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

Emmanuel Morin and Emmanuelle Martienne

Institut de Recherche en Informatique de Nantes (IRIN)
Université de Nantes
2, rue de la Houssinière - BP 92208
44322 Nantes Cedex 03 - France
{morin,martien}@irin.univ-nantes.fr

Abstract. Acquisition of patterns for information extraction systems is a common task in Natural Language Processing, mostly based on manual analysis of text corpora. We have developed a system called PROMÉTHÉE, which incrementally extracts lexico-syntactic patterns for a specific conceptual relation from a technical corpus. However, these patterns are often too general and need to be manually validated.
In this paper, we demonstrate how PROMÉTHÉE has been interfaced with the machine learning system EAGLE in order to automatically refine the patterns it produces. The empirical results obtained with this technique show that the refined patterns allows to decrease the need for the human validation.

1 Introduction

As the amount of electronic documents (corpora, dictionaries, newspapers, newswires, etc.) become more and more important and diversified, there is a need to extract information automatically from 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 at making sense of the entire text, but are only focused on fractions of the text that are relevant to a specific domain [6]. In information extraction, the data to be extracted from a text is given by a *syntactic pattern*, also called a *template*, which typically involves recognizing a group of entities, generally noun phrases, and some relationships between these entities.

In recent years, through Message Understanding Conferences, several information extraction systems have been developed for a variety of domains. However, many of the best-performing systems are difficult and time-consuming to build. They also generally contain domain-specific components. Therefore, their success is often tempered by their difficulties to adapt to new domains. Having the use of specialists' abilities for each domain is not reasonable.

* We would like to thank C. Jacquemin and M. Quafafou for helpful discussions on this work.

R. López de Mántaras, E. Plaza (Eds.): ECML 2000, LNAI 1810, pp. 292–299, 2000.
© Springer-Verlag Berlin Heidelberg 2000

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 [10]. However, based on our experience, we believe that such patterns are too general: indeed, without using manual constraints, their coverage is satisfying but their precision[1] is low. In order to refine these patterns, we propose to use a learning system, called EAGLE [8], which is based on the *Inductive Logic Programming* paradigm [11]. This latter extracts intensional descriptions of concepts, from their extensional descriptions including their ground examples and counter-examples, as well as a prior knowledge of the domain. The learned definitions, expressed in a logic-based formalism, are further used in recognition or classification tasks.

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

2 The Prométhée System

In the last few years, several information extraction systems have been developed to extract patterns from text. AutoSlog [13,14] creates a dictionary of extraction patterns by specializing a set of general syntactic patterns. CRYSTAL [15] is another system that generates extraction patterns dependant on domain-specific annotations. LIEP [7] 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, the following sentence of the [MEDIC] corpus[2]: *we measured the levels of asparate, glutamate, gamma-aminobutyric acid, and other amino acids in autopsied brain of 6 patients* contains a pair of terms, namely *(asparate,amino*

[1] The precision of a pattern is the percentage of sentences matching the pattern which really denote the conceptual relation modeled by this pattern.

[2] 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.

acid), linked by the hyponymy[3] relation. From this sentence, the following pattern modeling the relation is extracted: NP {, NP}* and other NP[4].

2.1 Overview of the Prométhée Architecture

The PROMÉTHÉE architecture is divided into three main modules:

1. *Lexical Preprocessor.* This module starts by reading the raw text. The text is divided into sentences which are individually tagged[5], *i.e.* noun phrases, acronyms, and a succession of noun phrases are detected by using regular expressions. 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 from the SGML corpus. Patterns are discovered by looking through the corpus, and by using a bootstrap of pairs of terms linked by the target relation. This procedure which consists of 7 steps 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 either the output of the lexico-syntactic analyzer or manually specified patterns.

2.2 Lexico-syntactic Analyzer

The lexico-syntactic analyser extracts new patterns by looking through a SGML corpus. This procedure, inspired by Hearst [4,5], is composed of 7 steps.

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

[3] According to [9], 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).

[4] NP is part of speech tag for a noun phrase.

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

2.3 Lexico-syntactic Expressions and Patterns

At the third step of the lexico-syntactic analyzer, 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 instance, the following element of the hyponymy relation: *(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[6].

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 pattern is a generalization of a set of lexico-syntactic expressions. For example, with the previous expression, and at least another similar one, the following lexico-syntactic pattern is deduced [10] : 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 coverage is satisfying but their precision is low. The low precision can be explain by general patterns which cover a set of more rarely specific patterns. Too general patterns do not prevent the further extraction of pairs of terms which are not linked by the target relation. At present, a human validation (the step 5 of the lexico-syntactic analyzer procedure) is necessary to exclude the patterns which are considered as too general. Through the interfacing of Prométhée and Eagle, we aim at automatically acquiring some knowledge refining these patterns, in order to decrease the need of human validation.

3 Interfacing Prométhée with Eagle

The goal of interfacing Prométhée with Eagle is to use the latter as a tool for refining too general patterns. Thus, Eagle fits between the steps 5 and 6 of the previous methodology (see Section 2.2).

For a specific pattern, the lexico-syntactic analyzer extracts sentences from the Sgml corpus. An expert classifies 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 some intensional descriptions of these patterns. Interpreted as syntactic or logic constraints on the general form of the

[6] LIST is part of speech tag for a succession of noun phrases.

patterns, these descriptions allow to refine them and to decrease the need for human validation.

Interfacing the two systems requires the translation of PROMÉTHÉE's lexico-syntactic analyzer output sentences into EAGLE's logic-based formalism. Here, a sentence is basically viewed as a lexico-syntactic expression including two main conceptually related noun phrases called NP1 and NP2. In EAGLE, the representation of such a sentence in the prior knowledge consists in describing, by means of predicates, how it is organized around NP1 and NP2 , *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. Additional predicates are used in the prior knowledge to indicate the part of speech tags (verb, adjective, etc) of the terms in the lexicon.

4 Experimental Results

In this experimentation, we have focused on the hyponymy relation. 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 exemplification and enumeration structures [2]. Some sentences instantiating these patterns were produced from a 43,000 sentences corpus [AGRO], and split into examples and counter-examples. The following clause $\mathsf{Pattern}(x) \longleftarrow \mathsf{Succ}(x, \mathsf{NP1}, y, z) \wedge \mathsf{Crd}(y)$ is an example of the results produced by EAGLE. It defines a constraint according to which a pattern x models an hyponymy relation if (1) its noun phrase NP1 is followed by a term y at a depth equal to z, and (2) y is a cardinal number.

4.1 Exemplification 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 the whole prior knowledge associated with the 36 sentences. 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 tool words have been ignored from the prior knowledge. The constraints which were learned 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 on Table 1.

Before learning, the pattern NP comme LIST is too general, since its precision is equal to 77.7%. As a consequence, all the 36 matching sentences must be

Table 1. Exemplification structure patterns accuracies before and after learning process

	Pattern	Matching sent.	Good sent.	False sent.
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

manually validated. After learning, two patterns have a precision of 100.0%, which allows to remove the matching sentences from the manual validation. Consequently, only 26 matching sentences must be manually validated. With these new constraints, around 27% (100-(26/36)*100) of matching sentences are automatically acquired.

4.2 Enumeration 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 on Table 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). Again, with these specific constraints, around 27% (100-(27/37)*100) of matching sentences are automatically acquired.

Table 2. Enumeration structure patterns accuracies before and after learning process

	Pattern	Matching sent.	Good sent.	False sent.
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			

5 Related Work

Previous research involving Machine Learning methods and Natural Language Processing has been devoted to the learning of syntactic patterns, such as noun phrases [12,1], name phrases [16], or specific-domain patterns [15,13,14,7,3]. Machine learning has the potential to significantly assist the acquisition of lexico-syntactic patterns.

Several information extraction systems, dedicated to the acquisition of patterns, are based on the use of machine learning techniques. AutoSlog [13] system uses a training corpus to generate candidate patterns, and rely on an expert to verify and reject each candidate pattern. Crystal [15] 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 [3] system uses relational learning to construct unbounded pattern-match rules. Liep [7] 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 the Prométhée system only to provide more information about the general forms of the patterns. Thus, it is involved only in a small part of the acquisition process. Consequently, few training examples are needed to produce syntactical constraints : around forties are enough to achieve good perfomance, rather than hundreds or thousands. Moreover, the constraints produced by Eagle provide some readable logical and syntactical information about lexico-syntactic-patterns. This is not the case of other systems only extract syntactical information.

6 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 show 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.

References

1. 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)*, Montreal, Canada, 1998. 298
2. Andrée Borillo. Exploration automatisée de textes de spécialité : repérage et identification de la relation lexicale d'hyperonymie. *LINX*, 34/35:113–124, 1996. 296
3. 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. 298
4. 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. 294
5. Marti A. Hearst. Automated Discovery of WordNet Relations. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 131–151. MIT Press, Cambridge, MA, 1998. 294
6. 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. 292
7. 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. 293, 298
8. 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. 293
9. George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Miller Katherine. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3:235–244, 1990. 294
10. Emmanuel Morin. *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques*. Thèse en informatique, Université de Nantes, December 1999. 293, 295
11. Stephen Muggleton. Inductive logic programming. *New Generation Computing*, 8:295–318, 1991. 293
12. 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. 298
13. 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. 293, 298
14. 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. 293, 298
15. 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. 293, 298
16. 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. 298