Information Extraction, Multilinguality and Portability

Jordi Turmo

TALP Research Center
Universitat Politècnica de Catalunya
c/ Jordi Girona, 1-3 , Barcelona, España.
turmo@lsi.upc.es

Abstract

The growing availability of on-line textual sources and the potential number of applications of knowledge acquisition approaches from textual data, such as Information Extraction (IE), has lead to an increase in IE research. Some examples of these applications are the generation of data bases from documents, as well as the acquisition of knowledge useful for emerging technologies like question answering and information integration, among others related to text mining. However, one of the main drawbacks of the application of IE refers to the intrinsic language and domain dependence. For the sake of reducing the high cost of manually adapting IE applications to new domains and languages, different Machine Learning (ML) techniques have been applied by the research community. This survey describes and compares the main approaches to IE and the different ML techniques used to achieve adaptable IE technology, as of today.

Keywords: Information Extraction, Multilingual Information Extraction, Machine Learning for Information Extraction.

1 Introduction

Traditionally, information involved in Knowledge-Based systems has been manually acquired in collaboration with experts on the subject of study. However, both the high cost of such a process and the existence of textual sources containing the required information have led to the use of automatic acquisition approaches. In the early eighties, Text-Based Intelligent (TBI) systems began to be used so as to automatically obtain the desired information by manipulating documents. These documents are usually highly structured when produced for computing use, and the process of extracting information from them can be carried out quite straightforwardly. However, documents have been frequently produced for human use and they lack an explicit structure. In such cases, they consist of an unrestricted Natural Language (NL) text, and the task of extracting information involves a great deal of linguistic knowledge. Sometimes, documents present a semi-structured style, such as on-line documents, where both chunks of NL text and structured pieces of information (e.g., meta-data) appear together. Roughly speaking, two major TBI areas have to be considered and distinguished: Information Retrieval (IR) and Information Extraction (IE). IR techniques are used to search documents from a collection that comply with the restrictions of a query, being these traditionally a list of keywords. As a consequence, IR techniques allow the retrieval of relevant documents according to the query, which may be used for the information acquisition process. The role of Natural Language Processing (NLP) techniques in IR tasks is controversial and, in any case,

marginal. The reader may find more detailed explanations about IR techniques in [Grefenstette, 1998; Strzalkowski, 1999; Baeza-Yates and Ribeiro-Neto, 1999].

IE technology arose as a more in-depth understanding task. The extraction tasks are clearly more difficult than the retrieval ones. While in IR the answer to a query is simply a list of potentially relevant documents, in IE the relevant content of such documents has to be located and extracted from the text. This relevant content, represented in a specific format, can be integrated within knowledge-based systems, as well as used within IR to allow more accurate responses. Some emerging technologies, such as Answer Finding and Question Answering, try to overcome such differences and take advantage from both IR and IE techniques (c.f., [Berger et al., 2000; Vilcedo, 2002]).

In order to deal with the IE difficulty, NLP is no longer limited to the text control step, as it occurs in IR, but is widely used in other steps of the extraction process, depending on the document style to be dealt with. Statistical methods, although present in many of the NL components of IE systems, are no longer the core of the procedure, and knowledge-based approaches are needed in several tasks involved. Besides, the definition of what content is relevant to be extracted is decided *a priori*. This fact implies a clear domain dependence of IE technology, leading to portability drawbacks that are present in most IE systems. When dealing with new domains, new specific knowledge is needed and has to be acquired by such systems.

This paper is organized as follows. Section 2 briefly describes the problem that IE deals with, with an example. Section 3 describes the historical framework in which IE systems have been developed. Within such a framework, the general architecture of IE systems is described in section 4, while extensions to support multilingual information access are presented in section 5. The complexity of IE systems and their intrinsic domain dependence make it difficult for them to be accurately applied to any situation. In order to reduce such drawbacks, empirical methods in NLP have been used, as described in section 6. Finally, a classification of different state-of-the-art IE systems is presented from two different points of view in section 7.

2 The goal of IE

IE defines the problem of extracting pieces of information related to a prescribed set of related concepts, namely *scenario of extraction*, from restricted-domain documents. As an example, let us consider the following scenario of extraction related to the domain of mushrooms:[1]

> A mushroom can be represented as the set of its morphological **parts**, each of them presenting different **colors**. Such colors can change, taking different **states** under different **circumstances**.

and the following sentences occurring in a document related to mycology:

> *Fine tubes, which are separated from the stem, generally very little, and of a toasted yellow color that turns to a dark brown when old.*

An IE system should be able to recognize the following chunks as relevant information: *tubes* and *stem* as instances of **part**, *toasted yellow* and *dark brown* as instances of **color**, and *when old* as the instance of **circumstance** in which the latter color occurs. Moreover, the system should recognize the fact that both colors are related to the *tubes*. Then, the output of the extraction process could be the set of templates shown in Figure 1.

3 Historical Framework of IE

The development of IE technology is closely related to Message Understanding Conferences (MUC[2]), developed until 1998. MUC efforts, among others, have consolidated IE as a useful technology for TBI systems. Such conferences started in 1987 and were sponsored by the United States Advanced Research Projects Agency (DARPA[3]). In 1990, DARPA launched the TIPSTER Text program[4] to fund the research efforts of several of the MUC participants.

[1] The concepts to be dealt with are written in bold.
[2] www.itl.nist.gov/iaui/894.02/related_projects/
[3] www.darpa.mil/
[4] www.ldc.upenn.edu/Catalog/Tipster.html

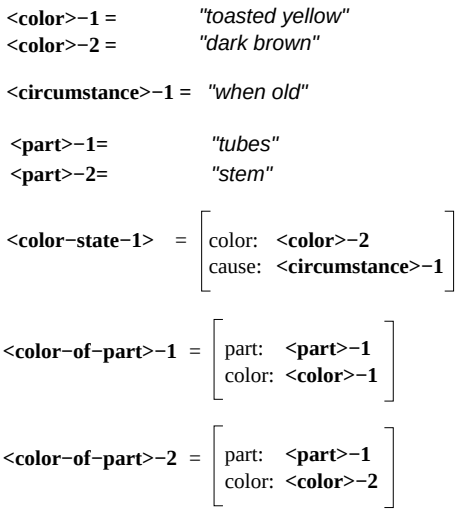

Figure 1: Example of output templates extracted by an IE system.

The general goal of MUC conferences was the evaluation of IE systems developed by different research groups in order to extract information from restricted-domain free-style texts. A different domain was selected for each conference. In order to evaluate the systems and before providing the set of evaluation documents to be dealt with, both a set of training documents and the scenario of extraction were provided to the participants by the MUC organization.

A brief description of MUC conferences will be done in chronological order, as follows below.

MUC-1 (1987). It was basically exploratory. In this first competition, neither extraction tasks nor evaluation criteria were defined by organizers, although *Naval Tactical Operations* was the selected domain of the documents. Each group designed its own format to record the extracted information.

MUC-2 (1989). The same MUC-1 domain was proposed. However, this time, organizers defined a task: template filling. A description of naval sightings and engagements, consisting of 10 slots (type of event, agent, time and place, effect, etc.) was given to the participants. For every event of each class, a template with the relevant information had to be filled. The evaluation of each system was done by the participants themselves. As a consequence, consistent comparisons among the competing systems were not achieved.

MUC-3 (1991). The domain of the documents was changed to *Latin American Terrorism* events. The template consisted of 18 slots (type of incident, date, location, perpetrator, target, instrument, etc.). The evaluation was significantly broader in scope than in previous MUCs. A training set of 1300 texts was given to the participants, while over 300 texts were set aside as test data. Four measures were defined over correct extracted slots (COR), incorrect extracted slots (INC), spurious extracted slots (SPUR), missing slots (MISS) and partially extracted ones (PAR). The two most relevant measures were *recall* (R) and *precision* (P), which measure the completeness and accuracy of the system, respectively. They were defined as follows:

$$R = \frac{COR + (0.5 \cdot PAR)}{COR + PAR + INC + MISS}$$

$$P = \frac{COR + (0.5 \cdot PAR)}{COR + PAR + INC + SPUR}$$

However, a conclusion of the evaluation was that a single overall measure was needed to achieve a better global comparison among systems.

MUC-4 (1992). Basically, the same MUC-3 task was used. However, the MUC-3 template was slightly modified and increased to 24 slots. The evaluation criteria were improved to allow global comparisons among the different competing systems. As a solution, the F-measure was proposed as a form of the harmonic mean between both recall and precision:

$$F = \frac{(\beta^2 + 1.0) \cdot P \cdot R}{\beta^2 \cdot P + R} \qquad 0 < \beta \leq 1$$

where β is the relative importance given to recall over precision. If $\beta = 1$, recall and precision are of equal weight. For recall half as important as precision, $\beta = 0.5$.

MUC-5 (1993). In the previous conferences, competing IE systems were only applied to extract information from documents in English. In the MUC-5 conference, documents in Japanese were also dealt with. Moreover, two different domains were proposed, named *Joint Ventures* (JV) and *Microelectronics* (ME), which consisted of financial news and spots on microelectronics products, respectively. The efforts in this competition focused on the template design, which were intended to affect the success in capturing information from texts. Two different sets of object-oriented templates were defined (11 templates for

JV, and 9 for ME). The evaluation was done by using the same MUC-4 measures, which were named *recall-precision-based metrics*. However, *error-based metrics* were included in order to classify systems by their error rates.

MUC-6 (1995). The *Financial* domain was used. Three main goals were proposed. Firstly, identifying functions that would be largely domain independent from the component technologies being developed for IE. In order to meet this goal, the named entity (NE) recognition task was proposed to deal with names of persons and organizations, locations and dates, among others. Secondly, focusing on portability in the IE task to different event classes. The *template element* (TE) task was proposed to standardize the lowest-level concepts (people, organizations, etc.), since they were involved in many different types of events. Like the NE task, this was also seen as a potential demonstration of the ability of systems to perform a useful, relatively domain independent task with near-term extraction technology (it required merging information from several places in the text). The old-style MUC IE task, based on a description of a particular class of event, was called *scenario template* (ST). Finally, a latter goal was encouraging participants to build up the mechanisms needed for deeper understanding. Three new tasks were proposed: *coreference resolution*, *word sense disambiguation* and *predicate-argument syntactic structuring*. However, only the first one was evaluated. Focusing on the evaluation of all the tasks, no credit, even partial, was given to partially extracted slots. As a consequence, recall and precision metrics were formulated as follows:

$$R = \frac{COR}{COR + INC + MISS}$$

$$P = \frac{COR}{COR + INC + SPUR}$$

MUC-7 (1998). The *Airline Crashes* domain was proposed. The differences between this and latter competitions were not substantial. The NE task was carried out in Chinese, Japanese and English. Moreover, a new task was evaluated, which focused on the extraction of relations between TEs, as *location-of*, *employee-of* and *product-of* in the *Financial* domain. Such a new task was named *Template Relation* (TR).

Table 1 presents the best results reported through the MUC competitions in which evaluation measures where defined. These results show the difficulty of each type of IE task and subtask. ST tasks are the most difficult ones, for which F-scores lower than 60% were achieved. On the contrary, NE subtasks are easier to be dealt with than the rest.

EVAL	NE	COR	TE	TR	ST
MUC-3					R<50 P<70
MUC-4					F<56
MUC-5					F<53
MUC-6	F<97	R<63 P<72	F<80		F<57
MUC-7	F<94	F<62	F<87	F<76	F<51

Table 1: Best results reported in MUC-3 through MUC-7 by task.

IE technology has been applied to other domains in free text, from those used in MUC. In [Soderland et al., 1995], diagnosis and signs or symptoms in the medical domain are extracted from hospital discharge reports. In [Holowczak and Adam, 1997], information useful to classify legal documents is extracted. In [Glasgow et al., 1998], relevant information related to the life insurance domain is extracted to support subscribers.

Other works refer to the application of IE technology to semi-structured documents. Within such a framework, traditional IE systems do not fit. This is because such documents are a mixture of grammatical and ungrammatical texts, sometimes telegraphic, and including meta-tags (e.g., HTML). Some examples can be found in [Soderland, 1999], for the apartment rental advertising domain, in [Craven, 1999], for the biomedical domain, and in [Freitag, 1998b; Califf, 1998], for web pages.

4 The Architecture of IE Systems

Within the historical framework presented in the previous section, the general architecture of an IE system was defined by Hobbs [Hobbs, 1993] in MUC-5. It was described as *a cascade of transducers or modules* (such as that depicted in Figure 2) *that, at each step, add structure to the documents and, sometimes, filter relevant information, by means of applying rules*. Most current systems follow this general architecture, although specific systems are characterized by their own set

of modules, and most of the architectures are currently in progress. In general, the combination of such modules allows some of the following functions in a higher or lower degree:

- Document preprocessing
- Full or partial syntactic parsing.
- Semantic interpretation, to generate either a logical form or a partial template from a parsed sentence.
- Discourse analysis, to link related semantic interpretations between sentences. This is done by means of coreference and anaphora resolution and other kinds of semantic inferences.
- Output template generation, to translate the final interpretations into the desired format.

A brief description of each function is presented below.

4.1 Document Preprocessing

The preprocessing of the documents can be achieved by a variety of modules, such as the following: *text zoners* (turning a text into a set of text zones), *segmenters*, also named *splitters* (in charge of segmenting zones into appropriate units, usually sentences), *filters* (selecting the relevant segments), *tokenizers* (to obtain lexical units), *lexical analyzers* (includes morphological analysis and NE recognition and classification), engines dealing with unknown words, *disambiguators* (POS taggers, semantic taggers, etc.), *stemmers* and *lemmatizers*, etc.

Most systems take advantage of available resources and general purpose (domain independent) tools for the preprocessing step. Specially interesting for IE are the NE recognition modules. The process of NE recognition may be quite straightforwardly performed by using *finite-state transducers* and dictionary lookup (domain specific dictionaries, terminological databases, etc.). In spite of this, results heavily depend on the information sources involved. Grishman, for instance, in [Grishman, 1995], used the following specific sources: a small gazetteer, containing the names of all countries and most major cities; a company dictionary derived from the Fortune 500; a Government Agency dictionary; a dictionary of common first names; and a small dictionary of scenario specific terms.

4.2 Syntactic Parsing and Semantic Interpretation

A more important controversy arises from parsing. At the beginning of MUC conferences, traditional NL understanding architectures were adopted for IE technology. Such approaches were based on full parsing, followed by a semantic interpretation of the resulting in-depth syntactic structure and a discourse analysis. In MUC-3, however, best scores were achieved by a simpler approach presented by Lehnert's research group [Lehnert et al., 1991], named *selective concept extraction*. Such a new approach was based on the assumption that only those concepts being within the scenario of extraction are relevant to be detected in the documents. Consequently, syntactic and semantic analysis should be simplified by means of a more restricted, deterministic and collaborative process. Their strategy was to replace the traditional parsing, interpretation and discourse modules with a simple phrasal parser, to find local phrases, an event pattern matcher, and a template merging procedure, respectively. In MUC-4, Hobbs' group recasted such an approach in terms of a more flexible model, which was based on finite-state transducers (FST) [Appelt et al., 1992].

The simplification of the understanding process, presented by both Lehnert and Hobbs, is being widely adopted by the IE community. In general, this fact is due to different drawbacks of the use of full parsing [Grishman, 1995]:

- Full parsing involves a large and relatively unconstrained search space, and is consequently expensive.
- Full parsing is not a robust process because a global parse tree is not always achieved. In order to correct such incompleteness, the parse covering the largest substring of the sentence is attempted. Sometimes, however, this global goal led to incorrect local choices of analyses.
- Full parsing may produce ambiguous results. More than one syntactic interpretation is usually achieved. In this situation, the most correct interpretation must be selected.

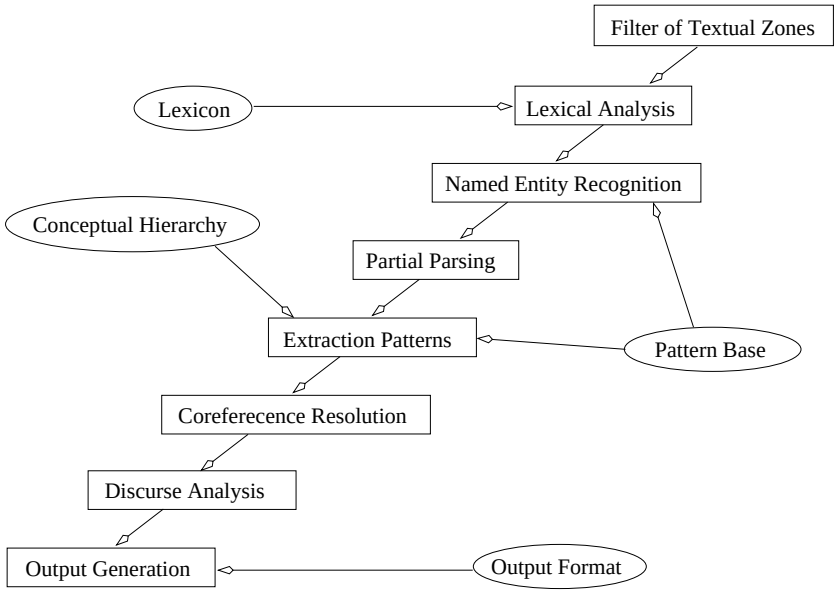

Figure 2: Usual architecture of an IE system.

- Broad-coverage grammars, needed for full parsing, are difficult to be consistently tuned. Dealing with new domains, new specialized syntactic constructs could occur in texts and be unrecognized by broad-coverage grammars. Adding new grammar rules could produce a non-consistent final grammar.

- A full parsing system cannot manage off-vocabulary situations.

Nowadays, most existing IE systems are based on partial parsing, in which non-overlapping parse fragments, being phrasal constituents, are generated. Generally, the process of finding constituents consists in using a cascade of one or more parsing steps with fragments. The resulting constituents are tagged as noun, verb, or prepositional phrases, among others. Sometimes, these components are represented as chunks of words, and the parsing process is named *chunking*[5]. However, they can also be represented as parse subtrees.

Once constituents have been parsed, systems resolve domain-specific dependencies between them, generally by using the semantic restrictions imposed by the scenario of extraction. Two different approaches are usually followed to resolve such dependencies:

- *Pattern matching.* This approach is followed by most IE systems. Syntax simplification allows reducing semantic processing to simple pattern matching, where scenario-specific patterns, also named *extraction patterns* or *IE rules*, are used to identify both modifier and argument dependencies between constituents. In fact, such IE rules are sets of ambiguity resolution decisions to be applied during the full parsing process. They can be seen as sets of syntactic-semantic expectations from the different extraction tasks. On the one hand, some IE rules allow to identify properties of entities and relations between such entities (TE and TR extraction tasks as defined in MUC). In general, this is done by using local syntactic-semantic information about nouns and modifiers. On the other hand, IE rules using predicate-argument relations (object, subject, modifiers) allow to identify events between entities (ST extraction task as defined in MUC). The representation of these IE rules greatly differs among different IE systems.

- *Grammatical relations.* Generally, the pattern-matching strategy requires a proliferation of task-specific IE rules, with explicit variants for each verbal form, explicit variants for different lexical heads, etc. Instead of using IE rules, a more flexible syntactic model is proposed in [Vilain, 1999] similar to that in [Carroll et al., 1998]. It consists in defining a set of grammatical relations be-

[5] In fact, the formal definition of chunk is a bit more complex (c.f., [Abney, 1996]). A chunk is usually considered as a simple non-recursive phrasal constituent.

tween entities as general relations (subject, object and modifier), some specialized modifier relations (temporal and location) and relations for arguments that are mediated by prepositional phrases, among others. In a way similar to dependency grammars, a graph is built by following general rules of interpretation for the grammatical relations. Previously detected chunks are nodes within such a graph, while relations between them are labeled arcs.

4.3 Discourse Analysis

IE systems generally proceed by representing the information extracted from a sentence either as partially filled templates or as logical forms. Such information can be incomplete due to the occurrence of ellipsis, and sometimes, it can refer to the same entities in the presence of coreference, anaphora. The main goal of the Discourse Analysis phase is the resolution of these semantic aspects. Systems working with partial templates make use of some merging procedure for such a task. However, working with logical forms allows IE systems to use traditional semantic interpretation procedures at this phase.

4.4 Output Template Generation

Finally, the Output Template Generation phase mainly aims at mapping the extracted pieces of information onto the desired output format. However, some inferences can occur in this phase due to domain-specific restrictions in the output structure, like in the following cases:

- Output slots that take values from a predefined set.
- Output slots that are forced to be instantiated.
- Kinds of extracted information that generate a set of different output templates. For instance, in the MUC-6 financial domain, when a succession event is found that involves a person leaving and another person taking up the same job in an organization, two different output templates have to be generated: one for the person leaving and another for the person starting.
- Output slots that have to be normalized. For instance, dates, products that have to be normalized with a code from a standard list, etc.

5 Multilingual IE

In MUC competitions, English was the unique language used, with the exception of Japanese in MUC-5 and Chinese and Japanese in the NE recognition task of MUC-7. Even the introduction of these tasks, however, the monolingual status of the competition was not changed. In general, a different monolingual IE system was used for each language. However, the growing availability of on-line texts in languages different to English has increased the interest in Multilingual IE (MIE), focused on extracting information from documents written in different source languages and presenting the output templates to the user's language.

In general, relatively little research has adressed in MIE technology. In parallel to MUC conferences, the European Comission[6] funded, under the LRE (Linguistic Research and Engineering) program[7], a number of projects devoted to develop tools and components for IE adressed to more than one language (also for IR), such as automatically or semi-automatically acquiring and tuning lexicons from corpus, extracting entities, parsing in a flexible and robust way, and so on. Some of these projects are ECRAN[8], SPARKLE[9], FACILE[10], and AVENTINUS[11].

Beyond MUC competitions, the research on IE technology has been included in the TIDES (Translingual Information Detection, Extraction and Summarization) program[12], funded by DARPA in 1999. It is an initiative on fast machine translation and information access, in which translingual IE technology is involved. Some of the sponsored projects are RIPTIDES[13], PROTEUS[14], CREST[15], Coreference.com[16], UMass

[6] europa.eu.int/comm/index_en.htm
[7] www.ejeisa.com/nectar/t-book/html/lre.htm
[8] www.dcs.shef.ac.uk/research/groups/nlp/funded/ecran.html
[9] www.ilc.pi.cnr.it/sparkle.html
[10] tcc.itc.it/research/textec/projects/facile/facile.html
[11] svenska.gu.se/aventinus/aventinus.html
[12] www.darpa.mil/iao/TIDES.htm
[13] www.cs.cornell.edu/Info/People/cardie/tides/
[14] cs.nyu.edu/cs/projects/proteus/
[15] crl.nmsu.edu/Research/Projects/Crest/
[16] www.coreference.com/

system[17], and EMPOWER[18]. Within this new framework, the Document Understanding Conferences (DUC[19]) aim at evaluating the systems presented by different research groups.

Within the framework of these programs and competitions, two different approaches for dealing with MIE tasks have been described. As explained in the following sections, both approaches are based on the use of *language guessers* and the reuse of monolingual IE architectures in different ways: either using machine translation or expanding a monolingual IE architecture to a translingual one.

5.1 The Role of Language Guessers

As described before, a MIE system should be able to process input documents written in different languages. However, the process of extracting information from documents written in a particular language may require a set of strategies and linguistic knowledge different to that required to deal with another language (e.g., lexical knowledge, POS tagging strategies, syntactic grammars, extraction patterns, etc). Consequently, in order to apply the correct knowledge to the extraction process,a basic and preliminar goal for MIE consists in identifying the language in which an input document is written. This is, in fact, the role of language guessers.

Traditionally, language guessers require a list of functional words that identify each language (i.e., identifying vocabulary). The main idea of this linguistic approach is that at least one word from a typical sentence written in some language should be included in the corresponding identifying vocabulary. However, this dictionary is manually built by an expert. Among the conventional methods to automate this task, stochastic language guessers are the most widely used. In general, they are based on a) generating a frequency table for each language by checking elements that appear commonly in a particular language and b) comparing the frequencies of elements found in the input document with the frequency table of each language. Generally, these elements are special characters [Beesley and Kenneth, 1988], or word sequences [Grefenstette, 1995], or byte sequences [Cavnar and Trenkle, 1994]. These methods achieve good results (over 95% in accuracy). However, one of their most important difficulties is dealing with short texts. A stochastic word-based approach taht copes with this problem is presented in [Zhdanova and Mankevich, 2002], which is able to identify 8 languages using frequency tables of limited size (less than 1500 keys per language) achieving an accuracy of 99% even for texts containing one sentence.

[17] ciir.cs.umass.edu/research/tides.html
[18] www.cis.upenn.edu/~josephr/TIDES/tides.html
[19] duc.nist.gov/

5.2 Use of Machine Translation

An extended method to deal with multilingual IE tasks consists in combining existing Machine Translation (MT) systems with monolingual IE systems. This can be done in two different ways. The first approach consists in using a battery of monolingual IE systems, each one connected to a MT system. Each input document written in a source language is manipulated by the appropriate IE system. Once the output templates have been generated, they are translated into the target language by the required MT system. Full MT systems are not necessary here because only the lexical items occurring in the output templates have to be translated. This approach needs an IE system for each possible source language and a mini MT system for each pair <source language, target language>.

The second approach requires less number of resources. It consists in using a unique monolingual IE system connected to a set of full MT systems that translate documents into the language of the IE system, and to a mini MT system that allows to translate the output templates into the desired target language. A particular case occurs when the output templates are wanted to be expressed in the same language than that of the IE system. In this situation, the mini MT system is not required. Note, however, that this second approach implies the repetition of many NL processes that are common for both MT and IE (e.g., tokenization, sentence splitting, POS tagging, parsing, etc), although for different languages. Due to this fact, the whole process becomes highly time-consuming. As described below, translingual approaches try to solve this problem.

5.3 Translingual IE Systems

Translingual IE can be seen as the fusion of both IE and interlingua MT IE technologies. As de-

scribed in [Gaizauskas et al., 1997], translingual IE is based on the assumption that, when dealing with a particular domain, it is possible to construct a language-independent conceptual model of the particular scenario of extraction (i.e., a language-independent domain model). This assumption implies that a translingual IE system requires:

- A different lexical analysis for each source language. A separate lexicon for each possible language. Language-specific lexical information is represented separately from the language-independent domain model. In this representation, lexical items and language-independent concepts are linked by many-to-many relations.
- A different syntactic-semantic analysis for each possible language. In the case of IE systems in which such analysis is based on pattern-matching (c.f. Section 4.2), a separate set of extraction patterns is also needed for each language (and domain).

When performing the extraction process from a particular language, a translingual IE system makes use of the appropriate extraction patterns and the relations between lexical items and concepts in order to represent the results in a language-independent internal form (logical forms or partial templates, as described in Section 4.3). Once the extraction process ends, the output templates are generated from that language-independent internal representations via the lexicon of the target language. It is important to note that this step may require some technique for lexical choice: several lexical entries may refer to the same concept, thus a selection is required. This is a significant problem for NL generation in general, as explained in [Horacek and Zock, 1993].

6 Machine Learning for Portable IE

It is well known that each module involved in the extraction process achieves results with higher or lower accuracy. This fact leads to the *error propagation* problem, meaning that a small error could produce a greater one along the extraction process.

On the other hand, IE systems should to be as portable as possible to new situations. Dealing with new domains, new languages and/or author styles implies that different kinds and amounts of new specific knowledge will be needed to achieve results. Moreover, new scenarios of extraction could imply new concepts to be dealt with, which is beyond the capabilities of the IE system.

As a conclusion, the difficulty of exploiting IE technology is mainly due to the intrinsic error propagation along the understanding process and the difficulty of adapting IE systems to deal with *portability drawbacks*, such as those explained above.

In order to handle such difficulties, IE technology has benefited from the improvements achieved in all the involved tasks along the last two decades. Most of such improvements are based on the use of empirical methods in NLP. Due to the characteristics of knowledge needed by NLP empirical approaches, machine learning (ML) techniques have been widely used for acquisition. [Sekine et al., 1998; Borthwick et al., 1998; Baluja et al., 1999; Borthwick, 1999], for NE recognition, [Cardie et al., 2000], for chunking, and [McCarthy and Lehnert, 1995; Aone and Bennet, 1996; Cardie and Wagstaff, 1999; Mitkov, 1998], for coreference and anaphora resolution, are interesting examples of such approaches. A detailed thorough survey on the use of ML techniques for NLP tasks can be also found in [Young and Bloothooft, 1997; Manning and Schütze, 1999; Mooney and Cardie, 1999].

Most of the research effort in this area has been devoted to applying symbolic inductive learning methods to acquire domain-dependent, language-dependent knowledge that is useful for extraction tasks. Most of the approaches focus on acquiring IE rules from a set of training documents written in a particular language. These rules can be classified either as *single-slot rules* or *multi-slot rules*, by taking into account that a concept can be represented as a template (e.g., a template element, a template relation or a scenario template in MUC terms). A single-slot rule is able to extract document fragments related to one slot within a template, whilst a multi-slot rule extracts tuples of document fragments related to the set of slots within a template. The representation of extraction rules depends heavily on the document style, from which rules have to be learned. In general, the less structured the documents, the higher the variety of linguistic constraints within the rules.

APPROACH	KL	PARADIGM	STRATEGY	EXACT	DOC
AutoSlog [Riloff, 1993] AutoSlog-TS [Riloff, 1996] [Riloff and Jones, 1999]	rules	propositional learning	heuristic driven specialization	no	f
[Harabagiu and Maiorano, 2000]					
PALKA [Kim and Moldovan, 1995]			candidate elimination		
[Chai and Biermann, 1997]			brute force		
TIMES [Chai et al., 1999]					
[Basili et al., 2000]			heuristic driven generalization		
CRYSTAL [Soderland et al., 1995]			bottom-up covering		
WAVE [Aseltine, 1999]					
ESSENCE [Català et al., 2000]					
ExDISCO [Yangarber, 2000]					
LIEP [Huffman, 1995]		relational learning	top-down covering	yes	s/ss/f
WHISK [Soderland, 1999]					ss/f
EVIUS [Turmo and Rodríguez, 2002]					
SRV [Freitag, 1998a]			bottom-up compression		ss
RAPIER [Califf, 1998]					
SNoW-IE [Roth and Yih, 2001]	linear separators	propositional learning	multi-class classifier		
[Seymore et al., 1999]	Hidden Markov Models	statistical learning			
[Freitag and McCallum, 2000]					
[McCallum et al., 2000]					

Table 2: A classification of ML approaches for IE specific knowledge acquisition.

Surveys presenting different kinds of rules used for IE can be found in [Muslea, 1999; Glickman and Jones, 1999].

From MUC competitions, typical IE rule learners focus on learning rules from free text to deal with event-extraction tasks. With some exceptions, these rules are useful to extract document fragments containing the slot-filler values, and a postprocessing is needed to extract the exact values from the fragments. However, the large amount of online documents available on the Internet has recently increased the interest in algorithms that can automatically process and mine these documents by extracting relevant exact values. Within this framework, some systems have been developed/applied to learn single-slot rules. Few works have focused on learning other type of knowledge useful for such a task (e.g., Hidden Markov Models (HMMs), linear separators).

Table 2 shows a classification of different stateof-the-art ML approaches on acquiring such knowledge. In this table, different versions of an initial approach have been grouped in the first column. The second column (KL) refers to the type of knowledge learned by the approaches. The learning paradigm and the learning strategy of the approaches are shown in the third (Paradigm) and fourth (Strategy) columns, respectively. The fifth column (Exact) shows the approaches that learn knowledge that is useful to extract exact slot-filler fragments. The final column (Doc) refers to the type of documents from which approaches learn (free text – f, semi-structured documents – ss, and structured ones – s). More details on the different approaches are provided below.

6.1 Rule Learning

Since the MUC-5 conference, some approaches have focused on the application of methods to automatically acquire IE rules. Such methods can be classified from different points of view: the degree of supervision, the kind of rules learned, the kind of training document styles (i.e., the kind of document style to which learned rules can be applied), the learning paradigm, or the learning strategy used, among others. In this section, some state-of-the-art IE rule learning systems are classified from these points of view. For the sake of simplicity, the degree of supervision they need has been taken as the starting point of the comparison.

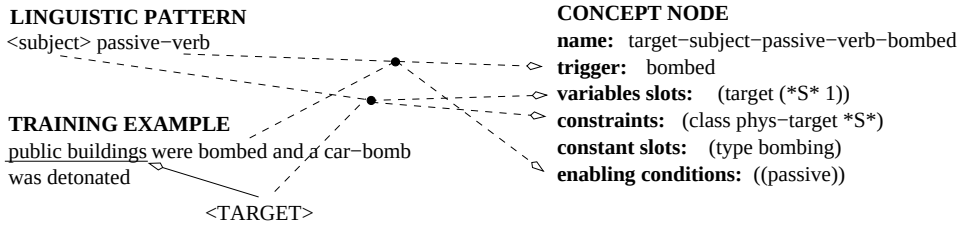

Figure 3: A concept node induced by AutoSlog.

6.1.1 Supervised Approaches

Under the symbolic inductive learning paradigm, supervised approaches are the most common ones to learn IE rules. In general, a learning method is supervised if some intervention is required from the user for the learning process. Some approaches, however, require the user to provide with training examples. This supervision can be carried out as a preprocessing (i.e., appropriately tagging the examples occurring in the training set) or as an online process (i.e., dynamically tagging the examples as needed during the learning process). Some of these learning approaches work in the context of propositional learning, while others perform in the context of relational learning.

Propositional Learning. Propositional learning is based on representing the examples of a concept in terms of the zero order logic or attribute-value logic, which have equivalent expressiveness in a strict mathematical sense. Within this learning paradigm, some IE research groups have developed learning systems in order to learn IE rules from positive examples. In general, examples of a concept are represented as sets of attributes (i.e. slots) whose values (i.e., slot fillers) are heads of syntactic phrases occurring within the training documents. Rules learned by these approaches are useful to extract information from parsed free text. They identify the syntactic phrases that contain the heads of the slot fillers. Often, however, partial matches occur between these phrases and the exact slot filler. This is enough when the aim is to approximately extract the relevant information. When exact values are required, a postprocessing is mandatory in order to extract these values.

One of the earliest approaches was AutoSlog [Riloff, 1993]. AutoSlog generates single-slot rules, named *concept nodes* by applying a heuristic-driven specialization strategy. A concept node is defined as a *trigger word* that will activate the concept node when performing the extraction, and a set of restrictions involving the trigger and the slot-filler values. Each training example is a chunk within a sentence annotated as slot-filler value. The generation of concept nodes is based on the specialization of a set of predefined heuristics being general linguistic patterns. This process is carried out examining each example only once. For instance, in Figure 3, the linguistic pattern **<subject> passive-verb** is specialized into **<target>bombed** when examining the training example *public buildings were bombed and a car-bomb was detonated* for the bombing event (*public buildings* was previously annotated as slot-filler value). As a consequence, a concept node for the **<target>** slot of a bombing template is generated having **bombed** as trigger and constraining the slot-filler value to be subject within the training example. The resulting set of concept nodes is proposed to the user, in order to be reviewed. This is due to the fact that AutoSlog makes no attempt to generalize examples, and, consequently, generates dubious or very specific slot-filler definitions.

Other approaches learn rules by generalizing examples. Some of them can learn single-slot rules and multi-slot rules, such as PALKA [Kim and Moldovan, 1995], CRYSTAL [Soderland et al., 1995], WAVE [Aseltine, 1999] and the one presented in [Chai and Biermann, 1997], while some others learn single-slot rules, like TIMES [Chai et al., 1999]. With the exception of PALKA, these systems translate examples manually annotated in sentences into specific rules, to which a generalization process is applied.

PALKA is based on a candidate-elimination algorithm. Specific rules (*Frame-phrasal Pattern Structures* – FP-structures) are requested from the user. These specific rules are similar to AutoSlog's. However, chunks that are slot-filler values are represented as the semantic class of their heads. PALKA generalizes and specializes such semantic classes by using an ad-hoc semantic hi-

erarchy until the resulting FP-structures cover all the initial specific ones. The use of the semantic hierarchy allows PALKA learn rules that are more general than AutoSlog's. However, no generalizations are made on trigger words.

Specific rules used by CRYSTAL are in the form of concept nodes, but they consist of a set of features related to the slot-fillers and the trigger. Values for these features can be terms, heads, semantic classes, syntactic functions (subject, direct or indirect object), verbal modes, etc. CRYSTAL uses a bottom-up covering algorithm in order to relax such features in the initial specific rules. This relaxation is achieved by means of both dropping out irrelevant features, and generalizing semantic constraints by using an ad-hoc semantic hierarchy. For instance, the concept node for the succession event shown in Figure 4 was learned by CRYSTAL to be used in MUC6 competition. Filler values for slots *Person-In*, *Position* and *Organization* of a succession template are extracted when their constraints are satisfied by a parsed sentence. WAVE is similar to CRYSTAL, but it relies on an incremental learning approach in which a reusable hierarchy of partially learned rules is maintained along the learning process.

CONCEPT NODE
concept type: Succession Event
constraints:
SUBJ::
classes include: <Person>
extract: Person_In
VERB::
terms include: NAMED
mode: passive
OBJ::
terms include: OF
classes include: <Corporate Post>, <Organization>
extract: Position, Organization

Figure 4: A concept node learned by CRYSTAL.

Rules learned by CRYSTAL are more expressive than those learned by AutoSlog or PALKA. This is due to the fact that CRYSTAL can learn rules where constraints on any noun and verb within both the slot-filler values and the triggers could be based not on the words but on some generalizations. However, the ad-hoc semantic hierarchy has to be manually built when dealing with new domains.

A different approach is presented in [Chai and Biermann, 1997], in which a broad covering semantic hierarchy, WordNet[20][Miller et al., 1990], is used. By default, the system assigns the most frequently used synset[21] in WordNet to the head of each slot-filler within a specific rule. The user, however, can assign a more appropriate sense. At the first step, specific rules are semantically generalized by using a brute-force generalization algorithm to keep recall as high as possible. This process consists in replacing each noun synset in a specific rule by its top hypernym in WordNet. At the second step, these top synsets are specialized by tuning the rules to adjust the precision. A more recent version of this approach is followed by TIMES, in which two generalization steps (syntactic and semantic) are performed with specific rules in order to generate the maximum number of candidates for single-slot rules.

Rule sets learned by TIMES are more general than those learned by CRYSTAL and by the approach presented in [Chai and Biermann, 1997]. This is due to the use of permutations of constituents in the generalization step. However, as opposed to CRYSTAL, TIMES and the approach in [Chai and Biermann, 1997] also need the user's supervision during the learning process.

Relational Learning. Relational learning is based on representing the examples of a concept in terms of the first order logic. Within this learning paradigm, IE rule learning systems represent training examples in terms of attributes and relations between textual elements (e.g., tokens, constituents).

Within this paradigm, LIEP [Huffman, 1995] automatically generalizes training examples of events occurring in free text into multi-slot rules by using a bottom-up covering algorithm. If a given example is not matched by any learned rule, LIEP attempts to further generalize a rule. If this is not possible, LIEP builds a new specific rule from the example. Specific rules used by LIEP consist of a feature set, similar to that used by CRYSTAL. However, as opposed to CRYSTAL, LIEP has no prior information about syntactic functions of chunks. LIEP learns such information as syntactic relations ($subject(A, B)$, $object(A, B)$, $prep\_object(A, B)$, etc.), by using a form of explanation-based learning with an overgenerated and incomplete theory. This is why LIEP works within relational learning. The gen-

[20] www.cogsci.princeton.edu/~wn
[21] A synset in WordNet groups together a set of word senses, variants, related by a loose kind of synonymy.

eralization proceeds by creating disjunctive values, so LIEP rules cannot take into account missing values in the training corpus. As a consequence, CRYSTAL rules are more expressive than LIEP's.

Some other approaches are based on general relational learning systems, and more specifically, on Inductive Logic Programming (ILP) systems well known by the ML community (e.g., FOIL [Quinlan, 1990; Quinlan and Cameron-Jones, 1993], CIGOL [Muggleton and Buntine, 1988], GOLEM [Muggleton and Feng, 1992], CHILLIN [Zelle and Mooney, 1994] and PROGOL [Muggleton, 1995]). Two examples of these approaches are SRV [Freitag, 1998a] and RAPIER [Califf, 1998]. Both systems are applied to semi-structured documents for learning single-slot rules to extract exact values.

SRV is an ILP system based on FOIL. SRV transforms the problem of learning IE rules into a classification problem: is a document fragment a possible slot value? The input of this system is a training set of documents, and a set of attributive and relational features related to tokens T (e.g., $capitalized(T)$, $next(T_1, T_2)$) that control the generalization process. Introducing domain-specific linguistics or any other information is a task separate from the central invariable algorithm. SRV uses a top-down covering algorithm to learn IE rules from positive and negative examples. Slot-filler fragments within training documents are manually annotated as positive examples. Negative examples are automatically generated taking into account empirical observations related to the number of tokens of positive examples. In Figure 5, a rule learned by SRV from semi-structured documents related to seminar announcements is depicted. This rule extracts exact values for the slot *speaker*.

RAPIER is a relational learning system based on GOLEM, CHILLIN and PROGOL. It uses a bottom-up compression algorithm in which rules are iteratively merged instead of generalized from training examples. RAPIER considers training examples as specific rules. At each iteration, two rules (specific or not) are selected to be compressed into a new one. Rules used in this process are discarded and the resulting rule is added to the set of possible ones. The input documents are represented as token sequences, optionally POS tagged. No parsing process is required for them. Each training example consists of three text fragments, where one of them is the slot-filler value and the other two are the left and right contexts of such value. In order to learn rules from such examples, RAPIER takes into account the implicit token succession relation and some token generalizations: token sets, POS tags or semantics derived from WordNet hierarchy.

A more flexible system within the relational learning paradigm is WHISK [Soderland, 1999]. WHISK deals with structured and semistructured documents, and also with free text. Following an approach different to RAPIER, WHISK represents documents as sequences of tokens, some of them being tags representing meta-data (HTML tags, delimiters of parsed chunks, features of heads, etc.), and allows learning single-slot and multi-slot rules to extract exact slot values. Specifically, rules are represented as pairs $< pattern, output >$, where *pattern* is meant to be matched by documents and *output* is the output template when a match occurs. The pattern is a regular expression that represents possible slot fillers and their boundaries. These rules are learned in a top-down fashion from a training set of positive examples. An unusual selective sampling approach is used by WHISK. Initially, a set of unannotated documents is randomly selected as training input from those satisfying a set of key words. These documents are presented to the user for tagging the slot-fillers. WHISK starts learning a rule from the most general pattern. Growing the rule proceeds one slot at a time. This is done by adding tokens just inside the slot-filler and outside them. Growing a rule continues until it covers at least the training set.

Although WHISK is the most flexible state-of-the-art approach, it cannot generalize on semantics when learning from free text, as CRYSTAL, PALKA, SRV and RAPIER do. Another limitation of WHISK is that no negative constraints can be learned.

6.1.2 Towards Unsupervised Approaches

The learning approaches presented above require the user to provide positive training examples for automatic learning rules. One of the main drawbacks of this supervision is the high cost of annotating positive examples in the training documents. Some approaches focus on dealing with this drawback by means of requiring a lower degree of supervision.

```
speaker:-                                          // Fragment F is a speaker if
  some(?A, [], word, *unknown*)                    // F contains a token (A), and
  every(capitalizedp, true)                        // every token in F is capitalized, and
  length(=, 2))                                    // F contains exactly 2 tokens, and
  some(?B, [], word, *unknown*)                    // F contains another token (B), and
  some(?B, [prev_token], word, ":")                // B is preceded by a colon, and
  some(?A, [next_token], doubletonp, false)        // A is not followed by a 2-char token, and
  every(quadruple_char_p,false)                    // every token in F does not consists of
                                                   // exactly 4 alpha. characters, and
  some(?B, [prev_token prev_token], word, "who")   // two tokens before B is the word "who"
```

Figure 5: A rule learned by SRV.

A supervised learning approach requiring less manual effort than previous ones was AutoSlog-TS [Riloff, 1996]. It was a new version of AutoSlog, where the user only had to annotate documents containing free text as relevant or non-relevant before learning. Recent approaches learning from unannotated free text are ESSENCE [Català et al., 2000], and those presented in [Basili et al., 2000], [Harabagiu and Maiorano, 2000], [Riloff and Jones, 1999] and [Yangarber and Grishman, 2000; Yangarber, 2000]. In general, they require some initial domain-specific knowledge (i.e., a few keywords or initial hand-crafted rules) and/or some validations from the user to effective learning.

ESSENCE is based on inducing linguistic patterns from a set of observations, instead of examples. These observations are automatically generated from unannotated training documents as a keyword (provided by the user) in a limited context. These observations are generalized by performing a bottom-up covering algorithm and using WordNet. After learning, the user is required to validate the resulting patterns, and the learning process may be repeated by using both the set of validated patterns and a set of new observations generated from new keywords. Finally, the user has to manually mark slot fillers occurring in the linguistic patterns. The resulting rules are similar to CRYSTAL's.

In [Basili et al., 2000], linguistic patterns useful to extract events from free text of specific domains are induced. The approach requires documents classified into a set of specific domains, D_i. At an initial step, the set of verbs that are relevant triggers of events is automatically selected using a statistical test. Those verbs satisfying this test are used as triggers for event matching, and for each of them, a verb sub-categorization structure is extracted by applying a conceptual clustering algorithm. Each of these structures contains the verb and its arguments, the latter being represented by their noun heads. These heads are semantically tagged using WordNet synsets. The resulting specific patterns are generalized by using heuristics: a measure of conceptual density a set of rules for passivization and alternations. Finally, multi-slot IE rules are built by manually marking the slot fillers within patterns. Validation from the user could be necessary to eliminate possible noisy verbs and/or too specific patterns obtained during the learning process.

The approach explained in [Harabagiu and Maiorano, 2000] is based on heuristic-driven specializations. Authors explain that relations between words and synsets in WordNet can be insufficient to achieve good results. Their initial experiments using domains from MUC's showed that less than 8% of relations were accounted for in WordNet. In order to improve results on semantic tagging they start by creating a semantic space that models the domain from information contained in WordNet. This is done by applying a set of domain keywords and heuristics to WordNet concepts and relevant connections between them. This semantic space can be seen as a set of linguistic patterns more general than those used by AutoSlog. In the second step, parsed chunks being subject, verb and object of sentences are scanned so as to allocate domain concepts within the semantic space. Production rules are induced using the principle of maximal coverage of allocated concepts. The performance of the induced rules generates correct syntactic links emerging from domain concepts, a fact that enables the derivation of linguistic patterns in a FASTSPEC [Appelt et al., 1995] manner. Finally, these patterns are classified according to WordNet and only the most general ones are retained. The validation of the semantic space could be necessary.

Recently, some research groups have been focusing on the use of some forms of bootstrapping, as

in [Riloff and Jones, 1999] and in the ExDISCO system [Yangarber and Grishman, 2000; Yangarber, 2000]. In the former work, some predefined seed words for the semantic category of interest are used by AutoSlog in order to learn an initial set of rule candidates. These rules are applied to unannotated corpora. The rule achieving best results is selected, and its extractions are added to the set of relevant words related to the semantic category for further bootstrapping.

On the contrary, in ExDISCO, a few seed rules representing events of interest are initially handcrafted. Documents containing matches for seed rules are retrieved, and taken as relevant documents. Matches are taken as specific rules, which are ranked based on their frequencies in relevant and non-relevant documents. The best specific rule is automatically selected and added to the set of seed rules, while those documents containing the best rule are added to the relevant ones. At each step, discovered specific rules are semantically generalized in a way similar to LIEP's. The process is repeated until no new relevant documents are found.

Learning approaches based on bootstrapping could require to iteratively reject noisy extractions resulting from matching rules with documents.

6.2 Other ML Approaches

Although rule learning techniques have been the most common ones used for IE, some other ML techniques have also been applied, such as linear separators, HMMs, Maximum Entropy models, or a combination of different techniques.

6.2.1 Linear Separators

The work presented in [Roth and Yih, 2001] describes a new approach for learning to extract slot fillers from semi-structured documents. This approach, called SNoW-IE, follows a two-step strategy. In the first step, a classifier is learned to achieve high recall. Given that a common property of IE tasks is that negative examples are extremely more frequent than positive ones, this classifier aims at filtering most of the former ones without discarding the latter ones. In the second step, another classifier is learned to achieve high precision when applied to the output of the first classifier. Both classifiers are learned as sparse networks of linear functions (i.e. linear separators of positive and negative examples) from manually annotated training set by applying SNoW [Roth, 1998], a propositional learning system. Training examples are represented as conjunctions of propositions. Basically, each proposition refers to an attributive feature related to a token. This token can occur in the slot filler, in its left context (l_window) or in its right context (r_window). Positive and negative training examples are automatically generated by using a set of constraints, such as (a) the appropriate lengths (in tokens) of the context windows, (b) the maximum length of the slot filler, and (c) the set of appropriate features for tokens within the filler and both of its context windows (e.g., word, POS tag, location within the context). These constraints are defined by the user for each concept to be learned.

6.2.2 Statistical Learning

Within this framework, some efforts have focused on learning HMMs as useful knowledge to extract relevant fragments from online documents available on the Internet.

In general, these efforts take into account only words. For instance, in [Seymore et al., 1999] a method is presented for learning HMMs state/transition structure. The approach uses a single HMM to extract a set of fields from semi-structured texts (e.g., research paper headers). However, large amounts of training data are required to accurately learn a model. Another approach is presented in [Freitag and McCallum, 2000]. This work focuses on robustly learning an HMM for each field from limited training data.

Other approaches take benefit from word features (e.g., POS tags, capitalization, position in the document, etc.), or from features of sequences of words (e.g., length, indentation, total amount of white-space, grammatical features, etc.). This is the case of the approach presented in [McCallum et al., 2000].

6.2.3 Multi-strategy Approaches

In [Freitag, 1998a], the advantage of using a multi-strategy approach to learn to extract information is demonstrated within the framework of learning from online documents. Single ap-

proaches for this purpose take a specific view of the documents (e.g., HTML tags, typographic information, lexical information). This fact implies the introduction of biases that make such approaches less suitable for some kind of documents than for others.

The experiment in [Freitag, 1998a] focused on combining three separate machine learning paradigms to learn single-slot rules: rote memorization, term-space text classification, and relational rule induction. When performing extraction, learners' confidences were mapped into probabilities of correctness by using a regression model. Such probabilities were combined in order to produce a consensus among learners. This combination of learners (each one of them using different kinds of information to learn) achieved better results than the individual learners.

Within the relational learning paradigm, a different multi-strategy approach is used by EVIUS [Turmo and Rodríguez, 2002; Turmo, 2002] to learn single-slot and multi-slot IE rules from semi-structured documents and free text. Learning systems explained previously are single-concept learners. They can learn knowledge useful to extract instances of a concept within the scenario of extraction in an independent way. Instead, EVIUS assumes the fact that the scenario of extraction imposes some dependencies between concepts to be dealt with. When a concept depends on another one, knowledge about the former seems to be useful to learn to extract instances of the latter one.

EVIUS is a supervised multi-concept learning system based on a multi-strategy constructive learning approach [Michalski, 1993] that integrates closed-loop learning, deductive restructuring [Ko, 1998] and constructive induction. Closed-loop learning allows EVIUS to incrementally learn IE rules similar to Horn clauses for the whole scenario of extraction. This is done by means of determining which concept to learn at each step. Within this incremental process, the learning of IE rules for each concept is basically accomplished using FOIL, that requires positive and negative examples. On the one hand, positive examples are annotated in the training example using an interface. Artificial examples can be automatically generated when the training corpus is insufficient to learn enough rules for a concept and no more corpus is available for training. On the other hand, the most relevant negative examples are generated using clustering techniques. Once IE rules for a concept have been learned, the learning space is updated using deductive restructuring and constructive induction. These techniques allow to assimilate knowledge possibly useful for further learning: the training examples of learned concepts and new predicates related to these concepts.

The experiments in [Turmo, 2002] show that better results are achieved when applying the multi-strategy constructive learning approach of EVIUS, the clustering-based method for selecting relevant negative examples and the incremental addition of artificial examples, when needed.

7 Examples of IE Systems

Extracting from structured or semi-structured documents can be performed without making use of any postprocessing, and frequently with the use of few preprocessing steps. Within this framework, automatically induced wrappers and IE rules learned either by using SRV, RAPIER, or WHISK, can be either directly applied to the extraction task, as a whole IE system, or integrated as a component of an already existing IE system for specific tasks. This is why this section aims at comparing architectures of IE systems for free text only, in particular 16 of the most representative ones in the state of the art: CIRCUS [Lehnert et al., 1991, 1992, 1993] and its successor BADGER [Fisher et al., 1995], FASTUS [Appelt et al., 1992, 1993a,b, 1995], LOUELLA [Childs et al., 1995], PLUM [Weischedel et al., 1991, 1992, 1995; Weischedel, 1995] and its successor IE2 [Aone et al., 1998], PROTEUS [Grishman and Sterling, 1993; Grishman, 1995; Yangarber and Grishman, 1998], ALEMBIC [Aberdeen et al., 1993], HASTEN [Krupka, 1995], LOLITA [Morgan et al., 1995; Garigliano et al., 1998], LaSIE [Gaizauskas et al., 1995], its successor LaSIE-II [Humphreys et al., 1998], PIE [Lin, 1995], SIFT [Miller et al., 1998] and TURBIO [Turmo, 2002]. All of them are monolingual IE systems, although a translingual extension of LaSIE exists (M-LaSIE [Gaizauskas et al., 1997]).

The comparisons do not take into account either the preprocessing or the output template generation methods because there are no important differences in this aspect. We summarize the comparison describing the way different systems face the main extraction tasks: syntactic parsing, semantic interpretation and discourse anal-

<table>
<tr><th>SYSTEM</th><th>SYNTAX</th><th>SEMANTICS</th><th>DISCOURSE</th></tr>
<tr><td>LaSIE</td><td colspan="3" rowspan="3">in-depth understanding</td></tr>
<tr><td>LaSIE-II</td></tr>
<tr><td>LOLITA</td></tr>
<tr><td>CIRCUS</td><td rowspan="5">chunking</td><td rowspan="3">pattern matching</td><td rowspan="2">template merging</td></tr>
<tr><td>FASTUS</td></tr>
<tr><td>BADGER</td></tr>
<tr><td>HASTEN</td><td rowspan="2">-</td></tr>
<tr><td>PROTEUS</td><td rowspan="3">traditional semantic interpretation procedures</td></tr>
<tr><td>ALEMBIC</td><td>grammatical relation interpretation</td></tr>
<tr><td>PIE</td><td rowspan="3">partial parsing</td><td rowspan="3">pattern matching</td></tr>
<tr><td>TURBIO</td><td>-</td></tr>
<tr><td>PLUM</td><td rowspan="2">template merging</td></tr>
<tr><td>IE2</td><td colspan="2">pattern matching</td></tr>
<tr><td>LOUELLA</td><td colspan="2">syntactic-semantic parsing</td><td>-</td></tr>
<tr><td>SIFT</td><td colspan="3">syntactic-semantic parsing</td></tr>
</table>

Table 3: Methodology of state-of-the-art IE systems.

<table>
<tr><th>SYSTEM</th><th>SYNTAX</th><th>SEMANTICS</th><th>DISCOURSE</th></tr>
<tr><td>LaSIE</td><td>general grammar extracted from the Penn TreeBank corpus [Gaizauskas, 1995]</td><td rowspan="2">λ-expressions</td><td rowspan="5">-</td></tr>
<tr><td>LaSIE-II</td><td>hand-crafted stratified general grammar</td></tr>
<tr><td>LOLITA</td><td>general grammar</td><td>hand-crafted semantic network</td></tr>
<tr><td>CIRCUS</td><td rowspan="6">phrasal grammars</td><td>concept nodes learned from AutoSlog</td></tr>
<tr><td>FASTUS</td><td>hand-crafted IE rules</td></tr>
<tr><td>BADGER</td><td>concept nodes learned from CRYSTAL</td><td>trainable decision trees</td></tr>
<tr><td>HASTEN</td><td>E-graphs</td><td rowspan="5">-</td></tr>
<tr><td>PROTEUS</td><td>IE rules learned from ExDISCO</td></tr>
<tr><td>ALEMBIC</td><td>hand-crafted grammatical relations [Vilain, 1999]</td></tr>
<tr><td>TURBIO</td><td>IE rules learned from EVIUS</td></tr>
<tr><td>PIE</td><td rowspan="2">general grammar</td><td rowspan="2">hand-crafted IE rules</td></tr>
<tr><td>PLUM</td></tr>
<tr><td>IE2</td><td colspan="2" rowspan="2">hand-crafted IE rules</td><td>hand-crafted rules and trainable decision trees</td></tr>
<tr><td>LOUELLA</td><td>-</td></tr>
<tr><td>SIFT</td><td colspan="3">statistical model for both syntactic-semantic parsing and coreference resolution learned from the Penn TreeBank corpus and domain-specific annotated texts</td></tr>
</table>

Table 4: Knowledge used by state-of-the-art IE systems.

ysis. Table 3 describes their characteristics from the viewpoint of the methodology they use to perform extraction. The pros and contras of each method have been commented in section 4. Table 4 describes the kind of knowledge representation used by the selected IE systems. As shown in this table, only a few of the systems take advantage of ML techniques to automatically acquire domain-specific knowledge useful for the extraction (CIRCUS, BADGER, PROTEUS, TURBIO and SIFT). These IE systems are more easily portable than the others. However, in the last MUC competition (MUC-7), the best results in IE tasks were achieved by IE2, which uses handcrafted knowledge.

8 Conclusion

Information Extraction is now a major research area within the text-based intelligent systems discipline mainly thanks to two factors. On the one hand, there are many applications that require specific-domain knowledge (i.e. knowledge-based systems), and manually building this knowledge can become highly expensive. On the other hand, given the growing availability of on-line documents in different languages, the number of them that may contain such knowledge is high, and can be automatically extracted and integrated into such applications.

One of the main drawbacks of IE technology, however, refers to the difficulty of adapting IE systems to new domains and languages. Classically, this task involves manual tuning of the domain-dependent linguistic knowledge integrated into IE systems, such as terminological dictionaries, domain-specific lexico-semantics and extraction patterns, among others.

Since the early 90's, the research efforts focus on the use of empirical methods to automate and reduce the high cost of dealing with these portability issues. Special efforts concentrate on the use of ML techniques for the automatic acquisition of the extraction patterns useful to deal with a particular language and IE domain, which is one of the most expensive issues. Supervised learning approaches are most commonly applied in the state of the art. However, the task of annotating positive examples in training documents is hard, and the research is being directed to the development of less supervised learning approaches, such as those using observation-based learning or forms of bootstrapping, among others.

References

J. Aberdeen, J. Burger, D. Connolly, S. Roberts, and M. Vilain. Description of the Alembic system as Used for MUC-5. In *Proceedings of the 5th Message Understanding Conference (MUC–5)*, 1993.

S. Abney. *Principle-Based Parsing: Computation and Psycholinguistics*, chapter Parsing by Chunks. Kluwer Academic Publishers, Dordrecht, 1996.

E. Agirre and G. Rigau. Word Sense Disambiguation Using Conceptual Density. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING*, Copenhagen, Denmark, 1996.

C. Aone and W. Bennet. Evaluation Automated and Manual Acquisition of Anaphora Resolution. In E. Riloff, S. Wermter, and G. Scheler, editors, *Lecture Notes in Artificial Intelligence*, volume 1040. Springer, 1996.

C. Aone, L. Halvenson, T. Hampton, and M. Ramos-Santacruz. Description of the IE2 System Used for MUC–7. In *Proceedings of the 7th Message Understanding Conference (MUC–7)*, 1998.

D. Appelt, J. Bear, J. Hobbs, D. Israel, M. Kameyama, and M. Tyson. Description of the JV-FASTUS System Used for MUC-5. In *Proceedings of the 5th Message Understanding Conference (MUC–5)*, 1993a.

D. Appelt, J. Bear, J. Hobbs, D. Israel, and M. Tyson. Description of the jv-fastus system used for muc-4. In *Proceedings of 4th Message Understanding Conference (MUC–4)*, 1992.

D. Appelt, J. Hobbs, J. Bear, D. Israel, M. Kameyama, and M. Tyson. Description of the FASTUS System as Used for MUC-6. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, 1995.

D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. FASTUS: A finite-state Processor for Information Extraction. In *Proceedings of 13th International Join Conference On Artificial Intelligence (IJCAI)*, 1993b.

J.H. Aseltine. WAVE: An incremental Algorithm for Information Extraction. In *Proceedings of the AAAI Worshop on Machine Learning for Information Extraction*, 1999.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, 1999.

S. Baluja, V. Mittal, and R. Sukthankar. Applying Machine Learning for High Performance Named-Entity Extraction. In *Proceedings of the International Conference of Pacific Association for Computational Linguistics (PACLING)*, 1999.

R. Basili, M.T. Pazienza, and M. Vindigni. Corpus-driven Learning of Event Recognition Rules. In *Proceedings of the ECAI Workshop on Machine Learning for Information Extraction*, 2000.

Beesley and Kenneth. Language Identifier: a Computer Program for Automatic Language Identification of On-line Text. In *Proceedings of the 29th Annual Conference of the American Translators Association*, 1988.

A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000.

A. Borthwick. *A Maximum Entropy Approach to Named Entity Recognition.* PhD thesis, Computer Science Department. New York University, 1999.

A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Proceedings of the 6th ACL Workshop on Very Large Corpora*, 1998.

M.E. Califf. *Relational Learning Techniques for Natural Language Information Extraction.* Ph.d. thesis, University of Texas at Austin, 1998.

C. Cardie, W. Daelemans, C. Nédellec, and E. Tjong Kim Sang, editors. *Proceeding of the Fourth Conference on Computational Natural Language Learning*, 2000.

C. Cardie and K. Wagstaff. Noun Phrase Coreference as Clustering. In *Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP / VLC)*, 1999.

J. Carroll, T. Briscoe, and A. Sanfilippo. Parser Evaluation: A Survey and a New Proposal. In *Proceedings of 1st. International Conference on Language Resources and Evaluation (LREC)*, pages 447–454, Granada, Spain, 1998.

N. Català, N. Castell, and M. Martín. ESSENCE: a Portable Methodology for Acquiring Information Extraction Patterns. In *Proceedings of 14th European Conference on Artificial Intelligence (ECAI)*, pages 411–415, 2000.

W. B. Cavnar and J. Trenkle. N-gram Based Text Categorization. In *Proceedings of the 3rd. Annual Symposium on Document Analysis and Information Retrieval*, 1994.

J.Y. Chai and A.W. Biermann. The Use of Lexical Semantics in Information Extraction. In *Proceedings of the ACL Workshop on Natural Language Learning*, 1997.

J.Y. Chai, A.W. Biermann, and C.I. Guinn. Two-Dimensional Generalization in Information Extraction. In *Proceedings of the 11th AAAI National Conference on Artificial Intelligence (AAAI)*, 1999.

L. Childs, D. Brady, L. Guthrie, J. Franco, D. Valdes-Dapena, B. Reid, J. Kielty, G. Dierkes, and I. Sider. LOUELLA PARSING, an NL-Toolset System for MUC–6. In *Proceedings of the 6th Message Understanding Conference (MUC–6)*, 1995.

M. Craven. Learning to Extract Relations from MEDLINE. In *Proceedings of the AAAI Workshop on Machine Learning for Information Extraction*, 1999.

D. Fisher, S. Soderland, J. McCarthy, F. Feng, and W. Lehnert. Description of the UMass System Used for MUC–6. In *Proceedings of the 6th Message Understanding Conference (MUC–6)*, 1995.

D. Freitag. *Machine Learning for Information Extraction in Informal Domains.* Ph.d. thesis, Computer Science Department, Carnegie Mellon University, 1998a.

D. Freitag. Toward General-Purpose Learning for Information Extraction. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Anual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 1998b.

D. Freitag and A. McCallum. Information Extraction with HMM Structures Learned by Stochastic Optimization. In *Proceedings of the 21th National Conference on Artificial Intelligence (AAAI)*, 2000.

R. Gaizauskas. Investigations into the Grammar Underlying the Penn Treebank II. Research Memorandum CS-95-25, Department of Computer Science, Unviersity of Sheffield, UK, 1995.

R. Gaizauskas, K. Humphreys, S. Azzam, and Y. Wilks. Concepticons vs. Lexicons: Architecture for Multilingual Information Extraction. In M.T. Pazienza, editor, *Lecture Notes in Artificial Intelligence*, volume 1299, pages 28–43. Springer, 1997.

R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. Description of the LaSIE System as Used for MUC–6. In *Proceedings of the 6th Message Understanding Conference (MUC–6)*, 1995.

R. Garigliano, A. Urbanowicz, and D. Nettleton. Description of the LOLITA System as Used for MUC–7. In *Proceedings of the 7th Message Understanding Conference (MUC–7)*, 1998.

B. Glasgow, A. Mandell, D. Binney, L. Ghemri, and D. Fisher. MITA: An Information Extraction Approach to Analises of Free-form Text in Life Insurance Applications. *Artificial Intelligence*, 19:59–72, 1998.

O. Glickman and R. Jones. Examining Machine Learning for Adaptable End-to-End Information Extraction Systems. In *Proceedings of the AAAI Workshop on Machine Learning for Information Extraction*, 1999.

G. Grefenstette. Comparing Two Language Identification Schemas. In *Bolasco S., Lebart L. Salem A. (eds.) JADT*, 1995.

G. Grefenstette. *Cross-Language Information Retrieval.* Kluwer AP, 1998.

R. Grishman. Where is the syntax? In *Proceedings of the 6th Message Understanding Conference (MUC–6)*, 1995.

R. Grishman and J. Sterling. Description of the PROTEUS System as Used for MUC-5. In *Proceedings of the 5th Message Understanding Conference (MUC–5)*, 1993.

S.M. Harabagiu and S.J. Maiorano. Acquisition of Linguistic Patterns for Knowledge-Based Information Extraction. In *Proceedings of 3rd International Conference on Language Resources and Evaluation (LREC)*, 2000.

J. Hobbs. The Generic Information Extraction System. In *Proceedings of the 5th Message Understanding Conference (MUC–5)*, pages 87–92, 1993.

R.D. Holowczak and N.R. Adam. Information Extraction based Multiple-Category Document Classification for the Global Legal Information Applications. In *Proceedings of the 9th AAAI National Conference on Artificial Intelligence (AAAI)*, pages 992–999, 1997.

H. Horacek and M. Zock. *New Concepts in Natural Language Generation: Planning, Realization and Systems.* Pinter Publishers, London, 1993.

S. Huffman. Learning information extraction patterns from examples. In *Proceedings of the IJCAI Workshop on New Approaches to Learn for NLP*, 1995.

K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. Description of the LaSIE-II System as Used for MUC–7. In *Proceedings of the 7th Message Understanding Conference (MUC–7)*, 1998.

J. Kim and D. Moldovan. Acquisition of Linguistic Patterns for Konwledge-based Information Extraction. In *IEEE Transactions on Knowledge and Data Engineering*, 1995.

H. Ko. Empirical Assembly Sequence Planning: A Multistrategy Constructive Learning Approach. In R. S. Michalsky, I. Bratko, and M. Kubat, editors, *Machine Learning and Data Mining.* John Wiley & Sons LTD, 1998.

G.R. Krupka. Description of the SRA System Used for MUC–6. In *Proceedings of the 6th Message Understanding Conference (MUC–6)*, 1995.

W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. Description of the CIRCUS System as Used for MUC–4. In *Proceedings of the 4th Message Understanding Conference (MUC–4)*, 1992.

W. Lehnert, C. Cardie, D. Fisher, E. Riloff, and R. Williams. Description of the CIRCUS

System as Used for MUC–3. In *Proceedings of the 3rd Message Understanding Conference (MUC–3)*, 1991.

W. Lehnert, J. McCarthy, S. Soderland, E. Riloff andC. Cardie, J. Peterson, F. Feng, C. Dolan, and S. Goldman. Description of the CIRCUS System as Used for MUC–5. In *Proceedings of the 5th Message Understanding Conference (MUC–5)*, 1993.

D. Lin. Description of the PIE System Used for MUC–6. In *Proceedings of the 6th Message Understanding Conference (MUC–6)*, 1995.

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, 2000.

J.F. McCarthy and W.G. Lehnert. Using Decision Trees for Coreference Resolution. In *Proceedings of the 14th International Join Conference on Artificial Intelligence (IJCAI)*, 1995.

R.S. Michalski. Towards a unified theory of learning: Multistrategy task–adaptive learning. In B.G. Buchanan and D. Wilkins, editors, *Readings in Knowledge Acquisition and Learning.* Morgan Kauffman, 1993.

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five Papers on WordNet. *Special Issue of International Journal of Lexicography*, 3(4), 1990.

S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, and R. Weishedel. Description of the SIFT System Used for MUC–7. In *Proceedings of the 7th Message Understanding Conference (MUC–7)*, 1998.

R. Mitkov. Robust Pronoun Resolution with Limited Knowledge. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Anual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 869–875, 1998.

R.J. Mooney and C. Cardie. Symbolic Machine Learning for Natural Language Processing. Tutorial in Workshop on Machine Learning for Information Extraction. AAAI, 1999.

R. Morgan, R. Garigliano, P. Callaghan, S. Poria, M. Smith, A. Urbanowicz, R. Collingham, M. Constantino, and C. Cooper. Description of the LOLITA System as Used for MUC–6. In *Proceedings of the 6th Message Understanding Conference (MUC–6)*, 1995.

S. Muggleton. Inverse Entailment and Progol. *New Generation Computing Journal*, 13:245–286, 1995.

S. Muggleton and W. Buntine. Machine Invention of First-Order Predicates by Inverting Resolution. In *Proceedings of the 5th International Conference on Machine Learning (ICML)*, 1988.

S. Muggleton and C. Feng. Efficient Induction of Logic Programs. In S. Muggleton, editor, *Inductive Logic Programming.* Academic Press, New York, 1992.

I. Muslea. Extraction Patterns for Information Extraction Tasks: A Survey. In *Proceedings of the AAAI Workshop on Machine Learning for Information Extraction.*, 1999.

J. R. Quinlan. Learning Logical Definitions from Relations. *Machine Learning*, 5(3):239–266, 1990.

J.R. Quinlan and R.M. Cameron-Jones. FOIL: A Midterm Report. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 3–20, Vienna, Austria, 1993.

E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI)*, pages 811–816, 1993.

E. Riloff. Automatically Generating extraction patterns from untagged texts. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI)*, pages 1044–1049, 1996.

E. Riloff and R. Jones. Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI)*, 1999.

D. Roth. Learning to Resolve Natural Language Ambiguities: A Unified Approach. In *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI)*, pages 806–813, 1998.

D. Roth and W. Yih. Relational Learning via Propositional Algorithms: An Information Extraction Case Study. In *Proceedings of the 15th*

International Conference On Artificial Intelligence (IJCAI), 2001.

S. Sekine, R. Grishman, and H. Shinnou. A Decision Tree Method for Finding and Classifying Names in Japanese Texts. In *Proceedings of the SIG NL/SI of Information Processing Society of Japan*, 1998.

K. Seymore, A. McCallum, and R. Rosenfeld. Learning Hidden Markov Model Structure for Information Extraction. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI).*, 1999.

S. Soderland. Learning to Extract Text-based Information from the World Wide Web. In *Proceedings of the 3th International Conference on Knowledge Discovery and Data Mining (KDD)*, 1997.

S. Soderland. Learning Information Extraction Rules for Semi–Structured and Free Text. *Machine Learning*, 34:233–272, 1999.

S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1314–1321, 1995.

T. Strzalkowski. *Natural Language Information Retrieval.* Kluwer, 1999.

C.A. Thompson, M.E. Califf, and R.J. Mooney. Active Learning for Natural Language Parsing and Information Extraction. In *Proceedings of 16th International Machine Learning Conference*, pages 406–414, 1999.

J. Turmo. *An Information Extraction System Portable to New Domains.* Ph.d. thesis, Technical University of Catalonia, 2002.

J. Turmo and H. Rodríguez. Learning Rules for Information Extraction. *Natural Language Engineering. Special Issue on Robust Methods in Analysis of Natural Language Data.*, 8:167–191, 2002.

M. Vilain. Inferential Information Extraction. In M.T. Pazienza, editor, *Information Extraction: Towards Scalability, Adaptable Systems. Lecture Notes in Artificial Intelligence*, volume 1714. Springer–Verlag, Berlin, 1999.

J.L. Vilcedo. *SEMQA: Un modelo semántico aplicado a los sistemas de Búsqueda de Respuestas.* Ph.d. thesis, University of Alicante, 2002.

R. Weischedel. Description of the PLUM System as Used for MUC–6. In *Proceedings of the 6th Message Understanding Conference (MUC–6)*, 1995.

R. Weischedel, D. Ayuso, S. Boisen, H. Fox, H. Gish, and R. Ingria. Description of the PLUM System as Used for MUC–4. In *Proceedings of the 4th Message Understanding Conference (MUC–4)*, 1992.

R. Weischedel, D. Ayuso, S. Boisen, H. Fox, R. Ingria, T. Matsukawa, C. Papageorgiou, D. MacLaughlin, M. Kitagawa, T. Sakai, L. Abe, H. Hosihi, Y. Miyamoto, and S. Miller. Description of the PLUM System as Used for MUC–5. In *Proceedings of the 5th Message Understanding Conference (MUC–5)*, 1995.

R. Weischedel, D. Ayuso, S. Boisen, R. Ingria, and J. Palmucci. Description of the PLUM System as Used for MUC–3. In *Proceedings of the 3rd Message Understanding Conference (MUC–3)*, 1991.

R. Yangarber. *Scenario Customization of Information Extraction.* Ph.d. thesis, Courant Institute of Mathematical Sciences. New York University, 2000.

R. Yangarber and R. Grishman. Description of the PROTEUS/PET System as Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC–7)*, 1998.

R. Yangarber and R. Grishman. Machine Learning of Extraction Patterns from Unnanotated Corpora: Position Statement. In *Proceedings of the ECAI Workshop on Machine Learning for Information Extraction*, 2000.

S. Young and G. Bloothooft. *Corpus-Based Methods in Language and Speech Processing.* Kluwer Academic Press, 1997.

J.M. Zelle and R. J. Mooney. Inducing Deterministic Prolog Parsers from Treebanks: A Machine Learning Approach. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)*, pages 748–753, 1994.

A.V. Zhdanova and P.V. Mankevich. Automatic Identification of European Languages. In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems*, 2002.