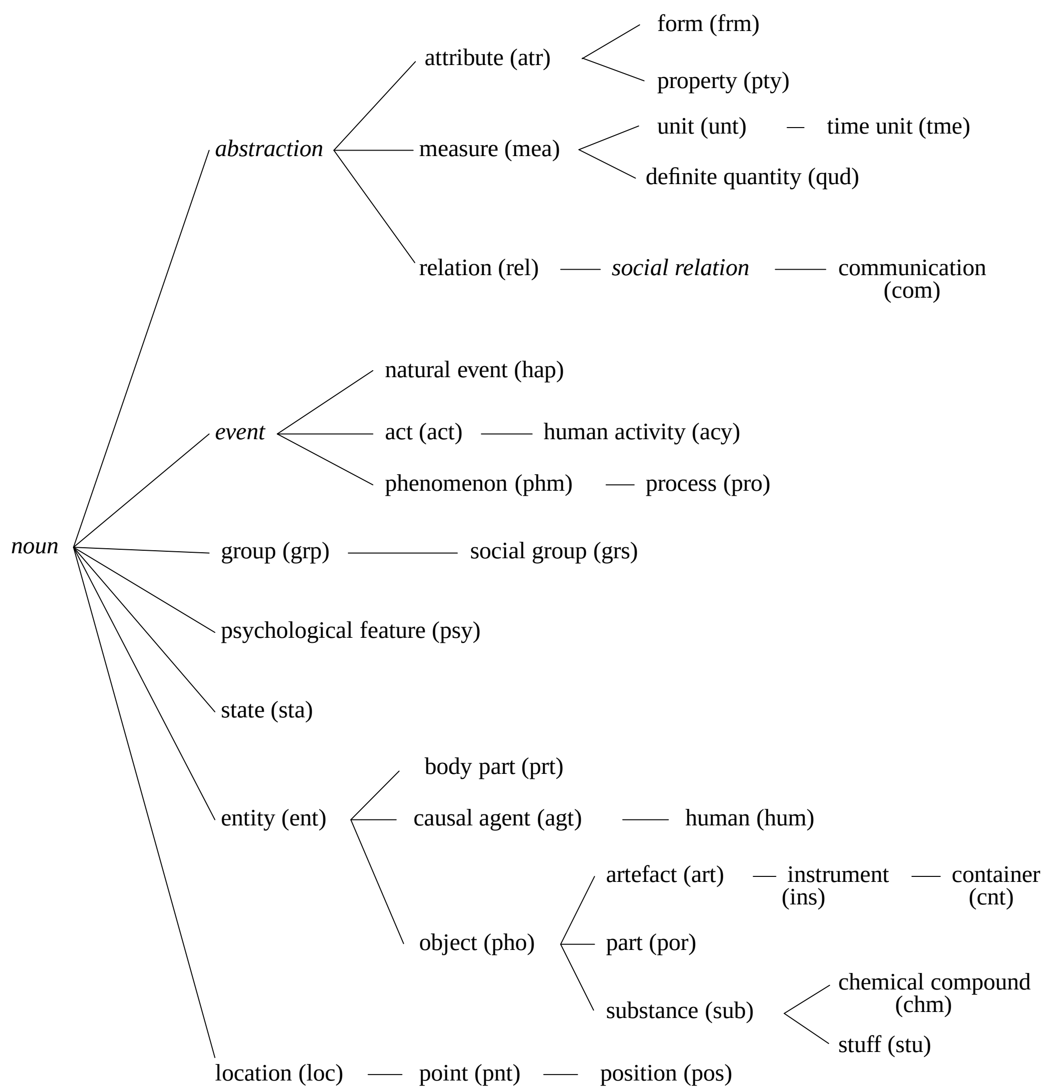

Figure 3: Hierarchy of classes for the semantic tagging of common nouns

cision. It contains 7.78% of ambiguous words; 85% of them have been correctly disambiguated (1.18% of semantic tagging errors).

All those POS and semantic tags in the MATRA-CCR corpus are the contextual key information used by the learning method that we have developed in order to automatically extract N-V qualia pairs. The next section explains its realization.

4. The Machine Learning Method

We aim at learning a special kind of semantic relations from our POS and semantically tagged MATRA-CCR corpus, that is, verbs playing a specific role in the semantic representation of common nouns, as defined in the qualia structure in GL formalism. Trying to infer lexical semantic information from corpora is not new: a lot of work has already been conducted on this subject, especially in the statistical learning domain. See Grefenstette (1994b), for example, or Habert et al. (1997) and Pichon and Sébillot (1997) for surveys of this field. Following Harris's framework (Harris et al., 1989), such research tries to extract both syntagmatic and paradigmatic information, respectively studying the words that appear in the same window-based or syntactic contexts as a considered lexical unit (first order word affinities Grefenstette, 1994a), or the words that generate the same contexts as the key word (second order word affinities). For example, Briscoe and Carroll (1997) and Faure and Nédellec (1999) try to automatically learn verbal argument structures and selectional restrictions; Agarwal (1995) and Bouaud et al. (1997) build semantic classes; Hearst (1992) and Morin (1999) focus on particular lexical relations, like hyperonymy. Some of this research is concerned with automatically obtaining more complete lexical semantic representations (Grefenstette, 1994b; Pichon and Sébillot, 2000). Among these studies, mention must be made of the research described by Pustejovsky et al. (1993) which gives some principles for acquiring GL qualia structures from a corpus; this work is however quite different from ours because it is based upon the assumption that the extraction of the qualia structure of a noun can be performed by spotting a set of syntactic structures related to qualia roles; we propose to go one step further as we have no *a priori* assumptions concerning the structures that are likely to convey these roles in a given corpus.

In order to automatically acquire N-V pairs whose elements are linked by one of the semantic relations defined in the qualia structure in GL, we have decided to use a symbolic machine learning method. Moreover, symbolic learning has led to several studies on the automatic acquisition of semantic lexical elements from corpora (Wermter et al., 1996) during the last few years. This section is devoted to the explanation of our choice and to the description of the method that we have developed.

Our selection of a learning method is guided by the fact that this method must not only provide a predictor (this N-V pair is relevant, this one is not) but also infer general rules able to explain the examples, that is, bring linguistically interpretable elements about the predicted qualia relations. This essential explanatory characteristic has motivated our choice of the inductive logic programming (ILP) framework (Muggleton and De Raedt, 1994) in which programs, that are inferred from a set of facts and a background knowledge, are logic programs, that is, sets of Horn clauses. Contrary to some statistical methods, it does not just give raw results but explains the concept that is learnt, that is, here, what characterizes a qualia pair (*versus* a non-qualia one). This choice is also especially justified by the fact that, up to now, linguists do not know what all the textual patterns that express qualia relations are; they cannot thus verbalize rules describing them. Therefore, ILP seems to be an appropriate option since its relational nature can provide a powerful expressiveness for these linguistic patterns. Moreover, as linguistic theories provide no clues concerning elements that indicate qualia relations, ILP's adaptable framework is particularly suitable for us. Lastly, the errors inherent in the automatic POS and semantic tagging process previously described make the choice of an error-tolerant learning method essential. The possibility of handling data noise in ILP guarantees this robustness.

For our experiments, we provide a set of N-V pairs related by one of the qualia relations (positive example set, E^+) within a POS and semantic context (elements from sentences containing those N-V pairs in the corpus), and a set of N-V pairs that are not semantically linked (negative example set, E^-). Generalizing rules from semantic and POS information about words that occur in the context of N-V qualia pairs in the corpus and from distances between N and V in the sentences from which examples are built is a particularly hard task. The difficulty is mainly due to the amount of information that has to be handled by the ILP algorithm. We must therefore focus on the efficiency of this learning step to be certain to obtain linguistically meaningful clauses in a relatively small amount of time. Most ILP systems provide a way to deal more or less with the problem of the form of the rules but only some of them enable a total control of this form and of the rule search efficiency. Moreover, the particular structure of our POS and semantic information makes it essential to use a system capable of processing relational background knowledge. For our project, we have thus chosen ALEPH[1], Srinivasan's ILP implementation that has already been proven well suited to deal with a large amount of data in multiple domains (mutagenesis, drug structure...) and permits complete and precise customization of all the settings of the learning task. For research use, ALEPH is also very attractive since it is entirely written in Prolog and thus allows the user to easily have a comprehensive view of the learning process, and in particular to write his/her own refinement operator to adequately perform rule search. However, this is certainly not the fastest choice: other ILP programs could be used that would perform in shorter time, but it would be to the detriment of a complete user control on the learning task. A few experiments have indeed been carried out with Quinlan's FOIL; the computing time was better (about half of the ALEPH time, see Section 5.1), but some of the produced rules did not match the linguistically motivated form requirements we defined in Section 4.2. These results are certainly due to the greedy search strategy used by FOIL.

In this section we first explain the construction of E^+ and E^- for ALEPH. We then define the space in which the rules that we want to learn are searched for (that is, what the rules we learn are and how they are related to each other). We finally describe how we improve the efficiency of the search by pruning some irrelevant hypotheses. The clauses that are obtained and their evaluation are detailed in Section 5.

4.1 Example Construction

Our first task consists in building up E^+ and E^- for ALEPH, in order for it to infer generalized clauses that explain what, in the POS and semantic contexts of N-V pairs, distinguishes relevant pairs from non-relevant ones. Here is our methodology for their construction.

First, every common noun in the MATRA-CCR corpus is considered. More precisely, we only deal with a 81,314 word occurrence subcorpus of the MATRA-CCR corpus, which is formed by all the sentences that contain at least one N and one V. This subcorpus contains 1,489 different N (29,633 noun occurrences) and 567 different V (9,522 verb occurrences). For each N, the 10 most strongly associated V, in terms of χ^2 (a statistical correlation measure based upon the relative frequencies of

1. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/aleph/aleph_toc.html

words), are selected. This first step produces at the same time pairs whose components are correctly bound by one qualia role (*(roue, gonfler)* (wheel, inflate)) and pairs that are fully irrelevant (*(roue, prescrire)* (wheel, prescribe)).

Each pair is manually annotated as relevant or irrelevant according to Pustejovsky's qualia structure principles. A Perl program is then used to find the occurrences of these N-V pairs in the sentences of the corpus.

For each occurrence of each pair that is supposed to be used to build one positive example (that is, pairs that have been globally annotated as relevant), a manual control has to be done to ensure that the N and the V really are in the expected relation within the studied sentence. After this control, a second Perl program automatically produces the positive example by adding a clause of the form is_qualia(noun_identifier,verb_identifier). to the set E^+. Information is also added to the background knowledge that describes each word of the sentence and the position in the sentence of the N-V pair. For example, for a five word long sentence whose word identifiers are w_1 ... w_5, and the N-V pair w_4-w_2, the following clauses are added:

```
tags(w_1,POS-tag,semantic-tag).
tags(w_2,POS-tag,semantic-tag).
pred(w_2,w_1).
tags(w_3,POS-tag,semantic-tag).
pred(w_3,w_2).
tags(w_4,POS-tag,semantic-tag).
pred(w_4,w_3).
tags(w_5,POS-tag,semantic-tag).
pred(w_5,w_4).
distances(w_4,w_2,distance in words,distance in verbs).
```

where pred(x,y) indicates that word y occurs just before word x in the sentence, the predicate tags/3 gives the POS and semantic tags of a word, and distances/4 specifies the number of words and the number of verbs between N and V in the sentence (a negative distance indicates that N occurs before V, a positive one indicates that V occurs before N in the studied sentence; distances are shifted by one in order to distinguish a positive null distance from a negative null one).

For example, the N-V qualia pair in boldface in the sentence "*L'installation se compose : de deux atterrisseurs* ***protégés*** *par des* ***carénages****, fixés et articulés...*" (the system is composed: of two landing devices **protected** by **streamlined bodies**, fixed and articulated...) is transformed into is_qualia(m11124_52,m11124_35). and

```
tags(m11123_3_deb,tc_vide,ts_vide).
tags(m11123_3,tc_noun_sg,ts_pro).
pred(m11123_3,m11123_3_deb).
tags(m11123_16,tc_pron,ts_ppers).
pred(m11123_16,m11123_3).
tags(m11123_19,tc_verb_sg,ts_posv).
pred(m11123_19,m11123_16).
tags(m11123_27,tc_wpunct_pf,ts_ponct).
pred(m11123_27,m11123_19).
tags(m11124_1,tc_prep,ts_rde).
pred(m11124_1,m11123_27).
tags(m11124_4,tc_num,ts_quant).
pred(m11124_4,m11124_1).
tags(m11124_9,tc_noun_pl,ts_art).
pred(m11124_9,m11124_4).
tags(m11124_35,tc_verb_adj,ts_acp).
```

```
pred(m11124_35,m11124_9).
tags(m11124_44,tc_prep,ts_rman).
pred(m11124_44,m11124_35).
tags(m11124_52,tc_noun_pl,ts_art).
pred(m11124_52,m11124_44).
tags(m11124_62,tc_wpunct,ts_virg).
pred(m11124_62,m11124_52).
tags(m11125_1,tc_verb_adj,ts_acp).
pred(m11125_1,m11124_62).
tags(m11125_7,tc_conj_coord,ts_rconj).
pred(m11125_7,m11125_1).
tags(m11125_10,tc_verb_adj,ts_acp).
pred(m11125_10,m11125_7).
...
distances(m11124_52,m11125_35,2,1).
```

where the special tags tc_vide and ts_vide describe the empty word which is used to indicate the beginning and the end of the sentence.

The negative examples are elaborated in the same way as the positive ones, with the same Perl program. They are automatically built from the above mentioned highly correlated N-V pairs that have been manually annotated as irrelevant, and from the occurrences in the corpus of potential relevant N-V pairs rejected during E^+ construction (see above). For example, the non-qualia pair in boldface in the following sentence: "*Au montage : gonfler la* ***roue*** *à la pression* ***prescrite****, ...*" (When assembling: inflate the **wheel** to the **prescribed** pressure, ...) is added to the set E^- as is_qualia(m7978_15,m7978_31). and the following clauses are stored into the background knowledge:

```
tags(m7977_1_deb,tc_vide,ts_vide).
tags(m7977_1,tc_prep,ts_ra).
pred(m7977_1,m7977_1_deb).
tags(m7977_3,tc_noun_sg,ts_acy).
pred(m7977_3,m7977_1).
tags(m7977_11,tc_wpunct_pf,ts_ponct).
pred(m7977_11,m7977_3).
tags(m7978_7,tc_verb_inf,ts_acp).
pred(m7978_7,m7977_11).
tags(m7978_15,tc_noun_sg,ts_ins).
pred(m7978_15,m7978_7).
tags(m7978_20,tc_prep,ts_ra).
pred(m7978_20,m7978_9).
tags(m7978_22,tc_noun_sg,ts_phm).
pred(m7978_22,m7978_20).
tags(m7978_31,tc_verb_adj,ts_acc).
pred(m7978_31,m7978_22).
tags(m7978_41,tc_wpunc,ts_virg).
pred(m7978_41,m7978_31).
...
distances(m7978_15,m7978_31,-3,-1).
```

During this step, as shown in the encoding of the previous positive and negative examples, some categories of words are not taken into account: the determiners, and some adjectives, which are not considered as relevant to bring up information about context of qualia or non-qualia pairs.

3,099 positive examples and about 3,176 negative ones are automatically produced this way from the MATRA-CCR corpus. ALEPH's background knowledge is also provided with other information, that describes special relationships among POS and semantic tags. Those relationships encode, for example, the fact that a tag tc_verb_pl indicates a conjugated verb in the plural (conjugated_plural), that can be considered as a conjugated verb (conjugated) or simply a verb (verb). Here is an example of those literals describing the words from a linguistic point of view:

```
verb( W ) :- conjugated( W ).
verb( W ) :- infinitive( W ).
...
conjugated( W ) :- conjugated_plural( W ).
conjugated( W ) :- conjugated_singular( W ).
conjugated_plural( W ) :- tagcat( W, tc_verb_pl ).
...
```

The background knowledge file describing all these relations and all the predicate definitions is given in appendix A. All the datasets (examples and background knowledge) used for the experiments are available from the authors on demand.

Let us define some terms we use later in this paper:

– "most general literals" are literals describing words that do not appear in the body of a clause in the background knowledge (for example, common_noun/1, verb/1). Note that every word can be described by one and only one "most general literal".

– "POS literals" (resp. "semantic literals") are literals describing the morpho-syntactic (semantic) aspect of a word (the most general literals are not considered as semantic or POS literals),

– "most general POS literals" (resp. "most general semantic literals") are POS (semantic) literals that appear in the body of most general literals (for example, infinitive/1, entity/1).

– for two literals l1 and l2 such that a rule l1:-l2. exists in the background knowledge, l1 is called the immediate generalization of l2 and l2 is the immediate specialization of l1. The immediate generalizations of literals are unique with respect to the background knowledge.

For any word W of our corpus, our background knowledge is such that all the literals describing W can be ordered in a tree whose particular structure is used in the learning process. Indeed, the root of the tree is the most general literal describing W and it has two branches, one for the POS literals and the other for the semantic literals. Any node (literal) of these two branches has only one upper node (its immediate generalization) and at most one lower (its immediate specialization if it exists). Other useful predicates are also stored in the background knowledge, for example tagcat/2 and tagsem/2, that are used as an interface between the examples and the *POS* and *semantic literals*, and the predicate suc/2 defined as suc(X,Y) :- pred(Y,X).; suc/2 is only used for reading convenience and is considered, especially in the hypothesis construction, as the equivalent of pred/2 (that is, is_qualia(A,B) :- suc(A,B). and is_qualia(A,B) :- pred(B,A). are considered as one unique hypothesis).

Given E^+, E^- and the background knowledge B, ALEPH tries to deal with that large amount of information and discover rules that explain (most of) the positive examples and reject (most of) the negative ones. To infer those rules, it uses examples to generate and test various hypotheses, and keeps those that seem relevant regarding what we want to learn. To sum up, ALEPH algorithm follows a very simple procedure that can be described in 4 steps, as stated in ALEPH's manual:

1 **select one example** to be generalized. If none exist, stop;

2 **build** $\perp$, that is, the most specific clause that explains the example;

3 **search** the space of solutions bounded below by $\perp$ for the hypothesis that maximizes a score function. This is done with the help of a *refinement operator*;

4 **remove examples** that are "covered" ("explained") by the hypothesis that has been found. Return to step 1.

The search of hypotheses (step 3) is the most complex task of this algorithm, and also the longest one. To improve the efficiency of the learning and control the expressiveness of the solutions, this search space must be characterized.

4.2 Hypothesis Search Lattice

Many machine learning tasks can be considered as a search problem. In ILP, the hypothesis H that has to be learnt must satisfy:

$\forall e^+ \in E^+ : B \cup H \models e^+$ (completeness)

$\forall e^- \in E^- : B \cup H \not\models e^-$ (correctness)

Such a hypothesis is searched for through the space of all Horn clauses to find the one that is complete and correct. Unfortunately, the tests required on the training data are costly and preclude an exhaustive search throughout the entire hypothesis space. Several kinds of biases are therefore used to limit that search space (see Nédellec et al., 1996). One of the most natural ones is the *hypothesis language bias* which defines syntactic constraints on the hypotheses to be found. This restriction on the search space considerably limits the number of potential solutions, prevents overfitting and ensures that only well-formed ones are obtained.

For us, a well-formed hypothesis is defined as a clause that gives (semantic and/or POS) information about words (N, V or words occurring in their context) and/or information about respective positions of N and V in the sentence. For example is_qualia(A,B) :- artefact(A), pred(B,C), suc(A,C), auxiliary(C).—which means that a N-V pair is qualia if N is an artefact, V is preceded by an auxiliary verb and N is followed by the same verb—is a well-formed hypothesis. We have therefore to indicate in ALEPH's settings that the predicates artefact/1, pred/2, suc/2, auxiliary/1... can be used to construct a hypothesis. Another constraint on the hypothesis language is that there can be at most one item of POS information and one item of semantic information about a given word. This means that the hypothesis is_qualia(A,B) :- pred(B,C), participle(C), past_participle(C). is not considered legal since there are two items of POS information about the word represented by C. Conversely, the hypotheses is_qualia(A,B) :- pred(B,C), participle(C), action_verb(C). or is_qualia(A,B) :- pred(B,C), past_participle(C), physical_action_verb(C). or even is_qualia(A,B) :- pred(B,C), suc(A,C). are well-formed with respect to our task. Redundant information on one word is indeed superfluous and useless since all our POS and semantic information is hierarchically organized: one of the literals is thus more specific than the others and describes the word in a precisely enough way; the other literals are therefore useless. In our example, there is no need to say that C is a participle (participle(C)) if it is known to be a past participle (past_participle(C)). This superfluousness issue is managed by our refinement operator. Several other predicates, in particular those dealing with the distances between N and V and their relative positions, are used in the hypothesis language. More than 100 different predicates can thus occur in a hypothesis.

Even with this language bias, our learning search space remains huge. Fortunately, the hypotheses can be organized by a generality relation (with the help of a quasi-order on hypotheses) which permits the algorithm to run intelligently across the space of solutions. Several quasi-orderings have been studied in the ILP framework. Logical implication would ideally be the preferred generality relation, but undecidability results lead to its rejection (Nienhuys-Cheng and de Wolf, 1996). Another order, commonly used by ILP systems, is θ-subsumption (Plotkin, 1970), defined below.

Definition 1 *A clause C_1 θ-subsumes a clause C_2 ($C_1 \succeq_\theta C_2$) if and only if (iff) there is a substitution θ such that $C_1\theta \subseteq C_2$ (considering the clauses as sets of literals).*

This order is weaker than implication ($C_1 \succeq_\theta C_2 \Rightarrow C_1 \models C_2$ but reverse is not true) but allows an easier handling of the clauses. θ-subsumption remains however too strong for our application. Indeed, let us consider $H_1 \equiv$ is_qualia(X_1,Z_1) :- suc(X_1,Y_1), pred(Z_1,W_1), verb(Y_1), verb(W_1). and $H_2 \equiv$ is_qualia(X_2,Z_2) :- suc(X_2,Y_2), pred(Z_2,Y_2), verb(Y_2).. Then, we have $H_1 \succeq_\theta H_2$ with $\theta = [X_1/X_2, Y_1/Y_2, Z_1/Z_2, W_1/Y_2]$ and since in our application, variables represent words, this means that θ-subsumption allows to consider one word as two different ones in a clause, as this is the case with the word Y_1/W_1 in H_1. This property is not considered as relevant for our learning task; we thus focus our attention on a coercive form of θ-subsumption: *θ-subsumption under object identity* (henceforth θ_{OI}-subsumption) (Esposito et al., 1996) defined below.

Definition 2 (after Badea and Stanciu (1999)) *A clause C_1 θ_{OI}-subsumes a clause C_2 ($C_1 \succeq_{OI} C_2$) iff there is a substitution θ such that $C_1\theta \subseteq C_2$ and θ is injective (that is, θ does not unify variables of C_1).*

θ_{OI}-subsumption is obviously weaker than θ-subsumption ($C_1 \succeq_{OI} C_2 \Rightarrow C_1 \succeq_\theta C_2$ but reverse is false) but preserves the expected property $H_1 \not\succeq_{OI} H_2$ (with H_1 and H_2 as defined above). This is handled in ALEPH by generating hypotheses with sets of inequalities stating that variables with two different names cannot be unified. For example, H_1 is internally represented in ALEPH by

is_qualia(X,Z):-suc(X,Y),pred(Z,W),verb(Y),verb(W),X≠Z,X≠Y,Z≠Y,X≠W,Y≠W,Z≠W.

For reading convenience, in the remaining of this paper we do not write these sets of inequalities and we assume that two differently named variables are distinct.

The notion of generality (we call it θ_{NV}-subsumption) that we use is derived from the θ_{OI}-subsumption and adapted to fit the needs of our application. Indeed, θ_{OI}-subsumption, as defined above, does not totally capture the generality notion we want to use in our hypothesis space. First, we wish to take into account the hierarchical organization of our POS and semantic information, that is, we want our generality notion to make the most of the domain theory described in the background knowledge, following ideas developed in the *generalized subsumption* framework (Buntine, 1988). For example, we want the hypothesis is_qualia(A,B) :- object(A). to be considered as more general than is_qualia(A,B) :- artefact(A). which must itself be considered as more general than is_qualia(A,B) :- instrument(A). (see Figure 3).

Moreover, we want to avoid clauses with no constraint set on a variable. For example, the hypothesis is_qualia(A,B) :- infinitive(B), pred(A,C). could simply be expressed by is_qualia(A,B) :- infinitive(B). since pred(A,C) does not bring any linguistically interesting information. However, is_qualia(A,B) :- suc(A,C), suc(C,D), object(D). is considered as well-formed since there is a semantic constraint on the word D, and C is coerced by the two suc/2. This condition is very similar to the well-known linkedness: according to Helft (1987), a clause is said to be linked if all its variables are linked; a variable V is linked in a clause C if and only if V occurs in the head of C, or there is a literal l in C that contains the variables V and W ($V \neq W$) and W is linked in C. It also corresponds to the connection constraint (Quinlan, 1990), *i1*-determinate clauses in the *ij*-determinacy framework (Muggleton and Feng, 1990) or chain-clause concept (Rieger, 1996), but in our case, every variable must not only be connected to head variables by a *path* of variables (with the help of pred/2 and suc/2), but besides, it must be "used" elsewhere in the hypothesis body. A hypothesis meeting all these conditions is said to be *well-formed* with respect to our learning task.

Therefore, we say that with respect to the background knowledge B, $C \succeq_{NV} D$ if there exist an injective substitution θ and a function f_D is such that $f_D(C)\theta \subseteq D$ ($f_D(\{l_1, l_2, ..., l_m\})$ means $\{f_D(l_1), f_D(l_2), ..., f_D(l_m)\}$) where f_D such that $\forall l \in C,\ B, f_D(l) \models l$.

Intuitively, this means that a clause D can be more specific than C if

1 – D has literals in addition to literals of C;

2 – D contains literals more specific (with respect to POS and semantic information hierarchy) on the same variables than C.

As for θ-subsumption and θ_{OI}-subsumption, θ_{NV}-subsumption induces a quasi-ordering upon the space of hypotheses with respect to our particular background knowledge and our definition of well-formed hypothesis, as stated by the three following results:

- $C \succeq_{NV} C$ (reflexivity)
- $C_1 \succeq_{NV} C_2$ and $C_2 \succeq_{NV} C_1 \Rightarrow C_1$ and C_2 are equivalent (written $C_1 \sim_{NV} C_2$); in our case (as well as for θ_{OI}-subsumption) $C_1 \sim_{NV} C_2$ means $C_1 = C_2$ up to variable renaming (antisymmetry)
- $C_1 \succeq_{NV} C_2$ and $C_2 \succeq_{NV} C_3 \Rightarrow C_1 \succeq_{NV} C_3$ (transitivity)

Proof
1 - Reflexivity: trivial.
2 - Antisymmetry: $C_1 \succeq_{NV} C_2$ and $C_2 \succeq_{NV} C_1$, thus there exist f_1, f_2, θ_1 and θ_2 such that $f_1(C_1)\theta_1 \subseteq C_2$ and $f_2(C_2)\theta_2 \subseteq C_1$, with $\forall l \in C_1, B, f_1(l) \models l$ and $\forall l \in C_2, B, f_2(l) \models l$. Therefore, $\forall l \in C_1, B, f_2(f_1(l)) \models f_1(l)$ and thus $\forall l \in C_1, B, f_2(f_1(l)) \models l$ with $f_2(f_1(l)) \in C_1$. Since C_1 is considered as well-formed and with respect to our background knowledge, we have $\forall l \in C_1, f_2(f_1(l)) = l$ and $f_1(l) = l$; similarly, $\forall l \in C_2, f_2(l) = l$. This means that $C_1\theta_1 \subseteq C_2$ and $C_2\theta_2 \subseteq C_1$ and since θ_1 and θ_2 are injective, C_1 and C_2 are only alphabetic variants.
3 - Transitivity: $C_1 \succeq_{NV} C_2$ and $C_2 \succeq_{NV} C_3$, thus there exist f_1, f_2, θ_1 and θ_2 such that $f_1(C_1)\theta_1 \subseteq C_2$ and $f_2(C_2)\theta_2 \subseteq C_3$. We have $f_2(f_1(C_1))\theta_1\theta_2 \subseteq C_3$, and $f_1 \circ f_2$ (composition of f_1 and f_2) and $\theta_1 \circ \theta_2$ are injective, therefore $C_1 \succeq_{NV} C_3$. ■

Thanks to our example representation and the background knowledge used, all the literals that can occur in hypotheses are deterministic; such hypotheses are said to be *determinate clauses*. With these linked determinate clauses, the θ_{NV}-subsumption quasi-ordering implies that the hypothesis space is structured as a lattice (detailed proof is given in appendix B for θ_{OI}-subsumption and θ_{NV}-subsumption). At the top of this lattice, we find the most general clause ($\top$) and below, a most

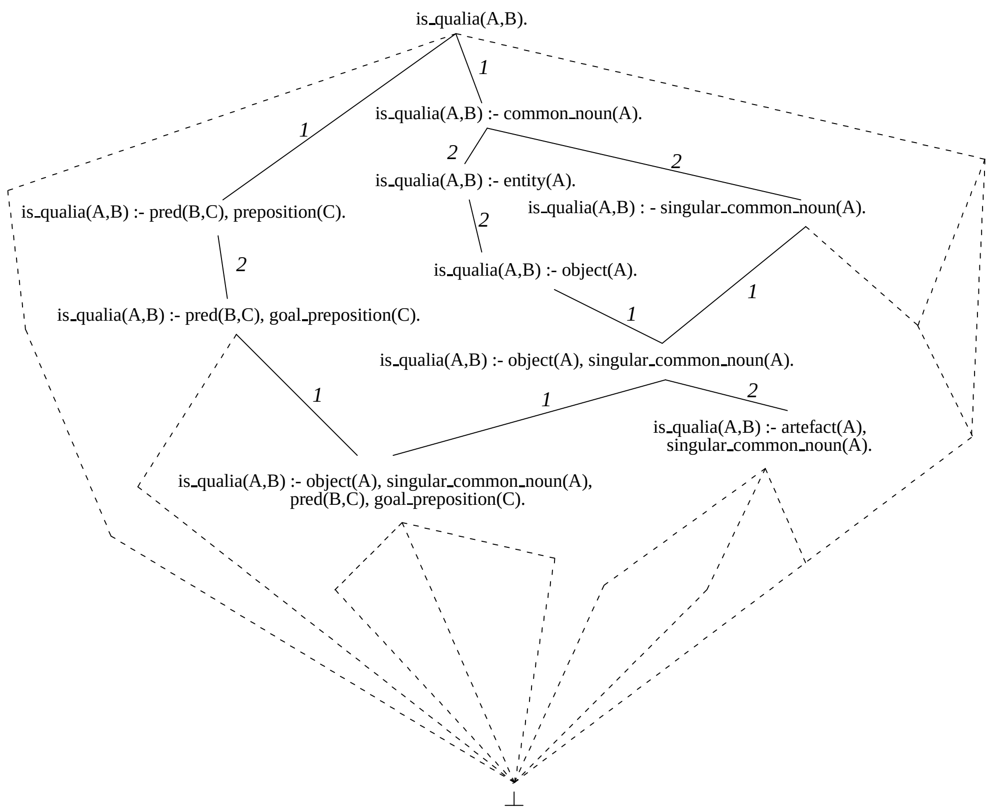

Figure 4: Hypothesis lattice for θ_{NV}-subsumption

specific clause (called MSC or bottom and henceforth written down $\perp$). In our case, $\top$ is the clause is_qualia(A,B). stating that all N-V pairs are qualia pairs, and $\perp$ is a constant-free clause containing all the literals that can be found to describe the example to be generalized (see Muggleton, 1995, for details about $\perp$ construction) minus superfluous literals (literals giving more general information about a word than other literals in $\perp$). Figure 4 shows a simple example of our lattice; numbers on the edges refer to the first or the second condition of the given definition of θ_{NV}-subsumption.

The way the search is performed in this lattice is really important to find the best hypothesis (with respect to the chosen score function) in the shortest possible time. As our background knowledge has the structure of a forest (a set of trees) and the relation introducing variables (the sequence relation pred/suc) is determinate, it is quite easy to build a perfect refinement operator (Badea and Stanciu, 1999) allowing an effective traversal of this hypothesis space ordered by θ_{NV}-subsumption using the methods described there. However, in order to save computation time, we avoid exploring parts of the hypothesis space (that is, refinements of hypotheses) if we know that there cannot be any good solution in those parts.

4.3 Pruning and Private Properties

Pruning the search is a delicate task and must be controlled so as not to "miss" a potential solution. The problem is that if a hypothesis violates some property P, one of its refinements can perhaps be correct with respect to P. Let us see how we manage pruning in our lattice with the guarantee of not leaving a valid solution out.

Some properties, called *private properties* (Torre and Rouveirol, 1997a,b,c), allow safe pruning with respect to a given refinement operator. They enable us to avoid refining a given hypothesis that does not satisfy the expected properties without taking the risk of missing a solution since no descendant of the hypothesis will satisfy those properties.

Definition 3 (from Torre and Rouveirol 1997c) *A property P is said to be private with respect to the refinement operator* ρ *into the search space S iff:*

$$\forall H, H' \in S: \ \forall(H' \in \rho^*(H) \wedge \overline{P(H)} \Rightarrow \overline{P(H')})$$

where $\overline{X}$ *indicates the negation of X and* $\forall F$*, with F a formula, denotes the universal closure of F, which is the closed formula obtained by adding a universal quantifier for every variable having a free occurrence in F.*

Let us examine a very simple and well-known private property (used as an example by Torre and Rouveirol, 1997c) that allows us to prune the search safely: the length of a clause. Formally, the property that binds the length of a clause to k literals can be expressed as $|H| \leq k$ ($|C|$ denotes the number of literals in clause C). This property is private with respect to the operator ρ in the search space S iff $\forall H, H' \in S: \forall k \in \mathbb{N}: (H' \in \rho^*_{nv}(H) \wedge |H| > k \Rightarrow |H'| > k)$. Our operator basically consists in adding literals ($H' \in \rho_{nv}(H) \wedge |H'| > |H|$) or in replacing a literal by a more specific one (then $H' \in \rho_{nv}(H) \wedge |H'| = |H|$). The clause length property is thus private with respect to ρ_{nv} and allows a safe pruning as soon as a hypothesis has too many literals.

Several other private properties are used to prune search in a safe way. We use for example the minimal number of positive examples to be covered, that is, if a clause does not explain at least a given number of positive examples this hypothesis is not considered as relevant. That property is obviously private with respect to ρ_{nv} since the numbers of covered positive and negative examples decrease through specialization.

In ILP systems, properties about the score function are often used to prune search. This function permits us to decide which hypothesis is the best one for the learning task. The one we have chosen is $s(H) = (P - N, |H|)$ where P is the number of positive examples and N the number of negative examples covered by hypothesis H. H_1 is said to be a better hypothesis than H_2 (with $s(H_1) = (P_1 - N_1, |H_1|)$ and $s(H_2) = (P_2 - N_2, |H_2|)$) iff $P_1 - N_1 > P_2 - N_2$ or $P_1 - N_1 = P_2 - N_2 \wedge |H_1| < |H_2|$. Unfortunately, since $P - N$ is not monotonic, we cannot say anything in general about the score of the refinements of a given hypothesis H that does satisfy a score criterion such that $s(H) < k$, where k can be the best score found until then in the search. This property would permit an optimal pruning, but since it is not private in our case, we cannot use it. The private property about this score function we make the most of to prune search is weaker: $s_{opt}(H) \geq S_{best}$ where S_{best} is the greatest difference $P - N$ found during the search and $s_{opt}(H) = P_{current} - N_{\perp}$. $P_{current}$ is the number of positive examples covered by the current hypothesis, $N_{\perp}$ is the number of negative examples covered by $\perp$ (evaluated at its construction time). $\forall H, H' \in S: \forall S_{best} \in \mathbb{N}: \ (H' \in \rho^*(H) \wedge s_{opt}(H) < S_{best} \Rightarrow s_{opt}(H') < S_{best})$ since P decreases through the search and $N_{\perp}$ is constant.

All those (safe) prunings ensure finding the best solution in a minimal amount of time. Two kinds of output are produced by this learning process: some clauses that have not been generalized (that is, some of the positive examples), and a set of generalized clauses, called G hereafter, and on which we shall focus our attention. Before using the clauses in G on the MATRA-CCR corpus to acquire N-V qualia pairs and automatically produce a GL-based semantic lexicon, we must validate our learning process in different ways and examine what kind of rules have been learnt.

5. Learning Validation and Results

This section is dedicated to three aspects of the validation of the machine learning method we have described. First we focus on the theoretical results of the learning, that is, we take an interest in the quality of G with respect to the training data (E^+ and E^-). The second step of the validation concerns its empirical aspect. We have applied the generalized clauses that have been inferred to the MATRA-CCR corpus and have evaluated the relevance of the decision made on the classification of N-V pairs as qualia or not. The last step concerns the linguistic relevance evaluation of the learnt rules, that is, from a linguistic point of view, what information do we learn about the semantic and syntactic context in which qualia pairs appear?

5.1 Theoretical Validation

This first point concerns the determination of a learning quality measure with the chosen parameter setting. We are particularly interested in the proportion of positive examples that are covered by the generalized clauses, and if we accept some noise in ALEPH parameter adjustment to allow more generalizations, by the proportion of negative examples that are rejected by those generalized clauses. The measure of the recall and precision rates of the learning method can be summed up using the Pearson coefficient, which is used to compare the results of different experiments:

$$\text{Pearson} = \frac{(TP*TN)-(FP*FN)}{\sqrt{PrP*PrN*AP*AN}}$$

where A = *actual*, Pr = *predicated*, P = *positive*, N= *negative*, T= *true*, F= *false*; a value close to 1 indicates a good learning.

In order to obtain good approximations of the main characteristic numbers of the learning method, we perform a 10-fold cross-validation (Kohavi, 1995) on the initial sets of 3,099 positive examples and 3,176 negative ones. Thus, the set of examples (E^+ and E^-) is divided into ten subsets, each of whose is in turn used as a testing set while the nine others are used as training set; ten learning processes are then performed with these training sets and evaluated onto the corresponding testing sets. Table 1 summarizes time,[2] precision, recall and Pearson coefficient averages and standard deviations obtained through this 10-fold cross-validation.

	Time (seconds)	Precision	Recall	Pearson
Average	10285	0.813	0.890	0.693
Standard deviation	1440	0.028	0.024	0.047

Table 1: Cross-validation results

2. Experiments were conducted on a 966MHz PC running Linux.

The entire set of examples is then used as training set by ALEPH; 9 generalized clauses (see Section 5.3) are found in less than 3 hours. We now try to estimate the performance of these rules by comparing their results on an unknown dataset with those obtained by 4 experts.

5.2 Empirical Validation

In order to evaluate the empirical validity of our learning method, we have applied the 9 generalized clauses to the MATRA-CCR corpus and have studied the appropriateness of their decisions concerning the classification of each pair as relevant or not. Since it is impossible to test all the N-V combinations found in the corpus, our evaluation has focused on 7 significant common nouns in the domain which were not used as examples, (*vis, écrou, porte, voyant, prise, capot, bouchon)* (screw, nut, door, indicator signal, plug, cowl, cap).

The evaluation has been carried out in two steps as follows. First, a Perl program retrieves all N-V pairs that appear in the same sentence in a part of the corpus and include one of the studied common nouns, and forwards them to 4 GL experts. The experts manually tag each pair as relevant or not. Divergences are discussed until complete agreement is reached.

In a second stage, this reference corpus is compared to the answers obtained for these N-V pairs of the same part in the corpus by the application of the clauses learnt with ALEPH. The results obtained for the seven selected common nouns are presented in Table 2. One N-V pair is considered as tagged *"relevant"* by the clauses if at least one of them covers this pair.

qualia pairs detected qualia	62
non-qualia pairs detected qualia	40
qualia pairs detected non-qualia	4
non-qualia pairs detected non-qualia	180
Pearson	0.666

Table 2: Empirical validation on the MATRA-CCR corpus

These results are quite promising, especially if we compare them to those obtained by χ^2 correlation (see Table 3) which was the first step of our selection of N-V couples in the corpus (see Section 4.1).

qualia pairs detected qualia	33
non-qualia pairs detected qualia	35
qualia pairs detected non-qualia	33
non-qualia pairs detected non-qualia	185
Pearson	0.337

Table 3: χ^2 results on the MATRA-CCR corpus

On one side, our ILP method detects most of the qualia N-V couples, like *porte-ouvrir* (door-open) or *voyant-signaler* (warning light-warn). The four non-detected pairs appear in very rare constructions in our corpus, like *prise-relier* (plug-connect) in *la citerne est reliée à l'appareil par des prises* (the tank is connected to the machine by plugs) where a prepositional phrase (PP) *à* *l'appareil* (to the machine) is inserted between the verb and the *par*-PP (by-PP). On the other side, only 8 pairs from the 40 non-qualia pairs detected qualia by our learning method cannot be linked syntactically. That means that the ILP algorithm can already reliably distinguish between syntactically and not syntactically linked pairs. In comparison, 25 of the 35 non-qualia pairs detected qualia by the χ^2 are not even syntactically related.

The main problem for the ILP algorithm is therefore to correctly identify N-V pairs related by a telic or agentive relation—the most common qualia links in our corpus—among the pairs that could be syntactically related. But here we should carefully distinguish two types of errors. The first ones are caused by constructions that are ambiguous and where the N-V can or cannot be syntactically related, as *enlever-prises* (remove-plugs) in *enlever les shunts sur les prises* (remove the shunts from the plugs). They cannot be disambiguated by superficial clues about the context in which the V and the N occur and show the limitation of using tagged corpus for the learning process. However they are very rare in our corpus (8 pairs). On the contrary, all remaining errors seem more related to the parameterizing of the learning method. For example, taking into consideration the number of nouns between the V and the N could avoid a lot of wrong pairs like *poser-capot* (put up-cover) in *poser les obturateurs capots* (put up cover stopcocks) or *assurer-voyant* (make sure-warning light) in *s'assurer de l'allumage du voyant* (make sure that the warning light is switched on).

The empirical validation can be therefore considered as positive and we can now focus on the last step of the evaluation that consists in assessing the linguistic validity of the generalized clauses.

5.3 Linguistic Validation

For the linguist, the issue is not only to find good examples of qualia relations but also to identify in texts the linguistic patterns that are used to express them. Consequently, the question is: what do these clauses tell us about the linguistic structures that are likely to convey qualia relations between a noun and a verb? We know from previous research (Morin, 1999) dealing with other types of semantic relations that a given relation can be instantiated by a wide variety of linguistic patterns, and that this set of structures may greatly vary from one corpus to another. Such research generally focuses on hyperonymy (is-a) and meronymy (part-of) relations, which provide the basic structure of ontologies. Our aim is thus similar, with the additional difficulty that some of the relations we focus on—such as the telic or agentive ones—have never been extensively studied on corpora, and are more difficult to identify than more conventional semantic relations. Previous research concerned with the acquisition of elements of GL (Pustejovsky et al., 1993) has looked at some solutions for identifying words linked by prespecified syntactic relations in texts, such as object relations between verbs and nouns, or certain types of N-N compounds. This research is not deeply evaluated and is however quite different from ours: contrary to this approach, we have indeed no *a priori* assumptions about the kind of structures in which telic, agentive or formal N-V pairs may be found.

We are thus faced with a set of nine clauses that we now try to interpret in terms of linguistic rules:

(1) is_qualia(A,B) :- precedes(B,A), near_verb(A,B), infinitive(B), action_verb(B).

(2) is_qualia(A,B) :- contiguous(A,B).

(3) is_qualia(A,B) :- precedes(B,A), near_word(A,B), near_verb(A,B), suc(B,C), preposition(C).

(4) is_qualia(A,B) :- near_word(A,B), pred(A,C), void(C).

(5) is_qualia(A,B) :- precedes(B,A), suc(B,C), pred(A,D), punctuation(D), singular_common_noun(A), colon(C).

(6) is_qualia(A,B) :- near_word(A,B), suc(B,C), suc(C,D), action_verb(D).

(7) is_qualia(A,B) :- precedes(A,B), near_word(A,B), pred(A,C), punctuation(C).

(8) is_qualia(A,B) :- near_verb(A,B), pred(B,C), pred(C,D), pred(D,E), preposition(E), pred(A,F), void(F).

(9) is_qualia(A,B) :- precedes(A,B), near_verb(A,B), pred(A,C), subordinating_conjunction(C).

where near_word(X,Y) means that X and Y are separated by at least one word and at most two words, and near_verb(X,Y) that there is no verb between X and Y.

What is most striking is the fact that, at this level of generalization, few linguistic features are retained. Previous learning on the same corpus with no semantic tagging using PROGOL and a poorer contextual information (Sébillot et al., 2000) had led to less generalized rules containing more linguistic elements; these rules were however less relevant for acquiring correct qualia pairs. The 9 clauses learnt here seem to provide very general indications and tell us very little about verb types (action verb is the only information we get), nouns (common noun) or prepositions that are likely to fit into such structures. But the clauses contain other information, related to several aspects of linguistic descriptions, like:

- proximity: this is a major criterion. Most clauses indicate that the noun and the verb must be either contiguous (Clause 2) or separated by at most one element (Clauses 3, 4, 6 and 7) and that no verb must appear between N and V (Clauses 1, 3, 8 and 9).

- position: Clauses 4 and 7 indicate that one of the two elements is found at the beginning of a sentence or right after a punctuation mark, whereas the relative position of N and V (precedes/2) is given in Clauses 1, 3, 5, 7 and 9.

- punctuation: punctuation marks, more specifically colons, are mentioned in Clauses 5 and 7.

- morpho-syntactic categorization: the first clause detects a very important structure in the text, corresponding to action verbs in the infinitive form.

These features shed light on linguistic patterns that are very specific to the corpus, a text falling within the instructional genre. We find in this text many examples in which a verb at the infinitive form occurs at the beginning of a proposition and is followed by a noun phrase. Such lists of instructions are very typical of the corpus:

- *débrancher la prise* (disconnect the plug)

- *enclencher le disjoncteur* (engage the circuit breaker)

- *déposer les obturateurs* (remove the stopcocks)

To further evaluate these findings, we have compared what we find by means of the learning process to linguistic observations obtained manually on the same corpus (Galy, 2000). Galy has listed a set of canonical verbal structures that convey telic information:

infinitive verb + det + noun (*visser le bouchon*) (to tighten the cap)

verb + det + noun (*ferment le circuit*) (close the circuit)

noun + past_participle (*bouchon maintenu*) (held cap)

noun + be + past_participle (*circuits sont raccordés*) (circuits are connected)

noun + verb (*un bouchon obture*) (a cap blocks up)

be + past_participle + par + det + noun (sont obturées par les bouchons) (are blocked up by caps)

The two types of results show some overlap: both experiments demonstrate the significance of infinitive structures and highlight patterns in which the verb and noun are very close to each other. Yet the results are quite different since the learning method proposes a generalization of the structures discovered by Galy. In particular, the opposition between passive and active constructions is merged in Clause 2 by the indication of mere contiguity (V can occur before or after N). Conversely, some clues, like punctuation marks and position in the sentence, have not been observed by manual analysis because they are related to levels of linguistic information that are usually neglected by linguistic observation, even if they are known to be good pattern markers (Jones, 1994).

Consequently, when we look at the results of the learning process from a linguistic point of view, it appears that the clauses give very general surface clues about the structures that are favored in the corpus for the expression of qualia relations. Yet, these clues are sufficient to give access to some corpus-specific patterns, which is a very interesting result.

6. Conclusions and Future Work

The acquisition method of N-V qualia pairs—as defined in Pustejovsky's generative lexicon formalism—that we have developed leads to very promising results. Concerning the ILP learning system itself, we have defined and made the most of a well-suited generality notion extending object identity subsumption, which has led to obtaining only well-formed hypotheses that can be linguistically interpreted. The speed of the learning step is improved by safely pruning the search of the best rules on certain conditions expressed as private properties. The rules that are learnt lead to very good results for the N-V qualia pairs acquisition task: 94% of all relevant pairs are detected for seven significant common nouns; these results have to be compared with the 50% results of χ^2. Moreover, from a practical point of view, the linguistic validation of the inferred rules confirms the ability of our method to help a linguist detect linguistic patterns dedicated to the expression of qualia roles.

One next step of our research will consist in repeating the experiment on new textual data, in order to see what types of specific structures will be detected in a less technical corpus; and we will also focus on N-N pairs, which very frequently exhibit telic relations in texts (as in: *bouchon de protection*, protective cap). Another potential avenue is to try to learn separately each qualia semantic relation (telic, agentive, formal) instead of all together as it is done up to now. Even if such a distinction is maybe not useful for an information retrieval application, it could result in linguistically interesting rules.

Other future studies should also be undertaken to improve the portability of the full method. In particular, the semantic tagging of a corpus needs an expert's supervision to build the semantic classification of all the words. Even if the determination of the relevant classes for one domain can be partly automated (see Agarwal, 1995; Grefenstette, 1994b, for example), it still remains too costly to be carried out on any new corpus. The last phase of the project will deal with the real use of the N-V (and possibly N-N) pairs obtained with the machine learning method within an information retrieval system (such as a textual search engine) and the evaluation of the improvement of its performances both from a theoretical (recall and precision rate) and empirical (with the help of real human users) point of view.

Acknowledgments

The authors wish to thank Céline Rouveirol for helpful discussions and for insightful comments on an earlier version of this paper. They would also like to thank James Cussens and the anonymous reviewers for their excellent advice.

Appendix A. Background Knowledge

Here is the listing of the background knowledge part describing the linguistic relations as used in the experiments described in Section 5.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% background knowledge

% common noun %%%%%%%%%%%%%
common_noun( W ) :- plural_common_noun( W ).
common_noun( W ) :- singular_common_noun( W ).
common_noun( W ) :- abstraction( W ).
common_noun( W ) :- event( W ).
common_noun( W ) :- group( W ).
common_noun( W ) :- psychological_feature( W ).
common_noun( W ) :- state( W ).
common_noun( W ) :- entity( W ).
common_noun( W ) :- location( W ).
plural_common_noun(W):- tagcat(W,tc_noun_pl).
singular_common_noun(W):- tagcat(W,tc_noun_sg).
abstraction( W ) :- attribute( W ).
abstraction( W ) :- measure( W ).
abstraction( W ) :- relation( W ).
event( W ) :- natural_event( W ).
event( W ) :- act(W).
event( W ) :- phenomenon(W ).
natural_event( W ) :- tagsem(W, ts_hap ).
phenomenon( W ) :- tagsem(W, ts_phm ).
phenomenon( W ) :- process( W ).
process( W ) :- tagsem(W, ts_pro ).
act( W ) :- tagsem(W, ts_act ).
act( W ) :- human_activity( W ).
human_activity( W ) :- tagsem(W, ts_acy ).
group( W ) :- tagsem(W, ts_grp ).
group( W ) :- social_group( W ).
social_group( W ) :- tagsem(W, ts_grs ).
psychological_feature( W ) :- tagsem(W, ts_psy ).
state( W ) :- tagsem(W, ts_sta ).
entity( W ) :- tagsem(W, ts_ent ).
entity( W ) :- body_part( W ).
entity( W ) :- causal_agent( W ).
entity( W ) :- object( W ).
body_part( W ) :- tagsem(W, ts_prt ).
object( W ) :- tagsem(W, ts_pho ).
object( W ) :- artefact( W ).
object( W ) :- part( W ).
object( W ) :- substance( W ).
part( W ) :- tagsem(W, ts_por ).
location( W ) :- tagsem(W, ts_loc ).
location( W ) :- point(W).
point( W ) :- tagsem(W, ts_pnt ).
```

```
point( W ) :- position( W ).
position( W ) :- tagsem(W, ts_pos ).
attribute( W ) :- tagsem(W, ts_atr ).
attribute( W ) :- form( W ).
attribute( W ) :- property( W ).
form( W ) :- tagsem(W, ts_frm ).
property( W ) :- tagsem(W, ts_pty ).
measure( W ) :- tagsem(W, ts_mea ).
measure( W ) :- definite_quantity( W ).
measure( W ) :- unit( W ).
time_unit( W ) :- tagsem(W, ts_tme ).
definite_quantity( W ) :- tagsem(W, ts_qud ).
unit( W ) :- tagsem(W, ts_unt ).
unit( W ) :- time_unit( W ).
relation( W ) :- tagsem(W, ts_rel ).
relation( W ) :- communication( W ).
communication( W ) :- tagsem(W, ts_com ).
causal_agent( W ) :- tagsem(W, ts_agt ).
causal_agent( W ) :- human( W ).
human( W ) :- tagsem(W, ts_hum ).
artefact( W ) :- tagsem(W, ts_art ).
artefact( W ) :- instrument(W).
instrument( W ) :- tagsem(W, ts_ins ).
instrument( W ) :- container( W ).
container( W ) :- tagsem(W, ts_cnt ).
substance( W ) :- tagsem(W, ts_sub ).
substance( W ) :- chemical_compound( W ).
substance( W ) :- stuff( W ).
chemical_compound( W ) :- tagsem(W, ts_chm ).
stuff( W ) :- tagsem(W, ts_stu ).

% verb %%%%%%%%%%%%%%%%
verb( W ) :- infinitive( W ).
verb( W ) :- participle( W ).
verb( W ) :- conjugated( W ).
verb( W ) :- action_verb( W ).
verb( W ) :- state_verb( W ).
verb( W ) :- modal_verb( W ).
verb( W ) :- temporality_verb( W ).
verb( W ) :- possesion_verb( W ).
verb( W ) :- auxiliary( W ).
infinitive( W ) :- tagcat(W, tc_verb_inf).
participle( W ) :- present_participle( W ).
participle( W ) :- past_participle( W ).
present_participle( W ) :- tagcat(W, tc_verb_prp).
past_participle( W ) :- tagcat(W, tc_verb_pap).
conjugated( W ) :- conjugated_plural(W).
conjugated( W ) :- conjugated_singular(W).
conjugated_plural( W ) :- tagcat(W, tc_verb_pl).
conjugated_singular( W ) :- tagcat(W, tc_verb_sg).
action_verb( W ) :- cognitive_action_verb( W ).
```

```
action_verb( W ) :- physical_action_verb( W ).
cognitive_action_verb( W ) :- tagsem(W, ts_acc ).
physical_action_verb( W ) :- tagsem(W, ts_acp ).
state_verb( W ) :- tagsem(W, ts_eta ).
modal_verb( W ) :- tagsem(W, ts_mod ).
temporality_verb( W ) :- tagsem(W, ts_tem ).
possesion_verb( W ) :- tagsem(W, ts_posv ).
auxiliary( W ) :- tagsem(W, ts_aux ).

% preposition %%%%%%%%
preposition( W ) :- tagcat(W, tc_prep).
preposition( W ) :- spat_preposition( W ).
preposition( W ) :- goal_preposition( W ).
preposition( W ) :- temp_preposition( W ).
preposition( W ) :- manner_preposition( W ).
preposition( W ) :- rel_preposition( W ).
preposition( W ) :- caus_preposition( W ).
preposition( W ) :- neg_preposition( W ).
preposition( W ) :- en_preposition( W ).
preposition( W ) :- sous_preposition( W ).
preposition( W ) :- a_preposition( W ).
preposition( W ) :- de_preposition( W ).
spat_preposition( W ) :- tagsem(W, ts_rspat ).
goal_preposition( W ) :- tagsem(W, ts_rpour ).
temp_preposition( W ) :- tagsem(W, ts_rtemp ).
manner_preposition( W ) :- tagsem(W, ts_rman ).
rel_preposition( W ) :- tagsem(W, ts_rrel ).
caus_preposition( W ) :- tagsem(W, ts_rcaus ).
neg_preposition( W ) :- tagsem(W, ts_rneg ).
en_preposition( W ) :- tagsem(W, ts_ren ).
sous_preposition( W ) :- tagsem(W, ts_rsous ).
a_preposition( W ) :- tagsem(W, ts_ra ).
de_preposition( W ) :- tagsem(W, ts_rde ).

% adjective %%%%%%%%%%%
adjective( W ) :- singular_adjective( W ).
adjective( W ) :- plural_adjective( W ).
adjective( W ) :- verbal_adjective( W ).
adjective( W ) :- comparison_adjective( W ).
adjective( W ) :- concrete_prop_adjective( W ).
adjective( W ) :- abstract_prop_adjective( W ).
adjective( W ) :- nominal_adjective( W ).
singular_adjective( W ) :- tagcat(W, tc_adj_sg).
plural_adjective( W ) :- tagcat(W, tc_adj_pl).
verbal_adjective( W ) :- tagcat(W, tc_verb_adj).
comparison_adjective( W ) :- tagsem(W, ts_acomp ).
concrete_prop_adjective( W ) :- tagsem(W, ts_apty ).
abstract_prop_adjective( W ) :- tagsem(W, ts_apa ).
nominal_adjective( W ) :- tagsem(W, ts_anom ).
```

```
% pronoun %%%%%%%%%%%%
pronoun(W):- rel_pronoun(W).
pronoun(W):- non_rel_pronoun(W).
pronoun(W):- tagsem(W, ts_pron).***
rel_pronoun(W) :- tagcat(W, tc_pron_rel).
non_rel_pronoun(W) :- tagcat(W, tc_pron).

% others %%%%%%%%%%%%
proper_noun( W ) :- tagsem(W, ts_nompropre ).
proper_noun( W ) :- tagsem(W, ts_numero ).
coordinating_conjunction(W) :- tagsem(W, ts_rconj).
subordinating_conjunction(W) :- tagsem(W, ts_subconj).
bracket( W ) :- tagsem(W, ts_paro ).
bracket( W ) :- tagsem(W, ts_parf ).
ponctuation( W ) :- comma( W ).
ponctuation( W ) :- colon( W ).
ponctuation( W ) :- dot( W ).
ponctuation( W ) :- tagcat(W, tc_wpunct).
comma( W ) :- tagsem(W, ts_virg ).
colon( W ) :- tagsem(W, ts_ponct ).
dot( W ) :- tagsem(W, ts_punct ).
void(W) :- tagcat(M,tc_vide).
figures( W ) :- tagsem(W, ts_quant ).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% order
%
precedes(V,N) :- distances(N,V,X,_), 0<X.
precedes(N,V) :- distances(N,V,X,_), 0>X.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% distances in verbs
%
near_verb(N,V) :- distances(N,V,_,1).
near_verb(N,V) :- distances(N,V,_,-1).
far_verb( N,V ) :- distances(N,V,_,X), -1>X , -3<X.
far_verb( N,V ) :- distances(N,V,_,X), 1<X , X<3.
very_far_verb( N,V ) :- distances(N,V,_,X), -2>X.
very_far_verb( N,V ) :- distances(N,V,_,X), X>2.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% distances in words
%
contiguous(N,V) :- distances(N,V,1,_).
contiguous(N,V) :- distances(N,V,-1,_).
near_word(N,V) :- distances(N,V,X,_), -1>X , -4<X.
near_word(N,V) :- distances(N,V,X,_), 1<X , X<4.
far_word(N,V) :- distances(N,V,X,_), -3>X, -8<X.
far_word(N,V) :- distances(N,V,X,_), X>3, X<8.
very_far_word(N,V) :- distances(N,V,X,_), -7>X.
```

```
very_far_word(N,V) :- distances(N,V,X,_), X>7.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% other predicates
suc(X,Y) :- pred(Y,X).
tagcat(Word, POStag) :- tags(Word, POStag, _).
tagsem(Word, Semtag) :- tags(Word, _, Semtag).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% information about examples
tags(m15278_1_deb,tc_vide,ts_vide).
tags(m15278_1,tc_verb_inf,ts_tem).
pred(m15278_1,m15278_1_deb).
...
```

Appendix B. Hypothesis Search Space

A clause space ordered by θ_{OI}-subsumption (see Definition 2, page 506) is in general not a lattice whereas this is the case under θ-subsumption (Semeraro et al., 1994). However, we show in the first section of this appendix that such a clause space can be a lattice when particular assumptions concerning the clauses that it contains are made. A similar proof for our application framework, that is, for the hypothesis search space presented in Section 4.2 with a θ_{NV}-subsumption quasi-ordering, is proposed in the second section.

B.1 Hypothesis Lattice under θ_{OI}-subsumption

A quasi-ordered set under θ_{OI}-subsumption is in general not a lattice since the infimum and supremum are generally not unique in these sets. However, let us consider determinate linked clauses (see Section 4.2) and a space bounded below by a bottom clause ($\perp$). All these conditions ensure the infimum and supremum of two clauses in our hypothesis space to be unique. In this first section, $C \succeq D$ (respectively $C \sim D$) means C is more general (equivalent) than D with respect to the θ_{OI}-subsumption order.

Proposition 4 *For any C and D in the space of linked determinate clauses ordered by θ_{OI}-subsumption, if $C \succeq D$ then the injective substitution θ such that $C\theta \subseteq D$ is unique.*

Proof Reductio ad absurdum. Let us consider that there exist two different injective substitutions θ_1 and θ_2 such that $C\theta_1 \subseteq D$ and $C\theta_2 \subseteq D$. Since θ_1 and θ_2 are injective, $C\theta_1$ and $C\theta_2$ only differ in variable naming. C and D are linked clauses, this means that there exists a literal $l \in C$ such that $l\theta_1 \in D$, $l\theta_2 \in D$ and $l\theta_1 \neq l\theta_2$ where input variables of l are identical in $l\theta_1$ and $l\theta_2$ and output variables are different. This contradicts the fact that all literals are determinate. ■

Proposition 5 *In the space of linked determinate clauses ordered by θ_{OI}-subsumption and bounded below by a bottom clause $\perp$, the supremum of any two clauses is unique.*

Proof Reductio ad absurdum. Let us consider A_1 and A_2 as two different suprema for C_1 and C_2. A_1, A_2, C_1 and C_2 are more general than $\perp$, so there exists a unique $\theta_\perp^{A_1}$ such that $A_1\theta_\perp^{A_1} \subseteq \perp$

(Proposition 4). In the same way, we have unique $\theta_\perp^{A_2}$, $\theta_\perp^{C_1}$, and $\theta_\perp^{C_2}$ such that $A_2\theta_\perp^{A_2} \subseteq \perp$, $C_1\theta_\perp^{C_1} \subseteq \perp$ and $C_2\theta_\perp^{C_2} \subseteq \perp$.

A_1 is a supremum for C_1 so $A_1 \succeq C_1\theta_\perp^{C_1}$ since $C_1 \sim C_1\theta_\perp^{C_1}$. Thus, there exists θ_1 such that $A_1\theta_1 \subseteq C_1\theta_\perp^{C_1}$. Now, $C_1\theta_\perp^{C_1} \subseteq \perp$ therefore $A_1\theta_1 \subseteq \perp$, which means that $\theta_1 = \theta_\perp^{A_1}$ (Proposition 4). Therefore, we have $A_1\theta_\perp^{A_1} \subseteq C_1\theta_\perp^{C_1}$ and in a similar way, $A_1\theta_\perp^{A_1} \subseteq C_2\theta_\perp^{C_2}$, $A_2\theta_\perp^{A_2} \subseteq C_1\theta_\perp^{C_1}$ and $A_2\theta_\perp^{A_2} \subseteq C_2\theta_\perp^{C_2}$.

Let us note $S = A_1\theta_\perp^{A_1} \cup A_2\theta_\perp^{A_2}$. Thus, $S \subseteq C_1\theta_\perp^{C_1}$ and $S \subseteq C_2\theta_\perp^{C_2}$. This means that $S \succeq C_1$ and $S \succeq C_2$ since $C_1\theta_\perp^{C_1} \sim C1$ and $C_2\theta_\perp^{C_2} \sim C2$. Besides, $A_1 \succeq S$, $A_2 \succeq S$ and $S \nsim A_1$, $S \nsim A_2$ because $A_1 \nsim A_2$. This contradicts the fact that A_1 and A_2 are suprema for C_1 and C_2. ■

Proposition 6 *In the space of linked determinate clauses ordered by θ_{OI}-subsumption and bounded below by a bottom clause $\perp$, the infimum of any two clauses is unique.*

Proof Same thing as for supremum, with C_1 and C_2 two infima for A_1 and A_2. Then consider $I = C_1\theta_\perp^{C_1} \cap C_2\theta_\perp^{C_2}$. ■

From Propositions 5 and 6, we can conclude that the space of linked determinate clauses ordered by θ_{OI}-subsumption and bounded below by a bottom clause $\perp$ is a lattice.

B.2 Hypothesis Lattice under θ_{NV}-subsumption

As for θ_{OI}-subsumption, we show that in our application framework, the hypothesis search space ordered by the θ_{NV}-subsumption is a lattice. In the remainder of this appendix, B represents the background knowledge used for our learning task, $\succeq$ and $\sim$ denote the θ_{NV}-subsumption order as defined in Section 4.2.

Proposition 7 *In the space of well-formed clauses ordered by θ_{NV}-subsumption, for any clause C and D, if $C \succeq D$ then the injective substitution θ such that $f(C)\theta \subseteq D$ (with f such that $\forall l \in C,\ B, f(l) \models l$) is unique.*

Proof Same proof as Propostion 4 by considering C^{chain}—the subset of C containing the head literal and all the pred/2 and suc/2 literals of C—and by noting that C^{chain} contains all the variables of C and that with respect to our particular background knowledge, for any f such that $f(C)\theta \subseteq D$ with f such that $\forall l \in C,\ B, f(l) \models l$, necessarily $\forall l \in C^{chain},\ f(l) = l$. ■

Proposition 8 *In the space of well-formed clauses ordered by θ_{NV}-subsumption, the supremum of any two clauses is unique.*

Proof Reductio ad absurdum. Let us consider A_1 and A_2 as two different suprema for C_1 and C_2. A_1 is more general than $\perp$, so $\exists\theta_\perp^{A_1}$ injective and $f_\perp^{A_1}$ such that $f_\perp^{A_1}(A_1)\theta_\perp^{A_1} \subseteq \perp$ and $\theta_\perp^{A_1}$ is unique (Proposition 7). In the same way, we have unique $\theta_\perp^{A_2}$, $\theta_\perp^{C_1}$, and $\theta_\perp^{C_2}$.

A_1 is a supremum for C_1 so there exist θ_1 and f_1 such that $f_1(A_1)\theta_1 \subseteq C_1\theta_\perp^{C_1}$. Now, with A_1^{chain} as defined above, $A_1^{chain}\theta_1 \subseteq C_1^{chain}\theta_\perp^{C_1}$ since $f(A^{chain}) = A^{chain}$. In the same way $C_1^{chain}\theta_\perp^{C_1} \subseteq \perp$. Therefore, we have $A_1^{chain}\theta_1 \subseteq C_1^{chain}\theta_\perp^{C_1} \subseteq \perp$ and then, from Proposition 7, $\theta_1 = \theta_\perp^{A_1}$. Finally, we have $f_1(A_1)\theta_\perp^{A_1} \subseteq C_1\theta_\perp^{C_1}$ and in a similar way, there exist f_2, f_3 and f_4 such that $f_2(A_1)\theta_\perp^{A_1} \subseteq C_2\theta_\perp^{C_2}$, $f_3(A_2)\theta_\perp^{A_2} \subseteq C_1\theta_\perp^{C_1}$ and $f_4(A_2)\theta_\perp^{A_2} \subseteq C_2\theta_\perp^{C_2}$.

Let us note that $S = A_1\theta_\perp^{A_1} \cup A_2\theta_\perp^{A_2} \setminus \{l_1 \mid l_1, l_2 \in (A_1\theta_\perp^{A_1} \cup A_2\theta_\perp^{A_2}), l_1 \neq l_2,$ and $B, l_2 \models l_1\}$. S is a well-formed hypothesis and by construction $S \preceq A_1\theta_\perp^{A_1}$ and $S \preceq A_2\theta_\perp^{A_2}$ and since $A_1\theta_\perp^{A_1} \sim A_1$ and $A_2\theta_\perp^{A_2} \sim A_2$, then $S \preceq A_1$ and $S \preceq A_2$. We define f_5 such that $\forall l_i^S \in S$, $f_5(l_i^S) = f_1(l_i^S)$ if $l_i^S \in A_1\theta_\perp^{A_1}$ and $f_5(l_i^S) = f_3(l_i^S)$ otherwise. Similarly, we define f_6 such that $\forall l_i^S \in S$, $f_6(l_i^S) = f_2(l_i^S)$ if $l_i^S \in A_1\theta_\perp^{A_1}$ and $f_6(l_i^S) = f_4(l_i^S)$ otherwise. Thus, $f_5(S) \subseteq C_1\theta_\perp^{C_1}$ and $f_6(S) \subseteq C_2\theta_\perp^{C_2}$, which means that $S \succeq C_1\theta_\perp^{C_1}$ and $S \succeq C_2\theta_\perp^{C_2}$. Therefore, $S \succeq C_1$ and $S \succeq C_2$. This contradicts the fact that A_1 and A_2 are suprema for C_1 and C_2. ■

Proposition 9 *In the space of well-formed clauses ordered by θ_{NV}-subsumption and with respect to our background knowledge, the infimum of any two clauses is unique.*

Proof Same thing as for supremum, with C_1 and C_2 two infima for A_1 and A_2. Then consider $I = (C_1\theta_\perp^{C_1} \cap C_2\theta_\perp^{C_2}) \cup \{l_1 \mid l_1, l_2 \in C_1\theta_\perp^{C_1} \cup C_2\theta_\perp^{C_2}, l_1 \neq l_2$ and $B, l_2 \models l_1\}$. ■

From Propositions 8 and 9, we can conclude that our hypothesis space ordered by θ_{NV}-subsumption is a lattice.

References

Rajeev Agarwal. *Semantic Feature Extraction from Technical Texts with Limited Human Intervention*. PhD thesis, Mississippi State University, USA, 1995.

Susan Armstrong. MULTEXT: Multilingual text tools and corpora. In H. Feldweg and W. Hinrichs, editors, *Lexikon und Text*, pages 107–119. Max Niemeyer Verlag, Tübingen, Germany, 1996.

Susan Armstrong, Pierrette Bouillon, and Gilbert Robert. Tagger overview. Technical report, ISSCO, University of Geneva, Switzerland, 1995. URL `http://issco-www.unige.ch/staff/robert/tatoo/tagger.html`.

Liviu Badea and Monica Stanciu. Refinement operators can be (weakly) perfect. In Sašo Džeroski and Peter Flach, editors, *Proceedings of the 9th International Conference on Inductive Logic Programming, ILP-99*, volume 1634 of *LNAI*, pages 21–32, Bled, Slovenia, 1999. Springer-Verlag.

Jacques Bouaud, Benoît Habert, Adeline Nazarenko, and Pierre Zweigenbaum. Regroupements issus de dépendances syntaxiques en corpus: Catégorisation et confrontation avec deux modélisations conceptuelles. In Manuel Zacklad, editor, *Proceedings of Ingénierie des Connaissances*, pages 207–223, Roscoff, France, 1997. AFIA - Éditions INRIA Rennes.

Pierrette Bouillon, Robert H. Baud, Gilbert Robert, and Patrick Ruch. Indexing by statistical tagging. In Martin Rajman and Jean-Cédric Chappelier, editors, *Proceedings of Journées d'Analyse statistique des Données Textuelles, JADT'2000*, pages 35–42, Lausanne, Switzerland, 2000a.

Pierrette Bouillon and Federica Busa. *Generativity in the Lexicon*. Cambridge University Press, Cambridge, UK, 2001.

Pierrette Bouillon, Vincent Claveau, Cécile Fabre, and Pascale Sébillot. Using part-of-speech and semantic tagging for the corpus-based learning of qualia structure elements. In Pierrette Bouillon and Kyoko Kanzaki, editors, *Proceedings of First International Workshop on Generative Approaches to the Lexicon, GL'2001*, Geneva, Switzerland, 2001. Geneva University Press.

Pierrette Bouillon, Cécile Fabre, Pascale Sébillot, and Laurence Jacqmin. Apprentissage de ressources lexicales pour l'extension de requêtes. *Traitement Automatique des Langues, special issue: Traitement automatique des langues pour la recherche d'information*, 41(2):367–393, 2000b.

Pierrette Bouillon, Sabine Lehmann, Sandra Manzi, and Dominique Petitpierre. Développement de lexiques à grande échelle. In André Clas, Salah Mejri, and Taïeb Baccouche, editors, *Proceedings of Colloque de Tunis 1997 "La mémoire des mots"*, pages 71–80, Tunis, Tunisia, 1998. Serviced.

Ted Briscoe and John Carroll. Automatic extraction of subcategorisation from corpora. In Paul Jacobs, editor, *Proceedings of 5th ACL conference on Applied Natural Language Processing*, pages 356–363, Washington, USA, 1997. Morgan Kaufmann.

Wray Lindsay Buntine. Generalized subsumption and its application to induction and redundancy. *Artificial Intelligence*, 36(2):149–176, 1988.

Floriana Esposito, Angela Laterza, Donato Malerba, and Giovanni Semeraro. Refinement of Datalog programs. In B. Pfahringer and J. Fürnkranz, editors, *Proceedings of the MLnet Familiarization Workshop on Data Mining with Inductive Logic Programming*, pages 73–94, Bari, Italy, 1996.

Cécile Fabre and Pascale Sébillot. Semantic interpretation of binominal sequences and information retrieval. In *Proceedings of International ICSC Congress on Computational Intelligence: Methods and Applications,CIMA'99, Symposium on Advances in Intelligent Data Analysis AIDA'99*, Rochester, N.Y., USA, 1999.

Cécile Fabre. *Interprétation automatique des séquences binominales en anglais et en français. Application à la recherche d'informations*. PhD thesis, University of Rennes 1, France, 1996.

David Faure and Claire Nédellec. Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM. In Dieter Fensel and Rudi Studer, editors, *Proceedings of 11th European Workshop EKAW'99*, pages 329–334, Dagstuhl, Germany, 1999. Springer-Verlag.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, USA, 1998.

Edith Galy. Repérer en corpus les associations sémantiques privilégiées entre le nom et le verbe: Le cas de la fonction dénotée par le nom. Master's thesis, Université de Toulouse - Le Mirail, France, 2000.

Gregory Grefenstette. Corpus-derived first, second and third-order word affinities. In W. Martin, W. Meijs, M. Moerland, E. ten Pas, P. van Sterkenburg, and P. Vossen, editors, *Proceedings of EURALEX'94*, Amsterdam, The Netherlands, 1994a.

Gregory Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Dordrecht, 1994b.

Gregory Grefenstette. SQLET: Short query linguistic expansion techniques, palliating one-word queries by providing intermediate structure to text. In Luc Devroye and Claude Chrisment, editors, *Proceedings of Recherche d'Informations Assistée par Ordinateur, RIAO'97*, pages 500–509, Montréal, Québec, Canada, 1997.

Benoît Habert, Adeline Nazarenko, and André Salem. *Les linguistiques de corpus*. Armand Collin/Masson, Paris, 1997.

Zelig Harris, Michael Gottfried, Thomas Ryckman, Paul Mattick (Jr), Anne Daladier, Tzvee N. Harris, and Suzanna Harris. *The Form of Information in Science, Analysis of Immunology Sublanguage*. Kluwer Academic Publisher, Dordrecht, 1989.

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In Christian Boitet, editor, *Proceedings of 14th International Conference on Computational Linguistics, COLING-92*, pages 539–545, Nantes, France, 1992.

Nicolas Helft. Inductive Generalization: A logical framework. In Ivan Bratko and Nada Lavrac, editors, *Proceedings of the 2nd European Working Session on Learning, EWSL*, pages 149–157, Bled, Yugoslavia, 1987. Sigma Press.

Nancy Ide and Jean Véronis. MULTEXT (multilingual tools and corpora). In *Proceedings of 15th International Conference on Computational Linguistics, COLING-94*, pages 90–96, Kyoto, Japan, 1994. Morgan Kaufmann.

Bernard Jones. Can punctuation help parsing? Technical Report 29, Centre for Cognitive Science, University of Edinburgh, UK, 1994.

Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Chris S. Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 95*, pages 1137–1145, Montréal, Québec, Canada, 1995. Morgan Kaufmann.

Emmanuel Morin. *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques*. PhD thesis, Université de Nantes, France, 1999.

Stephen Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13(3-4):245–286, 1995.

Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19-20:629–679, 1994.

Stephen Muggleton and Cao Feng. Efficient induction of logic programs. In Setsuo Arikawa, S. Goto, Setsuo Ohsuga, and Takashi Yokomori, editors, *Proceedings of the 1st Conference on Algorithmic Learning Theory*, pages 368–381, Tokyo, Japan, 1990. Springer-Verlag - Ohmsha.

Claire Nédellec, Céline Rouveirol, Hilde Adé, Francesco Bergadano, and Birgit Tausend. Declarative bias in inductive logic programming. In Luc De Raedt, editor, *Advances in Inductive Logic Programming*, pages 82–103. IOS Press, Amsterdam, 1996.

Shan-Hwei Nienhuys-Cheng and Ronald de Wolf. Least generalizations and greatest specializations of sets of clauses. *Journal of Artificial Intelligence Research*, 4:341–363, 1996.

Dominique Petitpierre and Graham Russell. MMORPH - the multext morphology program. Technical report, ISSCO, University of Geneva, Switzerland, 1998.

Ronan Pichon and Pascale Sébillot. Acquisition automatique d'informations lexicales à partir de corpus: Un bilan. Research report 3321, INRIA, Rennes, France, 1997.

Ronan Pichon and Pascale Sébillot. From corpus to lexicon: From contexts to semantic features. In Barbara Lewandowska-Tomaszczyk and Patrick James Melia, editors, *Proceedings of Practical Applications in Language Corpora, PALC'99*, volume 1 of *Lodz studies in Language*, pages 375–389. Peter Lang, 2000.

Gordon D. Plotkin. A note on inductive generalization. In B. Meltzer and D. Michie, editors, *Machine Intelligence 5*, pages 153–163, Edinburgh, 1970. Edinburgh University Press.

James Pustejovsky. *The Generative Lexicon*. MIT Press, Cambridge, MA, USA, 1995.

James Pustejovsky, Peter Anick, and Sabine Bergler. Lexical semantic techniques for corpus analysis. *Computational Linguistics*, 19(2):331–358, 1993.

John Ross Quinlan. Learning logical definitions from relations. *Machine Learning*, 5(3):239–266, 1990.

Anke Rieger. Optimizing chain Datalog programs and their inference procedures. LS-8 Report 20, University of Dortmund, Lehrstuhl Informatik VIII, Dortmund, Germany, 1996.

Gerard Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

Pascale Sébillot, Pierrette Bouillon, and Cécile Fabre. Inductive logic programming for corpus-based acquisition of semantic lexicons. In Claire Cardie, Walter Daelemans, Claire Nédellec, and Erik Tjong Kim Sang, editors, *Proceedings of the Fourth Conference on Computational Natural Language Learning (CoNLL-2000) and of the Second Learning Language in Logic Workshop (LLL-2000)*, pages 199–208, Lisbon, Portugal, September 2000.

Giovanni Semeraro, Floriana Esposito, Donato Malerba, Clifford Brunk, and Michael J. Pazzani. Avoiding non-termination when learning logic programs: A case study with FOIL and FOCL. In L. Fribourg and F. Turini, editors, *Proceedings of Logic Program Synthesis and Transformation - MetaProgramming in Logic, LOPSTR 1994*, volume 883 of *LNCS*, pages 183–198. Springer-Verlag, 1994.

Alan F. Smeaton. Using NLP or NLP resources for information retrieval tasks. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*, pages 99–111. Kluwer Academic Publishers, Dordrecht, 1999.

Karen Spärck Jones. What is the role of NLP in text retrieval? In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*, pages 1–24. Kluwer Academic Publishers, Dordrecht, 1999.

Tomek Strzalkowski. Natural language information retrieval. *Information Processing and Management*, 31(3):397–417, 1995.

Fabien Torre and Céline Rouveirol. Natural ideal operators in inductive logic programming. In M. van Someren and Widmer G., editors, *Proceedings of 9th European Conference on Machine Learning (ECML'97)*, volume 1224 of *LNAI*, pages 274–289, Prague, Czech Republic, April 1997a. Springer-Verlag.

Fabien Torre and Céline Rouveirol. Opérateurs naturels en programmation logique inductive. In Henri Soldano, editor, *12èmes Journées Françaises d'Apprentissage (JFA'97)*, pages 53–64, Roscoff, France, 1997b. AFIA - Éditions INRIA Rennes.

Fabien Torre and Céline Rouveirol. Private properties and natural relations in inductive logic programming. Technical Report 1118, Laboratoire de Recherche en Informatique d'Orsay (LRI), France, July 1997c.

Laurence Vandenbroucke. Indexation automatique par couples nom-verbe pertinents. DES information and documentation report, Faculté de Philosophie et Lettres, Université Libre de Bruxelles, Belgium, 2000.

Ellen M. Voorhees. Query expansion using lexical-semantic relations. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of ACM SIGIR'94*, Dublin, Ireland, 1994. ACM - Springer-Verlag.

Stefan Wermter, Ellen Riloff, and Gabriele Scheler, editors. *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, volume 1040 of *LNCS*. Springer-Verlag, 1996.

Yorick Wilks and Mark Stevenson. The grammar of sense: Is word-sense tagging much more than part-of-speech tagging? Technical report, University of Sheffield, UK, 1996.