Using a Symbolic Machine Learning Tool to Refine Lexico-syntactic Patterns

Rapport de Recherche IRIN – 183

Emmanuelle MARTIENNE & Emmanuel MORIN

Institut de Recherche en Informatique de Nantes
2, rue de la Houssinière - BP 92208
44322 Nantes cedex 03 - FRANCE

{martien,morin}@irin.univ-nantes.fr

April 17, 1999

Contents

List of Figures

Chapter 1

Introduction

As the amount of electronic documents (corpora, dictionaries, newspapers, news-wires, etc.) become more and more important and diversified, there is a need to extract information automatically from these texts.

Extracting information from text is an important task for Natural Language Processing researchers. In contrast to text understanding, information extraction systems do not aim to make sense of the entire text, but are only focused on fractions of the text that are relevant to a specific domain (Hobbs et al., 1992).

In information extraction, the data to be extracted from text is given by a syntactic pattern (or template) which typically involves recognizing a group of entities, generally noun phrases, and relationships between these entities. For instance, Figure 1.1 shows part of a text about terrorism domain from MUC-3 (1991), and the corresponding slots of the filled template.

SENTENCE

Bogota, 3 Apr 90 (Inravision Television Cadena 1) - [REPORT] Jorge Alonso Sierra Valencia [TEXT] Liberal Senator Federico Estrada Velez was kidnapped on 3 April at the corner oh 60th and 48th streets in Western Medellin, only 100 meters from a metropotitan police CAI (Immediate Attention Center).

...

TEMPLATE

DATE OF INCIDENT : 03 APR 90
TYPE OF INCIDENT : KIDNAPPING
CATEGORY OF INCIDENT : TERRORIST ACT
HUMAN TARGET : Federico Estrada Velez (Liberal Senator)
...

Figure 1.1: A sample message and associated filled template.

In recent years, through MUC conferences several information extraction systems have been developed for a variety of domains, such as Latin American terrorism (MUC-3, 1991; MUC-4, 1992), international joint ventures and electronic circuit fabrication (MUC-5, 1993), and

company management changes (MUC-6, 1995).

However, many of the best-performing systems are difficult and time-consuming to build, and generally contain domain-specific components. Therefore, their success is often tempered by difficulties of adapting to new domains. Having the use of specialists' abilities for each domain is not reasonable.

In order to overcome such weakness, we have developed the PROMÉTHÉE system, dedicated to the extraction of lexico-syntactic patterns relative to a specific conceptual relation from a technical corpus (Morin, 1998).

Based on our experience, we believe that such patterns are too general : indeed without using manual constraints, their *Recall* is satisfying but *Precision* is low. In order to refine these patterns, we propose to use a learning system, called EAGLE (Martienne and Quafafou, 1998), which is based on the *learning from examples* paradigm (Muggleton, 1991). This latter extracts intensional descriptions of concepts, from their extensional descriptions including their ground examples and counter-examples. The learned definitions are further used in recognition or classification tasks.

The interfacing of the two systems is performed as follows: (1) lexico-syntactic patterns are extracted by PROMÉTHÉE, (2) some instances of these patterns are then produced from a corpus, and classified between examples (i.e. instances which denote the patterns) and counter examples (i.e. instances which do not denote the patterns) of the patterns, and (3) from these labeled examples EAGLE produces descriptions which are interpreted as constraints refining the patterns.

The remainder of the paper is organized as follows. Section 2 presents a description of the information extraction system PROMÉTHÉE. Section 3 describes the inductive machine learning system EAGLE. Next, section 4 presents the interfacing between PROMÉTHÉE and EAGLE systems. Section 5 presents and evaluates results obtained on patterns of the hyponymy relation. Section 6 discusses related work in applying symbolic machine learning to information extraction. Finally, section 7 suggests future work and concludes the paper.

Chapter 2

The PROMÉTHÉE System

In the last few years, several information extraction systems have been developed to extract patterns from text. AutoSlog (Riloff, 1993, 1996) creates a dictionary of extraction patterns by specializing a set of general syntactic patterns. CRYSTAL (Soderland et al., 1995) is another system that generates extraction patterns dependant on domain-specific annotations. LIEP (Huffman, 1995) also learns extraction patterns, but relies on predefined keywords, a sentence analyzer to identify noun and verb groups, and an entity recognizer to identify entities of interest (people, company names, and management titles).

Our approach to extract patterns is based on a different technique which makes no hypothesis about the data to be extracted. The information extraction system PROMÉTHÉE uses only pairs of terms linked by the target relation to extract specific patterns, but relies on part-of-speech tag, and on local grammars. For instance, Figures 2.1 and 2.2 show two sentences from [MEDIC] corpus[1], and the corresponding pattern for two different relations.

SENTENCE

```
we measured the levels of asparate, glutamate, gamma-aminobutyric acid,
and other amino acids in autopsied brain of 6 patients.
```

PATTERN (simplify)

```
NP {, NP}* and other NP
```

RELATION

```
HYPONYM(asparate,amino acid)
HYPONYM(glutamate,amino acid)
HYPONYM(gamma-aminobutyric acid,amino acid)
```

Figure 2.1: Syntactic pattern for the hyponymy relation.

[1]All the experiments reported in this paper have been performed on [AGRO]: a 1.3-million words French agronomy corpus and on [MEDIC]: a 1.56-million words English medical corpus. These corpus are composed of abstracts of scientific papers owned by INIST-CNRS.

SENTENCE

```
Dermal hypoplasia is a rare ectomesodermal dysplasia syndrome
characterized by cutaneous, skeleta, dental, ocular, and soft-tissue
defects.
```

PATTERN (simplify)

```
NP (disorder|disease|syndrome|...)  characterized by NP, {NP,}* (and|or)
NP
```

RELATION

```
CHARACTERIZED_BY(ectomesodermal dysplasia syndrome,cutaneous defect)
CHARACTERIZED_BY(ectomesodermal dysplasia syndrome,skeleta defect)
CHARACTERIZED_BY(ectomesodermal dysplasia syndrome,dental defect)
CHARACTERIZED_BY(ectomesodermal dysplasia syndrome,ocular defect)
CHARACTERIZED_BY(ectomesodermal dysplasia syndrome,soft-tissue defect)
```

Figure 2.2: Syntactic pattern for the characterized by relation.

2.1 Overview of the PROMÉTHÉE Architecture

As illustrated in Figure 2.3, the PROMÉTHÉE architecture is divided into three main modules:

1. *Lexical Preprocessor.* This module begins by reading the raw text. The text is divided into sentences which are individually tagged[2]. Noun phrases, acronyms, and a succession of noun phrases are detected by using regular expressions[3]. The output is formated under the SGML (Standard Generalized Markup Language) formalism.

2. *Lexico-syntactic Analyzer.* This module extracts lexico-syntatic patterns modeling a semantic relation. Patterns are discovered by looking through text, and by using a bootstrap of pairs of terms linked by the target relation. This procedure which consists of 7 steps (see Figure 2.4) is described in the next section.

3. *Conceptually Relationship Extractor.* This module extracts pairs of conceptually related terms by using a database of patterns, which can be the output of the lexico-syntactic analyzer or manually specified.

2.2 Lexico-syntactic Analyzer

The lexico-syntactic analyser discovers new patterns by looking through text. This procedure is composed of 7 steps (see figure 2.4).

1. Select manually a representative conceptual relation, *e.g.* the hyponymy relation.

[2]We thanks Évelyne Tzoukermann (Bell Laboratories, Lucent Technologies) for having tagged and lemmatized the corpus [AGRO].

[3]All of these preprocess components are implemented with Perl.

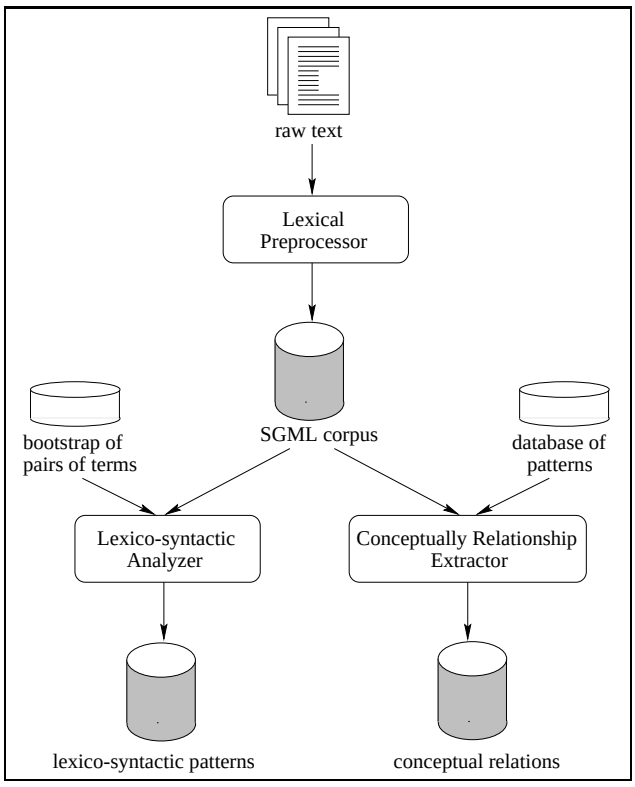

Figure 2.3: The PROMÉTHÉE system.

2. Collect a list of pairs of terms linked by the previous relation. This list of pairs of terms can be extracted from a thesaurus, a knowledge base or manually specified. For example, from a medical thesaurus (UML, 1994) and the hyponymy relation, we find that *glutamate* IS-A *amino acid*.

3. Find sentences where conceptually related terms occur. Thus, the pair (*glutamate*,*amino acid*) allows us to extract from the corpus [MEDIC] the sentence: *we measured the levels of asparate, glutamate, gamma-aminobutyric acid, and other amino acids in autopsied brain of 6 patients.*

4. Find a common environment that generalizes the sentences extracted at the third step. This environment indicates a candidate lexico-syntactic pattern.

5. Validate candidate lexico-syntactic patterns by an expert.

6. Use new patterns to extract more pairs of candidate terms.

7. Validate candidate terms by an expert, and go to step 3.

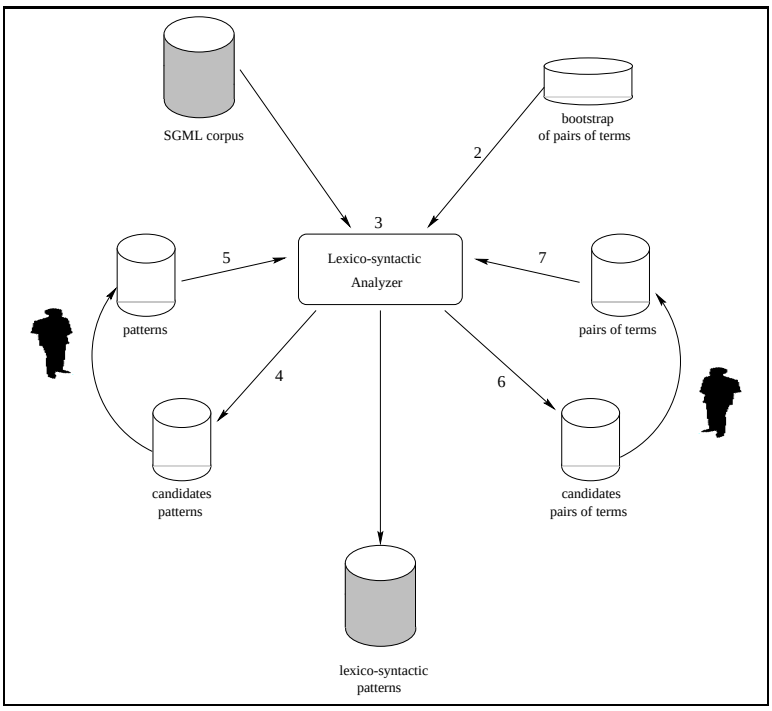

Figure 2.4: The lexico-syntactic analyser.

2.3 Lexico-syntactic Expression and Patterns

At the third step, a set of sentences is extracted. These sentences are lemmatised, and noun phrases are identified. So, we represent a sentence by a lexico-syntactic expression. For example, the previous relation HYPONYM(*neocortex*,*vulnereable area*) allows to extract from the corpus [MEDIC] the sentence:

> *Neuronal damage were found in the selectively vulnerable areas such as neocortex, striatum, hippocampus and thalamus*

From this sentence, we produce the lexico-syntactic expression:

NP be find in NP such as LIST[4]

A lexico-syntactic expression is composed of a set of elements, which can be either lemmas, punctuation marks, numbers, symbols (e.g. §, <, π, etc.) or words with specific part of speech tags, such as NP, LIST, CRD, etc. Through this simplification process, we have a more generic representation of relevant sentences, and comparing these sentences is easier.

A lexico-syntactic patterns is the generalization for a set of lexico-syntacic expressions. For example, with the previous expression, and at least another similar one, the following lexico-syntactic pattern is deduced (Morin, 1998):

NP such as LIST

2.4 Limitations of this technique

Using this technique some lexico-syntactic patterns are extracted. However, these patterns are too general: indeed without using manual constraints, their *Recall* is satisfying but *Precision* is low. They do not prevent the extraction of pairs of terms which are not linked by the target relation. The low *Precision* can be explain by general patterns which cover a set of more rarely specific patterns.

Thus, a refinement of these patterns is necessary. In order to improve the low *Precision* of general patterns, we propose to use the learning system EAGLE.

Chapter 3

Learning Concepts Descriptions with EAGLE

EAGLE is a system dedicated to the symbolic learning of concepts. In this framework, a concept is viewed as a class of objects which share common properties and have the same behavior, for instance mammal, mollusc, polygon, etc.

An *extensional description* of such a concept consists in collections of objects which belong to it, i.e. its examples, or those that do not, i.e. its counter-examples. By contrast, an *intensional description* states the peculiar properties of the concept, which allows to distinguishing it from others. For instance, the concept even number can be described extensionally by the following sets of examples and counter-examples: $\{2, 4, 6, 8, 10\}$ and $\{1, 3, 5\}$, and intensionally by the following statement: *"an even number is an integer which can be divided by 2"*.

From a semantic point of view, an intensional description is more powerful than an extensional one in the sense that it formulates a general definition of a concept, instead of specifying collections of objects which are often incomplete to extract the concept's own properties. The purpose of an intensional definition is to allow the recognition of any object as a member or not of the concept it defines. For instance, by using the previous description of the concept even number (*"an even number is an integer which can be divided by 2"*), it is possible to deduce that 12 is also an even number, which is not the case by only examining both sets $\{2, 4, 6, 8, 10\}$ and $\{1, 3, 5\}$.

The EAGLE system is based on the *learning from examples* paradigm, also called *induction*, which consists in extracting intensional descriptions of target concepts, from their extensional descriptions as well as prior knowledge about the given domain (Michell, 1997). This latter specifies general information, such as the objects features, relationships and so forth.

The induction task is illustrated on Figure 3.1, through a simple problem of learning a description for the concept of the semantic relationship SUPER-HYPONYM (i.e. the hyponym of a hyponym). The information which is provided consists in some concrete examples and counter-examples of the SUPER-HYPONYM relationship (*"BAREY is a SUPER-HYPONYM of PLANT PRODUCTS"*, *"ETHANOL is not a SUPER-HYPONYM of METHANOL"*, and so forth), as well as additional knowledge about existing HYPONYM relationships (*"OATS is*

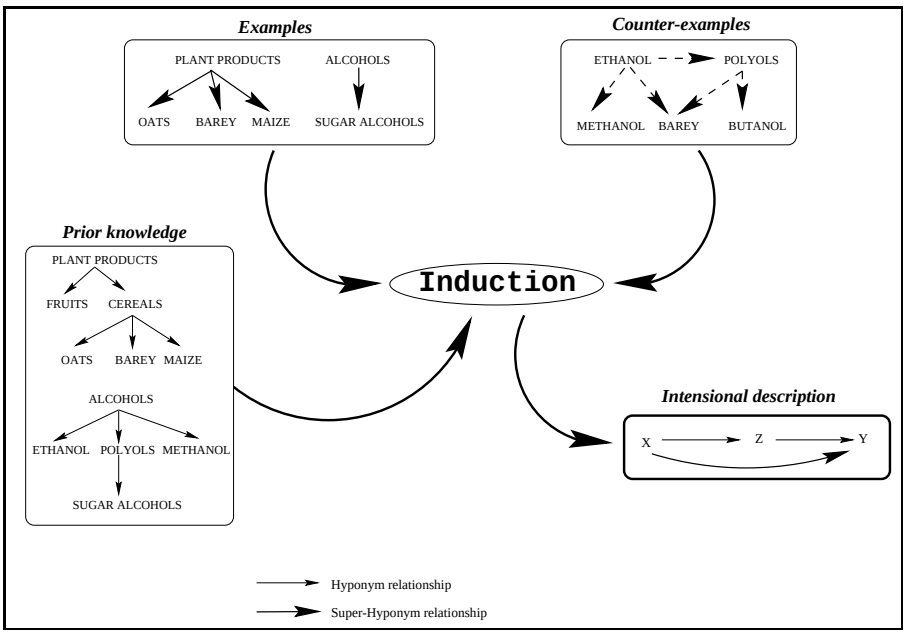

Figure 3.1: Learning concepts from examples.

a HYPONYM of CEREALS", "SUGAR ALCOHOLS is a HYPONYM of POLYOLS", and so forth). From these information, EAGLE aims at extracting an intentional description which generalizes the SUPER-HYPONYM relationship. On the example, a suitable description would specify that *"any entity X is a SUPER-HYPONYM of another Y, provided there exists a third entity Z such that the two following conditions hold: (1) X is a HYPONYM of Z and (2) Z is a HYPONYM of Y"*.

In order to translate the problem of induction into computational terms, it is necessary to choose a representation language for examples and background knowledge. The language used in EAGLE is the First Order Logic: various predicates stand for concepts, properties and relationships, which are described extensionally by n-ary relational tuples. For instance, the translation of the previous family example into First Order Logic leads to the following tuples:

$$E^+ = \begin{cases} \mathsf{SUPER-HYPONYM}(OATS, PLANT\ PRODUCTS) \\ \mathsf{SUPER-HYPONYM}(MAIZE, PLANT\ PRODUCTS) \\ \mathsf{SUPER-HYPONYM}(SUGAR\ ALCOHOLS, ALCOLOLS) \\ etc. \end{cases} \tag{3.1}$$

$$E^- = \begin{cases} \mathsf{SUPER-HYPONYM}(BAREY, ETHANOL) \\ \mathsf{SUPER-HYPONYM}(BUTANOL, POLYOLS) \\ etc. \end{cases} \tag{3.2}$$

$$P = \begin{cases} \mathsf{HYPONYM}(FRUITS, PLANT\ PRODUCTS) \\ \mathsf{HYPONYM}(CEREALS, PLANT\ PRODUCTS) \\ \mathsf{HYPONYM}(OATS, CEREALS) \\ etc. \end{cases} \tag{3.3}$$

E^+ and E^- describe respectively examples and counter-examples of the SUPER-HYPONYM target concept, whereas P is the available prior knowledge including the HYPONYM relationships between the entities.

Intensional descriptions which are induced by EAGLE are sets of Horn clauses of the form: *Conclusion* $\longleftarrow$ *Condition*. The *Condition* part of a clause specifies a conjunction of properties which must be satisfied by objects, or tuples of objects so that they belong to the concept in the *Conclusion* part. For instance, a description in First Order Logic for the SUPER-HYPONYM concept would be:

$$\mathsf{SUPER-HYPONYM}(X, Y) \longleftarrow \mathsf{HYPONYM}(X, Z) \wedge \mathsf{HYPONYM}(Z, Y) \tag{3.4}$$

This means that a tuple of objects (X,Y) is a member of the SUPER-HYPONYM concept, if and only if there exist two tuples (X,Z) and (Z,Y) which belong to the HYPONYM relationship (each variable X, Y and Z stands for a single object).

To achieve the learning goal, the inductive learning approach developed in EAGLE is based on Rough Set Theory, and more especially on its notion of concept approximation. A learning process thus comprises three steps, namely (1) partitioning of the knowledge, (2) approximation of the target concept, and finally (3) induction of a suitable description. The resulting definition must be complete, i.e. characterize all examples, but no counter-example of the target concept[1].

[1]For more information on the EAGLE, in particular a complete description of the learning process, see the following related publication: Martienne and Quafafou (1998).

Chapter 4

Interfacing Prométhée with Eagle

The goal of interfacing Prométhée with Eagle is to use the latter as a tool for refining specific pattern (see Figure 4.1). Thus, Eagle fits between the steps 5 and 6 of the previous methodology (see section 2.2).

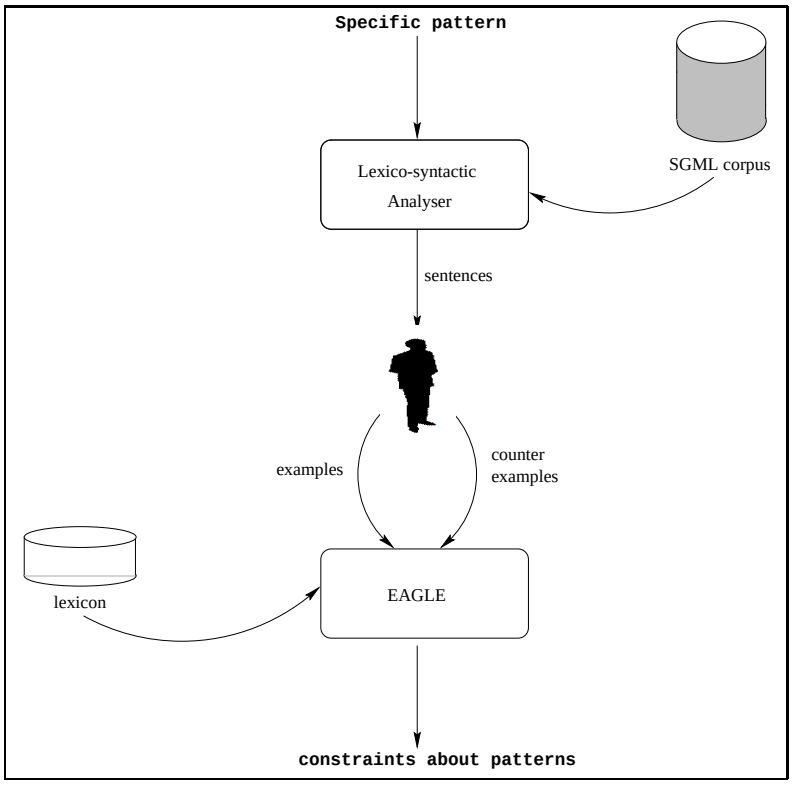

Figure 4.1: Interfacing Prométhée with Eagle.

For a specific pattern, the lexico-syntactic analyzer extracts sentences from the Sgml corpus. An expert classified these sentences between examples (i.e. sentences where pairs of terms are conceptually related) and counter-examples (i.e. sentences where pairs of terms are not

conceptually related). From this extensional description of the patterns and the prior knowledge consisting of a lexicon, the EAGLE system extracts an intensional description which refines the specific pattern (i.e. syntactic or logic constraints).

Interfacing the two systems requires the translation of PROMÉTHÉE's lexico-syntactic analyzer output sentences, into EAGLE's logic-based formalism (see Figure 4.2). Here, a sentence is basically viewed as a lexico-syntactic expression including two main conceptually related noun phrases called SN1 and SN2.

In EAGLE, the representation of such a sentence consists in describing how it is organized around SN1 and SN2, i.e. which terms precede or follow them, together with the corresponding separation depths. Given a noun phrase and a particular element in the sentence, the depth is defined here as the distance, i.e. the number of elements, which separate the noun phrases from the given element. Two predicates are used to describe the sentences, namely;

- $\mathsf{Pred}(X, Y, Z, T)$ to state that in the sentence X, the noun phrase Y (whose value can be either SN1 or SN2) is preceded by the element Z, at a depth equal to T,
- $\mathsf{Succ}(X, Y, Z, T)$ to state that in the sentence X, the noun phrases Y (whose value can be either SN1 or SN2) is followed by the element Z, at a depth equal to T.

Additional predicates are used to indicate the part of speech tags of the terms in the lexicon:

- $\mathsf{Verb}(X)$ to state that the term X is a verb,
- $\mathsf{Adj}(X)$ to state that the term X is an adjective,
- $\mathsf{Crd}(X)$ to state that the term X is a cardinal number, etc.

Sentence

```
002104.  we measured the levels of asparate, glutamate,
gamma-aminobutyric acid, and other amino acids in autopsied brain of
6 patients.
```

Prométhée's output

```
002101 PRO measure the level of SN2 comma; and other SN1 in SN of CRD SN
period;
HYPONYMY(SN1,SN2)
SN1→SN
SN2→LIST_3
```

Eagle's formalism

```
Pred(num_002104,SN2,BEGIN,p_6)
Pred(num_002104,SN2,PRO,p_5)
Pred(num_002104,SN2,measure,p_4)
Pred(num_002104,SN2,the,p_3)
Pred(num_002104,SN2,level,p_2)
Pred(num_002104,SN2,of,p_1)
Succ(num_002104,SN2,comma;,p_1)
Succ(num_002104,SN2,and,p_2)
Succ(num_002104,SN2,other,p_3)
Succ(num_002104,SN1,in,p_1)
Succ(num_002104,SN1,SN,p_2)
Succ(num_002104,SN1,of,p_3)
Succ(num_002104,SN1,CRD;,p_4)
Succ(num_002104,SN1,SN;,p_5)
Succ(num_002104,SN1,period,p_6)
Succ(num_002104,SN1,END,p_7)

Crd(CRD)
Ponctuation(comma;)
Ponctuation(period;)
Prep(of)
Pronoun(PRO)
Verb(measure)
...
```

Figure 4.2: Translation of PROMÉTHÉE's output into EAGLE's formalism.

Chapter 5

Experimental Results

In this experimentation, we have focused on the hyponymy relation[1]. For this relation, PROMÉTHÉE incrementally extracted 11 lexico-syntactic patterns from the corpus [AGRO]. We are particularly interested in two of them, namely: NP comme LIST (NP such as LIST in English), and NP (LIST), which model respectively enumeration and exemplification structures (Borillo, 1996). Some sentences instantiating these patterns were then produced from a 43 000 sentences corpus [AGRO], and split into examples and counter-examples.

5.1 Enumeration Structure Pattern

Among the 36 sentences instantiating the pattern NP comme LIST, the expert retained a sample of 28 sentences which denoted a hyponymy relation, i.e. the examples, and 8 sentences which did not, i.e. the counter-examples. In a first experimentation, constraints were induced by using whole prior knowledge. But the resulting constraints were not satisfying in the sense that they focused on tool words (e.g. preposition, article, etc.). In order to improve the results, some predicates regarding them have been ignored from the prior knowledge. The constraints which were learnt from the next experimentation can be split into two main categories: (1) the hyperonym term can be preceded by an undefined adjective, such as *différents* (*different*), *certains* (*some*) and *d'autres* (*others*), and (2) the hyperonym term can be preceded by the expression *chez d'autres*. It appears that sentences matching these constraints have a high level of reliability, and do not require validation by a expert. This is illustrated by the Table 5.1.

Before learning, the pattern NP comme LIST is too general, since its precision is equal to 77.7%[2]. As a consequence all the 36 matching sentences must be manually validated. After learning, two patterns have a precision of 100.0%, which allows us to remove the matching sentences from the manual validation. Consequently, only 28 matching sentences must be manually validated. With these new constraints, 33.3% ((10/36)×100) matching sentences are automatically acquired.

[1]According to Hearst (1992), a lexical term L_0 is said to be a hyponym of the concept represented by a lexical item L_1 if native speakers of English accept sentences constructed from the frame *An* L_0 *is a* (*kind of*) L_1. Here, L_0 (resp. L_1) is the hyponym (resp. hypernym) of L_1 (resp. L_0).

[2]The precision of a pattern is calculated by the following ratio: ♯ good sentences / ♯ matching sentences.

	Pattern	Matching sentences	Good sentences	False sentences
Before learning	NP comme LIST	36	28	8
After learning	chez d'autres NP comme LIST	2	2	0
	{certains|différents|d'autres|...} NP comme LIST	8	8	0
	NP comme LIST	26	18	8

Table 5.1: Enumeration structure patterns accuracies before and after learning process.

5.2 Exemplification Structure Pattern

Among the 603 sentences instantiating the pattern NP (LIST), the expert retained a sample of 21 sentences which denoted a hyponymy relation, i.e. the examples, and 16 sentences which did not, i.e. the counter-examples. As in the previous experimentation, some restrictions have been applied in the prior knowledge. Here, two categories of constraints have been acquired: (1) as previously the hyperonym term can be preceded by an undefined adjective, and (2) the cardinal before the hyperonym term must be equal to the number of elements of the list LIST. This is illustrated by the Table 5.2.

Before learning the precision of the pattern NP (LIST) is equal to 56.8% on 37 matching sentences. Once again, learning allows to decrease the number of matching sentences to be manually validated (i.e. 27 vs 37). With these specific constraints, 27% ((10/37)×100) matching sentences are automatically acquired.

	Pattern	Matching sentences	Good sentences	False sentences
Before learning	NP (LIST)	37	21	16
After learning	NP (LIST)	27	11	16
	{certains|différents|d'autres|...} NP (LIST)	4	4	0
	CRD1 NP (LIST-CRD2)	6	6	0
	CRD1 = CRD2			

Table 5.2: Exemplification structure patterns accuracies before and after learning process.

Chapter 6

Related Work

Previous research applying learning methods in Natural Language Processing has been devoted to learning syntactic patterns, such as noun phrases (Ramshaw and Marcus, 1995; Argamon et al., 1998), name phrases (Vilain and Day, 1996), or specific-domain patterns (Soderland et al., 1995; Riloff, 1993, 1996; Huffman, 1995; Califf and Mooney, 1997). Machine learning has the potential to significantly assist the acquisition of lexico-syntactic patterns.

Several information extraction systems dedicated to acquisition of patterns, are based on the use of machine learning techniques. AutoSlog (Riloff, 1993) system uses a training corpus to generate candidate patterns, and rely on an expert to verify and reject each candidate pattern. Crystal (Soderland et al., 1995) is one of the first systems to automatically induce a dictionary of information extraction rules by, generalizing patterns identified in the text by an expert. However, a training corpus is not often available for most information extraction tasks. The Rapier (Califf and Mooney, 1997) system uses relational learning to construct unbounded pattern-match rules. Liep (Huffman, 1995) learns information extraction patterns from example texts containing events. A user can choose which combinations of entities signify events to be extracted. These positive examples are used by Liep to build a set of extraction patterns. The general methodology is similar to Eagle's, but Prométhée, like AutoSlog, does not try to recognize relationships between multiple constituents.

Eagle system is used by Prométhée system only to provide more information about sentence instantiating patterns. Thus, it is involved only in a small part of the acquisition process. Consequently, few training examples are needed to produce syntactical constraints, on order of forty rather than hundreds or thousands, to achieve good perfomance. Moreover, the constraints produced by Eagle provide logical and syntactical information about lexico-syntactic-patterns. This is not the case of other systems only extract syntactical information.

Chapter 7

Conclusion and Future Work

In this paper, we have proposed an approach for refining lexico-syntactic patterns, based on the use of a machine learning tool. This technique interfaces an information extraction system PROMÉTHÉE with an inductive logic programming system EAGLE, which allows for refining the lexico-syntactic patterns produced by PROMÉTHÉE.

The empirical results obtained with this technique shows that the refined patterns allows to decrease the need for the human validation.

From a Natural Language Processing point of view, the use of a machine learning technique highlights some knowledge which usually required manual data mining. From a Machine Learning point of view, it illustrates the usefulness of an inductive learning technique on a real-world problem.

In future work, we plan to investigate the usefulness of EAGLE to extract constraints by using PROMÉTHÉE's syntactical and morphological information which allowed to generate lexico-syntactic expressions.

Bibliography

Shlomo Argamon, Ido Dagan, and Yuval Krymolowski. A memory-based approach to learning shallow natural language patterns. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING-ACL'98)*, page to appear, Montreal, Canada, 1998.

Andrée Borillo. Exploration automatisée de textes de spécialité : repérage et identification de la relation lexicale d'hyperonymie. *LINX*, 34/35:113–124, 1996.

Mary Elaine Califf and Raymond J. Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the Computational Natural Language Learning (CoNLL'97)*, pages 9–15, Madrid, Spain, July 1997.

Marti A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*, pages 539–545, Nantes, France, 1992.

Jerry R. Hobbs, Douglas E. Appelt, John S. Bear, David J. Israel, and W. Mabry Tyson. FASTUS : A system for extracting information from natural language text. Technical Report 515, SRI International, Menlo Park, CA, november 1992.

Scott B. Huffman. Learning information extraction patterns from examples. In *Workshop New Approaches to Learning for Natural Language Processing at IJCAI'95*, pages 127–133, Montreal, 1995.

Emmanuelle Martienne and Mohamed Quafafou. Learning Logical Descriptions for Document Understanding: a Rough Sets-based Approach. In *Proceedings of the first International Conference on Rough Sets and Current Trends in Computing (RSCTC'98)*, pages 22–26, Warsaw, Pologne, june 1998.

Tom M. Michell. *Machine Learning*. McGraw-Hill, New York, 1997.

Emmanuel Morin. PROMÉTHÉE un outil d'aide à l'acquisition de relations sémantiques entre termes. In *Actes, 5th National Conference on Traitement Automatique des Langues Naturelles (TALN'98)*, pages 172–181, Paris, France, june 1998.

MUC-3. *Proceedings of the third Message Understanding Conference*. Morgan Kauffmann, San Diego, CA, 1991.

MUC-4. *Proceedings of the fourth Message Understanding Conference*. Morgan Kauffmann, San Mateo, CA, 1992.

MUC-5. *Proceedings of the fifth Message Understanding Conference.* Morgan Kauffmann, San Mateo, CA, 1993.

MUC-6. *Proceedings of the sixth Message Understanding Conference.* Morgan Kauffmann, Columbia, Maryland, 1995.

Stephen Muggleton. Inductive logic programming. *New Generation Computing*, 8:295–318, 1991.

National Library of Medicine. *Unified Medical Language System, UMLS Knowledge Source*, 1994.

Lance A. Ramshaw and Mitchell P. Marcus. Text Chunking using Transformation-Based Learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 811–816, 1995.

Ellen Riloff. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI'93)*, pages 811–816, Menlo Park, CA, USA, July 1993.

Ellen Riloff. Automatically generating extraction from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI'96)*, pages 1044–1049, august 1996.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. CRYSTAL: Inducing a Conceptual Dictionay. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1314–1319, august 1995.

Marc Vilain and David Day. Finite-state phrase parsing by rule sequences. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, pages 274–279, Copenhagen, Denmark, 1996.