In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 199–208, Lisbon, Portugal, 2000.

Inductive Logic Programming for Corpus-Based Acquisition of Semantic Lexicons

Pascale Sébillot
IRISA - Campus de Beaulieu - 35042 Rennes cedex - France
sebillot@irisa.fr

Pierrette Bouillon
TIM/ISSCO - ETI - Université de Genève - 40 Bvd du Pont-d'Arve - CH-1205 Geneva - Switzerland
Pierrette.Bouillon@issco.unige.ch

Cécile Fabre
ERSS - Université de Toulouse II - 5 allées A. Machado - 31058 Toulouse cedex - France
cfabre@univ-tlse2.fr

Abstract

In this paper, we propose an Inductive Logic Programming learning method which aims at automatically extracting special Noun-Verb (N-V) pairs from a corpus 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 Inductive Logic Programming learning method that we have developed enables us to automatically extract from a corpus N-V pairs whose elements are linked by one of the semantic relations defined in the qualia structure in GL, and to distinguish them, in terms of surrounding categorial context from N-V pairs also present in sentences of the corpus but not relevant. This method has been theoretically and empirically validated, on a technical corpus. The N-V pairs that have been extracted will further be used in information retrieval applications for index expansion[1].

Keywords: Lexicon learning, Generative Lexicon, Inductive Logic Programming, Information indexing.

1 Introduction

Information retrieval (IR) systems aim at providing a user who asks a query to a database of documents with the most relevant texts. The quality of these systems is usually measured with the help of two criteria: the *recall rate*, which corresponds to the proportion of relevant answers that have been given by the system compared to the total number of relevant answers in the database, and the *precision rate*, which denotes the proportion of relevant answers that are present among the given answers.

In these IR systems, texts and queries are usually represented by indexes, that is, a collection of some of the words that they contain. The quality of the systems therefore highly depends on the type of indexing language that has been chosen. Two kinds of indexes exist: simple indexes, which correspond to simple nouns (N), verbs (V) and/or adjectives (A) that occur in a text or a query[2], and complex indexes, which correspond to the compounds (for example, NN compounds) present in the document or

[1]This works is funded by the Agence universitaire de la Francophonie (AUF) (Action de recherche partagée *"Acquisition automatique d'éléments du Lexique Génératif pour améliorer les performances de systèmes de recherche d'information"*, réseau FRANCIL).

[2]All the simple N, V and/or A can be kept as indexes, or the most frequent ones for a given text, or those whose frequencies in this text are especially high compared to their frequencies in the database, etc.

the question. The solutions that are given for a user query are the texts whose indexes better match the query index.

In order to obtain the hightest performances, IR systems usually offer some possibilities to expand both query and text indexes. Traditional index expansion concerns morpho-syntactic similarities; for example, the same index words in plural and singular forms can be matched. Some other systems deal with a kind of semantic similarities: if they possess a linguistic knowledge database, they can, for example, expand a nominal index by following synonymy or hyperonymy links. These systems are however usually limited to intra-categorial expansion, especially N-to-N one. Here we deal with a new kind of expansion that has been proven 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. For example, our objective is to permit a matching between a query index *disk store* and the text formulation *to sell disks*, related by the typical function of a store.

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, 2000) to define what a relevant N-V link is, that is, what is a N-V pair in which the N and the V are related by a semantic link which is close, and which can therefore be used to expand indexes.

In GL formalism, lexical entries consist in 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. We assert that these N-V links are especially relevant for index expansion in IR systems (Fabre and Sébillot, 1999), and what we call a relevant N-V pair afterwards in the paper is a pair composed of a N and a V which are related by one of the four semantic relations defined in the qualia structure in GL.

GL is however currently just a formalism; no generative lexicons exist that are precise enough for every domain and every application (for eg. IR), and the cost of a manual construction of a lexicon based on GL principles is prohibitive. Moreover the real N-V links that are the key-point of this formalism cannot be defined *a priori* and have to be acquired from corpora of the studied domain. The aim of this paper is therefore 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 qualia structure in GL, and to distinguish them, in terms of surrounding categorial (Part-of-Speech, POS) context from N-V pairs also present in sentences of the corpus but not relevant. It will be divided in three parts. Section 2 focusses on the motivation of this project regarding the use of GL. Section 3 explains the machine learning method that we have developed. Section 4 is dedicated to its theoretical and empirical validations, and to the results of its application to a technical corpus.

2 Motivation

As stated in the introduction, our work makes two strong claims: firstly N-V associations defined in GL are relevant for IR and secondly this information can be acquired from a corpus on the basis of surrounding POS context. These presuppositions have to be motivated before explaining the learning method:

1. The aim of GL is to define underspecified lexical representations that will acquire their specifications in context. For example, the qualia structure of *book* indicates that its default function is *read* and that it is created by the act of writing. But this information has to be enriched in context in order to characterize how words are used in specific domains. For example, the qualia structure of *book* will also have to indicate that the *book* can be *shelved* or *indexed* if this information is necessary to interpret texts from information science domain. GL

is therefore a theory of words in context. It can also be seen as a way to structure information in corpora and, in that sense, the relations it defines are therefore privileged information for IR. In this perspective, GL has been preferred to existing lexical resources such as WordNet (Fellbaum, 1998) for two main reasons: lexical relations that 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 can not be used as such to describe lexical associations in specific texts, considering the great variability of semantic associations from one domain to another.

2. In GL, the qualia structures are not arbitrary repository of information. They contain the information necessary to explain the syntactic behaviour of the item. We would therefore expect that there are strong connections between some specific syntactic phenomena and some specific qualia relations. For example, the middle construction seems to be only possible if a telic relation holds between the N and V (Bassac and Bouillon, 2000) (for example: *??this book writes well* vs *this book reads well*). Similarly, imperative constructions (e.g. *open the door*, *follow the links*) or adjectival sentences (*a book difficult to write/read*) may also indicate a qualia relation. These are some of the constructions that we want to identify primilarly in corpora by the learning method.

3 The machine learning method

Trying to infer lexical semantic information from corpora is not new: lots of works have already been conducted on this subject, especially in the statistical learning domain (see (Grefenstette, 1994b), for e.g., 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, 1997) focus on particular lexical relations, like hyperonymy. Some of these works are concerned with automatically obtaining more complete lexical semantic representations ((Grefenstette, 1994b; Pichon and Sébillot, 1999). Among these studies, (Pustejovsky et al., 1993) presents a research whose aim is to acquire GL nominal qualia structures from a corpus; this work is however quite different from ours because it supposes that the qualia structure contents are initialized and are only refined with the help of the corpus by using the type coercion[3] mechanism.

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 machine learning method. This section is devoted to the explanation of this choice and to the description of the method that we have developed.

Machine learning aims at automatically building programs from examples that are known to be positive or negative examples of their runnings. According to Mitchell (Mitchell, 1997), "*a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improve with experience E*".

Among different machine learning techniques, we have chosen the Inductive Logic Programming framework (ILP) (Muggleton and De-Raedt, 1994) to learn from a textual corpus N-V pairs that are related in terms of one of the relations defined in the qualia structure in GL. Programs that are infered from a set of facts and a background knowledge are here logic programs, that is, sets of Horn clauses. In the ILP framework, the main idea is to obtain a set of generalized clauses that is sufficiently generic to cover the majority of the positive examples (E^+), and sufficiently specific to rightly correspond to the concept we want to learn and to cover no (or a few - some noise can be allowed) negative example(s) (E^-). For our experiment,

[3] A semantic operation that converts an argument to the type which is expected by a function, where it would otherwise result in a type error.

we furnish a set of N-V pairs related by one of the qualia relations within a POS context (E^+), and a set of N-V pairs that are not semantically linked (E^-), and the method infers general rules (clauses) that explain these E^+. This particular explanatory characteristic of ILP has motivated our choice: ILP does not just provide a predictor (this N-V pair is relevant, this one is not) but also a data-based theory. Contrary to some statistical methods, it does not just give raw results but explains the concept that is learnt[4].

We use Progol (Muggleton, 1995) for our project, Muggleton's ILP implementation that has already been proven well suited to deal with a big amount of data in multiple domains, and to lead to results comparable to other ILP implementations (Roberts et al., 1998).

In this section we briefly describe the corpus on which our experiment has been conducted. We then explain the elaboration of E^+ and E^- for Progol. We finally present the generalized clauses that we obtain. The validation of the method is detailed in section 4.

3.1 The corpus

The French corpus used in this project is a 700 kBytes handbook of helicopter maintenance, given to us by MATRA CCR Aérospatiale, which contains more than 104000 word occurrences[5]. The MATRA CCR corpus has some special characteristics that are especially well suited for our task: it is coherent; it contains lots of 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.

This corpus has been POS-tagged with the help of annotation tools developed in the MULTEXT project (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 receive one POS-tag, with less than 2% of errors.

3.2 Example construction

The first task consists in building up E^+ and E^- for Progol, in order for it to infer generalized clauses that explain what, in the POS context of N-V pairs, distinguishes the relevant pairs from the not relevant ones. Work has to be done to determine what is the most appropriate context for this task. We just present here the solution we have finally chosen. Section 4 describes methods and measures to evaluate the "quality" of the learning that enable us to choose between the different contextual possibilities. Here is our methodology for the construction of the examples.

We first consider all the nouns of the MATRA CCR corpus. More precisely, we only deal with a 81314 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 1489 different N (29633 noun occurrences) and 567 different V (9522 verb occurrences). For each N of this subcorpus, the 10 most strongly associated V, in terms of Chi-square, are selected. This first step both produces pairs that are really bound by one qualia relation (*(écrou, serrer)*)[6] and pairs that are fully irrelevant (*(roue, prescrire)*)[7].

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 E^+, that is for each of the previous pairs that has 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 E^+. Here is the form of the positive examples:

POSITIVE(*category_before_N, category_after_N, category_before_V, V_type, distance, position*).

where *V_type* indicates if the V is an infinitive form, etc., *distance* corresponds to the number

[4]Learning with ILP has already been successfully used in natural language processing, for example in corpus POS-tagging (Cussens, 1996) or semantic interpretation (Mooney, 1999).

[5]104212 word occurrences.

[6](nut, tighten).

[7](wheel, prescribe)

of verbs between the N and the V, and *position* is POS (for positive) if the V appears before the N in the sentence, NEG if the N appears before the V.

For example,

POSITIVE(VRBINF, P_DE, VID, VRBINF, 0, POS).

means that a N-V pair, in which the N is surrounded with an infinitive verb on its left (VRBINF) and a preposition *de*[8] (P_DE) on its right, in which the V is preceded by nothing[9] (VID)[10] and is an infinitive one (VRBINF), in which no verb exists between the N and the V (0), and in which the V appears before the N in the sentence (POS), is a relevant pair (for example, in *ouvrir la porte de ...*).

The E^- are elaborated in the same way than the E^+, with the same Perl program. E^- and E^+ forms are identical, except the presence of a sign *:-* before the predicate POSITIVE to denote a E^-:

:-POSITIVE(*category_before_N, category_after_N, category_before_V, V_type, distance, position*).

These E^- are automatically built from the previous highly correlated N-V pairs that have been manually annotated as irrelevant. For example,

:-POSITIVE(VID, P_PAR, NC, VRBPP, 0, NEG).

means that a N-V pair, in which the N has nothing on its left (VID) and a preposition *par*[11] (P_PAR) on its right, in which the V is preceded by a noun (NC) and is a past participle (VRBPP), in which no verb exists between the N and the V (0), and in which the V appears after the N in the sentence (NEG), is an irrelevant pair (for example, in *freinage par goupilles fendues*).

4031 E^+ and about 7000 E^- are automatically produced this way from the corpus.

[8] Of.

[9] Or by one of the three categories that we do not consider for example elaboration, that is, determiners, adverbs and adjectives.

[10] Empty.

[11] By.

3.3 Learning with the help of Progol

These E^+ and E^- are then furnish to Progol in order for it to try to infer generalized clauses that explain the concept "qualia pair" *versus* "not qualia pair". We do not discuss here either parameter setting that concerns the choice of the example POS context, or evaluation criteria; this discussion is postponed to next section; we simply present the learning method and the type of generalized clauses that we have obtained.

Some information have to be given to Progol for it to know what are the categories that can undergo a generalization. For example, if two E^+ are identical but possess different locative prepositions as second arguments (for eg. *sur*[12] and *sous*[13]), must Progol produce a generalization corresponding to the same clause except that the second argument is replaced by the general one: *locative-preposition*, or by a still more general one: *preposition*?

The *background knowledge* used by Progol is knowledge on the domain. For example here, it contains the fact that a verb can be found in the corpus in an infinitive or a conjugated form, etc.

verbe(V) :- infinitif(V).
verbe(V) :- conjugue(V).

and that an infinitive form is denoted by the tag VERBINF, and a conjugated form by the tags VERB-PL and VERB-SG, etc.

infinitif(verbinf).
conjugue(verb-pl).
conjugue(verb-sg).

When Progol is provided with all this knowledge, learning can begun. The output of Progol is of two kinds: some clauses that have not at all been generalized (that is, some of the E^+), and some generalized clauses; we call the set of these generalized clauses G, and it is this set G that interests us here. Here is an example of one of the generalized clauses that we have obtained in our experiment:

POSITIVE(A, C, C, D, E, F) :-
PREPOSITIONLIEU(A), VIDE(C), VERBINF(D), PRES(E). **(1)**

[12] On.

[13] Under.

which means that N-V pairs (i) in which the category before the N is a locative preposition (PREPOSITIONLIEU(A)), (ii) in which there is nothing after the N and before the V (VIDE(C) for the second and third arguments), (iii) in which the V is an infinitive one (VERBINF(D)), and (iv) in which there is no verb between the N and the V (proximity denoted by PRES(E)[14]), are relevant. No constraint is set on N/V order in the sentences.

This generalized clause covers, for example, the following E^+:

POSITIVE(P_SUR, VID, VID, VERBINF, 0, POS).

which corresponds to the relevant pair *(prise, brancher)*[15] that is detected in the corpus in the sentence *"Brancher les connecteurs sur les prises électriques."*.

Some of the generalized clauses in G cover lots of E^+, others far less. We now present a method to detect what the "good" clauses are, that is, the clauses that explain the concept that we want to learn, and a measure of the "quality" of the learning that has been conducted.

4 Learning validation and results

This section is dedicated to two aspects of the validation of our machine learning method. First we define the theoretical validation of the learning, that is, we focus on the determination of a means to detect what are the "good" generalized clauses, and of a measure of the quality of the concept learning; this parameter setting and evaluation criterion phase explains how we have chosen the precise POS context for N-V pairs in the E^+ and E^- (as described in subsection 3.2): the six contextual elements in examples are the combination that leads to the best results in terms of the learning quality measure that we have chosen. The second step of the validation is the empirical one. We have applied the generalized clauses that have been selected to the Matra CCR corpus and have evaluated the quality of the results in terms of pairs that are indicated relevant or not. Here are these two phases.

[14]Close(E).

[15](plug, to plug in).

4.1 Theoretical validation

As we have previously noticed, among the generalized clauses produced from our E^+ and E^- by Progol (set G), some of them cover a lot of E^+, others only a few of them. What we want is to get a way to automatically find what are the generalized clauses that have to be kept in order to explain the concept we want to learn.

We have first defined a measure of the theoretical generality of the clauses[16]. The theoretical generality of a generalized clause is the number of not generalized clauses (E^+) that this clause can cover. For example, both

POSITIVE(P_AUTOURDE, VID, VID, VERBINF, 0, NEG).

and

POSITIVE(P_CHEZ, VID, VID, VERBINF, 0, POS).

can be covered by clause **(1)** (cf. subsection 3.3). During the study of, for example, the distribution of the number of clauses in G on these different theoretical generality values, our "hope" is to obtain a gaussian-like graph in order to automatically select all the clauses present under the gaussian plot, or to calculate two thresholds that cover 95% of these clauses and to reject the other 5%. This distribution is however not a gaussian one.

Our second try has not only concerned the theoretical coverage of G clauses but also their empirical coverage. This second measure that we have defined is the number of E^+ that are really covered by each clause of G. We then consider the distribution of the empirical coverage of G clauses on the theoretical coverages of these clauses, that is, we consider the graph in which, for each different theoretical measure value for G clauses, we draw a line whose length corresponds to the total number of E^+ covered by the G clauses that have *this* theoretical coverage value. Here two gaussians clearly appear (cf. figure 1), one for rather specific clauses and the other for more general ones. We have therefore decided to keep all the generalized clauses produced by Progol.

[16]We thank J. Nicolas, INRIA researcher at IRISA, for his help on this point.

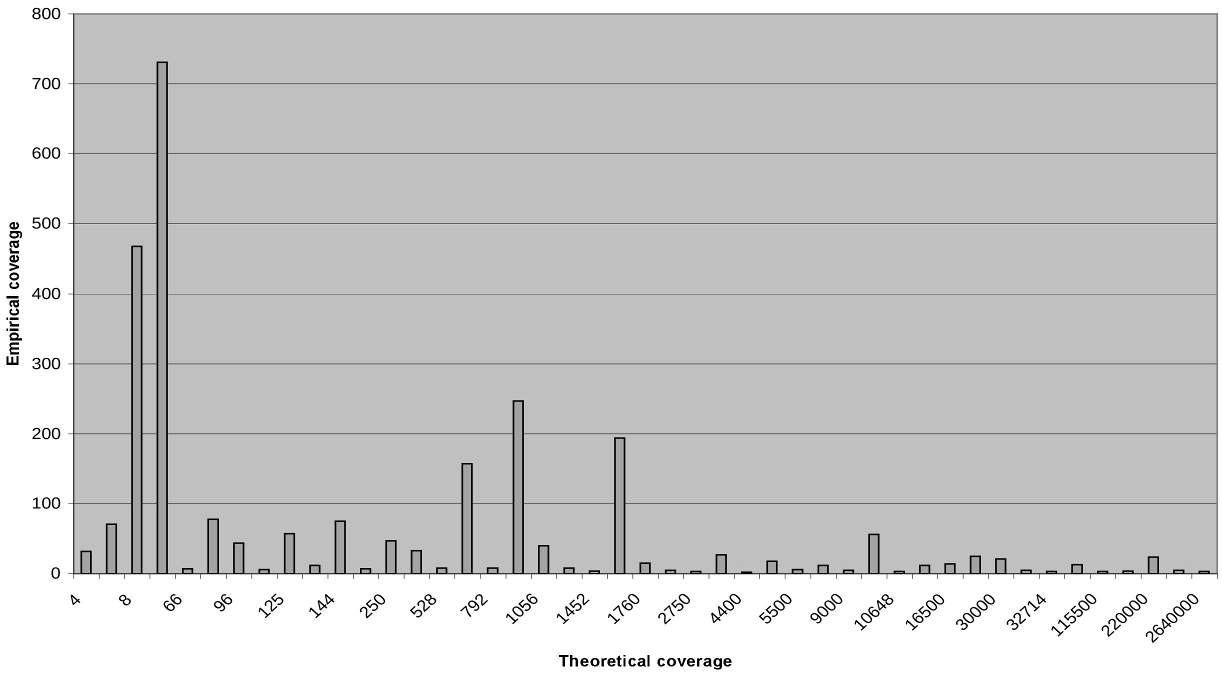

Figure 1: Distribution of positive examples on clauses

The second point concerns the determination of a measure of the quality of the learning for the parameter setting. We are especially interested in the percentage of E^+ that are covered by the generalized clauses, and if we permit some noise in Progol parameter adjustment to allow more generalizations, by the percentage of E^- that are rejected by these generalized clauses. The measure of the recall and the precision rates of the learning method can be summarized in a Pearson coefficient:

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

where A = *actual*, Pr = *predicated*, P = *positive*, N= *negative*, T= *true*, F= *false*; the more close to 1 this value is, the better the learning is.

The results for our learning method with a rate of Progol noise equal to 0 are the following: from the 4031 initial E^+ and the 6922 initial E^-, the 109 generalized clauses produced by Progol cover 2485 E^+ and 0 E^-; 1546 E^+ and 6922 E^- are therefore uncovered; the value of the Pearson coefficient is 0.71. (NB: Figure 1 illustrates these results).

We have developed a Perl program whose role is to find which Progol noise rate leads to the best results. This Progol noise rate is equal to 37. With this rate, the results are the following: from the 4031 initial E^+ and the 6922 initial E^-, the 66 generalized clauses produced by Progol cover 3547 E^+ and 348 E^-; 484 E^+ and 6574 E^- are therefore uncovered; the value of the Pearson coefficient is 0.84. The stability of the set of learnt generalized clauses has been tested.

4.2 Empirical validation

In order to evaluate the empirical validity of our learning method, we have applied the 66 generalized clauses to the Matra CCR corpus and have studied the appropriateness of the pairs that are stated relevant or irrelevant by them. Of course, it is impossible to test all the N-V combinations present in such a corpus. Our evaluation has focussed on some of the significant nouns of the domain.

A Perl program presents to one expert all the N-V pairs that appear in one sentence in a part of the corpus and include one of the studied nouns. The expert manually tags each pair as relevant or not. This tagging is then compared to the results obtained for these N-V pairs of the same part of the corpus by the application of the generalized clauses learnt with Progol.

The results for seven significant nouns (*vis, écrou, porte, voyant, prise, capot, bouchon*)[17] are presented in table 1. In the left column, one N-V pair is considered as tagged *"relevant"* by the generalized clauses if at least one of them covers this pair; in the right one, at least six different clauses of G must cover a pair for it to be said correctly detected by the generalized clauses; the aim of this second test is to reduce noise in the results.

1 occurrence	6 occurrences
correctly found: 49	correctly found: 23
incorrectly found: 54	incorrectly found: 4
not found: 10	not found: 36
Pearson = 0.5138	Pearson = 0.5209

Table 1: Empirical validation on Matra CCR corpus

The results are quite promising, especially if we compare them to those obtain by Chi-square correlation (cf. table 2). This comparison is interesting because Chi-square is the first step of our selection of N-V couples in the corpus (cf. subsection 3.2).

correctly found: 38
incorrectly found: 124
not found: 21
Pearson = 0.1206

Table 2: Chi-square results on Matra CCR corpus

5 Conclusion

The Inductive Logic Programming learning method that we have proposed in order to define what is a N-V pair whose elements are bound by one of the qualia relations in Pustejovsky's Generative Lexicon formalism leads to very promising results: 83.05% of relevant pairs (after one occurrence) are detected for seven significant nouns; these results have to be compared with the 64% results of Chi-square. It is worth noticing that beyond this simple comparison with one of the possible pure statistics based method[18], the interest of using ILP learning is its explanatory characteristic; and it is *this* characteristic that have motivated our choice: contrary to statistical methods, our ILP method does not just extract statistically correlated pairs but it permits to automatically learn rules that distinguish relevant pairs from others.

The fact that noise has to be used in Progol to obtain these results however means that something is missing in our E^+ to fully define the concept "qualia pair" *versus* "not qualia pair"; some E^- have to be covered to define it better. A piece of information, maybe syntactic and/or semantic is missing in our E^+ to fully characterize it. This fact can be easily illustrated by the following example: 'Verbinf det N' structures are generally relevant (*ouvrir la porte*[19], etc.), except when the N indicates a collection of objects (*nettoyer l'ensemble du réservoir*[20]) or a part of an object (*vider le fond du réservoir*[21]). A simple POS-tagging of the sentences offers no difference between them. We are currently working on a semantic tagging of the Matra CCR corpus in order to improve the results.

Another future work concerns the automatic distinction between the various qualia roles during learning. The last phase of the project will deal with the real use of the N-V pairs obtained by the machine learning method within one information retrieval system and the evaluation of the improvement of its performances.

References

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

Susan Armstrong, Pierrette Bouillon, and Gilbert Robert. 1995. Tagger Overview.

[17](screw, nut, door, indicator signal, plug, cowl, cap).

[18]This comparison could be extended to other corpus frequency based technics (mutual information, etc.).

[19]Open the door.

[20]Clean the whole tank.

[21]Empty the tank bottom.

Technical report, ISSCO, (http://issco-www.unige.ch/staff/robert/tatoo/tagger.html).

Susan Armstrong. 1996. Multext: Multilingual Text Tools and Corpora. In H. Feldweg and W. Hinrichs, editors, *Lexikon und Text*, pages 107–119. Tübingen: Niemeyer.

Christian Bassac and Pierrette Bouillon. 2000. The Polymorphism of Verbs Exhibiting Middle Transitive Alternations in English. Technical report, ISSCO.

Jacques Bouaud, Benoît Habert, Adeline Nazarenko, and Pierre Zweigenbaum. 1997. Regroupements issus de dépendances syntaxiques en corpus : catégorisation et confrontation avec deux modélisations conceptuelles. In *Proceedings of Ingénierie de la Connaissance*, Roscoff, France.

Pierrette Bouillon and Federica Busa. 2000. *Generativity in the Lexicon*. CUP:Cambridge, In Press.

Pierrette Bouillon, Sabine Lehmann, Sandra Manzi, and Dominique Petitpierre. 1998. Développement de lexiques à grande échelle. In *Proceedings of colloque de Tunis 1997 "La mémoire des mots"*, Tunis, Tunisie.

Ted Briscoe and John Carroll. 1997. Automatic Extraction of Subcategorisation from Corpora. In *Proceedings of 5th ACL conference on Applied Natural Language Processing*, Washington, USA.

James Cussens. 1996. Part-of-Speech Disambiguation using ILP. Technical report, Oxford University Computing Laboratory.

Cécile Fabre and Pascale Sébillot. 1999. 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.

David Faure and Claire Nédellec. 1999. Knowledge Acquisition of Predicate Argument Structures from Technical Texts using Machine Learning: the System ASIUM. In Dieter Fensel Rudi Studer, editor, *Proceedings of 11th European Workshop EKAW'99*, Dagstuhl, Germany. Springer-Verlag.

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

Gregory Grefenstette. 1994a. Corpus-Derived First, Second and Third-Order Word Affinities. In *Proceedings of EURALEX'94*, Amsterdam, The Netherlands.

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

Gregory Grefenstette. 1997. SQLET: Short Query Linguistic Expansion Techniques, Palliating One-Word Queries by Providing Intermediate Structure to Text. In McGill-University, editor, *Proceedings of Recherche d'Informations Assistée par Ordinateur, RIAO'97*, Montréal, Québec, Canada.

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

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

Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of 15th International Conference on Computational Linguistics, COLING-92*, Nantes, France.

Tom M. Mitchell. 1997. *Machine Learning*. McGraw-Hill.

Raymond Mooney. 1999. Learning for Semantic Interpretation: Scaling Up without Dumbing Down. In *Proceedings of Learning Language in Logic, LLL99*, Bled, Slovenia.

Emmanuel Morin. 1997. Extraction de liens sémantiques entre termes dans des corpus de textes techniques : application à l'hyponymie. In *Proceedings of Traitement Automatique des Langues Naturelles, TALN'97*, Grenoble, France.

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

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

Dominique Petitpierre and Graham Russell. 1998. Mmorph - the Multext Morphology Program. Technical report, ISSCO.

Ronan Pichon and Pascale Sébillot. 1997. Acquisition automatique d'informations lexicales à partir de corpus : un bilan. Research report n^{0}3321, INRIA, Rennes.

Ronan Pichon and Pascale Sébillot. 1999. From Corpus to Lexicon: from Contexts to Semantic Features. In *Proceedings of Practical Applications in Language Corpora, PALC'99, to appear*, Lodz, Poland.

James Pustejovsky, Peter Anick, and Sabine Bergler. 1993. Lexical Semantic Techniques for Corpus Analysis. *Computational Linguistics*, 19(2).

James Pustejovsky. 1995. *The Generative Lexicon*. Cambridge:MIT Press.

Sam Roberts, Wim Van-Laer, Nico Jacobs, Stephen Muggleton, and Jeremy Broughton. 1998. A Comparison of ILP and Propositional Systems on

Propositional Data. In Springer-Verlag, editor, *Proceedings of 8th International Workshop on Inductive Logic Programming, ILP-98*, Berlin, Germany. LNAI 1446.