ESSENCE: a Portable Methodology for Building Information Extraction Systems

Neus Català
ncatala@lsi.upc.es

Abstract

One of the most important issues when constructing an Information Extraction System is how to obtain the knowledge needed for identifying relevant information in a document. A manual approach not only is an expensive solution but also has a negative effect on the portability of the system across domains. To automatize the knowledge acquisition process may partially solve this problem even if a human expert takes part in it only for specific tasks. This work presents a methodology (ESSENCE) to automatically learn information extraction patterns from unrestricted text corpus representative of the domain. The methodology includes different steps from which we stress the specific pattern generalization process. Generalization reduces the pattern base and therefore reduces the amount of information to validate by an expert. As we will see, the use of the lexical knowledge along with the lexico-semantic relations from WordNet are our basis knowledge source, especially, for the generalization process.

Keywords: Information Extraction, WordNet and Information Extraction Systems Portability, Learning Information Extraction Patterns.

1 Introduction

The goal of an Information Extraction (IE) system is to identify and extract specific information from a document. The kind of information to extract is up to a prespecified set of events, entities and relationships.

While an Information Retrieval (IR) system given a keywords list returns a set of relevant documents which contain them, an IE system only returns the required information in a prefixed format [4].

When building an IE system there is an unavoidable task concerning acquisition of some sort of extraction structures . In the last years different approaches have been proposed in order to automatize this task starting from preprocessed texts, such as training corpus tagged with domain-specific tags, hand-written syntactic and semantic patterns, etc. But these approaches has to face the time cost of manual effort, usually done by an expert, along with the lost of work when moving the system to a new domain.

This work proposes a new methodology, named ESSENCE, for automatic building of IE pattern bases from corpus without specific tagging and representative of the domain. This new approach reduces the human expert effort concentrating his intervention in the validation and typification tasks over IE patterns, and it also allows to reuse the obtained patterns in other extraction related applications.

The next section gives a brief description of Information Extraction tasks. We comment troubles when constructing pattern bases for IE systems and we point out some existing systems in this area. Fourth section presents ESSENCE, the methodology proposed, and discusses the advantages it introduces. At the end, we want to give a future watch of research to do in order to build "fully" automatic IE systems.

2 Information Extraction

IE is a Natural Language Processing (NLP) task [2] which goal is to extract predetermined kinds of information from a document. IE systems are domain specific because they extract particular events or facts from a particular domain skipping over the irrelevant ones. For instance, in the aircraft crashes domain an IE system must extract information about the aircraft involved in the accident, and the location and the date of the crash, the number of victims, etc. Figure 1 shows an overall of IE systems architecture[1].

[1]Lexical resources and components may vary among IE systems.

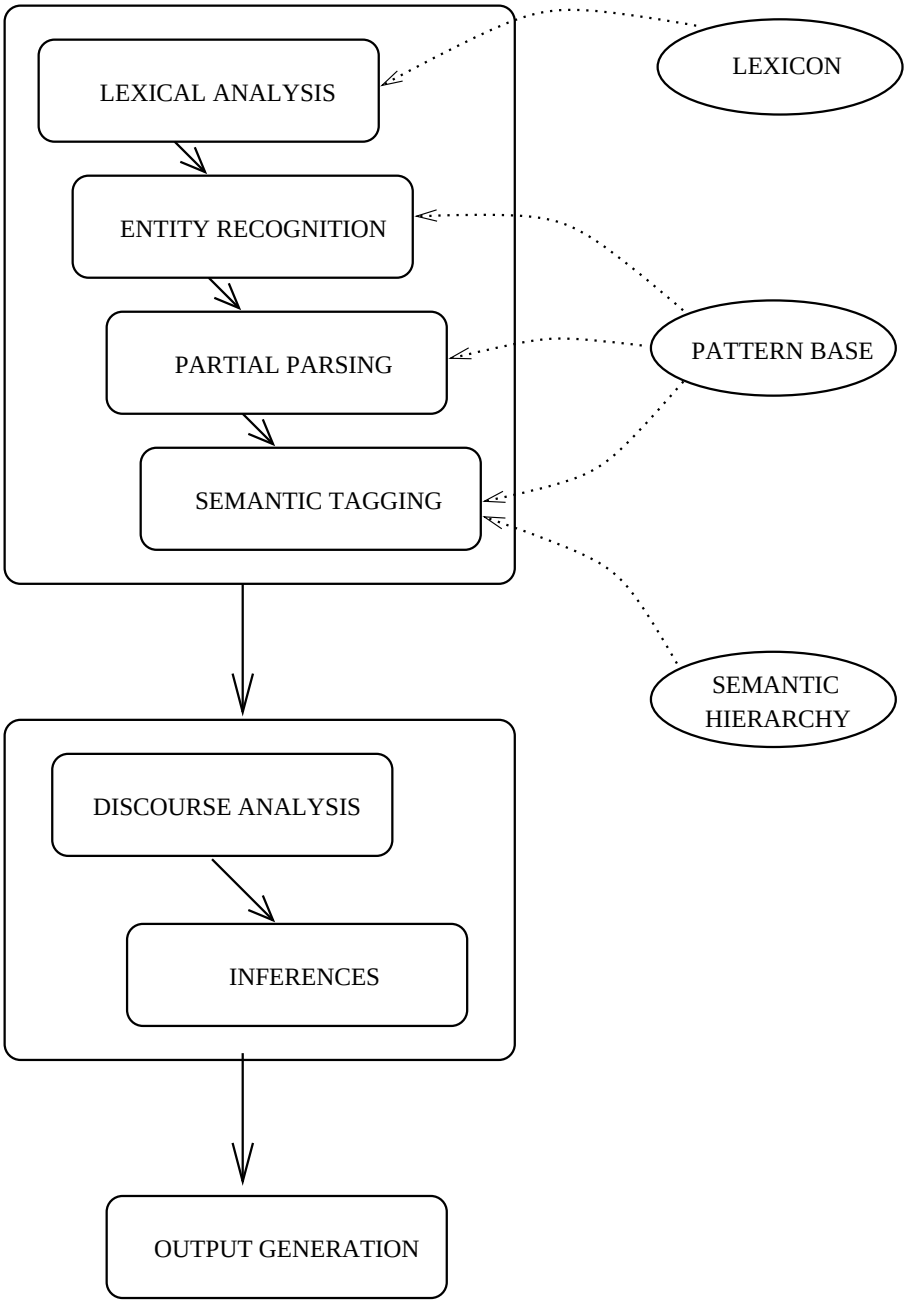

Figure 1: *Overall of IE systems architecture.*

In the last years, several conferences centered on IE, the wide known *Message Understanding Conferences*, had motivated the development of new approaches and systems in this area. The goal of these conferences is the evaluation of IE systems developed by different research groups and each conference suggests a new domain (e.g., Latin American terrorism [15] [16], Joint Ventures and Microelectronics [17], Management Succession [18]). The MUC organization members give a set of training text and a set of *answer keys*, manually done, which are intended to show the kind of information a system must to extract. The evaluation metrics are based on the values of two[2] factors: 1) *recall* is the percentage of possible answers which were correct and 2) *precision* is the percentage of actual answers given which were correct. From both definitions one can deduce that attempts to improve one factor impact negatively on attempts to improve the other; this is the reason why for the evaluation of IE systems measures that combines both factors together, weighted according to organization criteria, are used. For example, sometimes a F-measure is used as a combined recall(R)-precision(P) score, which is:

$$\frac{(\beta^2 + 1.0)\ P\ R}{\beta^2\ P + R}$$

$\beta = 1$ if P and R are equally important.

At the same time, systems participating in the MUC competition had tended to unify the kind of processing they do, mainly due to new focusing proposed by the organization. MUC oriented IE systems aims to identify and extract all information pertaining to a set of prespecified events (defined as *scenario*), within a limited *domain*, and store that information in prestructured *templates*. This one, is a general definition that involves several tasks which are Named Entity (NE), Coreference (CO), Template Element (TE) and Scenario Template (ST)[3]. The main motivation for the preparation of these tasks is to minimize non-NLP requirements while improving system portability across domains. Precisely, the significance given to portability has prompted successful approaches that have faced the problem using a

[2]Currently these metrics have been extended to enable different types of evaluations [18] [7].

[3]Currently, in MUC-7 competition, these tasks have been extended with Template Relation (TR)

pattern-matching technique[4] by obtaining frequent structures from training corpus which identify key constituents of the final output template. The ESSENCE methodology is also related to this technique.

Building an IE system requires considerable knowledge about the domain it has to deal with. On the one side, it requires knowledge about the entities and their relationships in the domain; on the other side, it must know how these entities usually appear on texts. Very often, entities in a domain that are relevant for a concrete extraction task fail in new domains and may be they are expressed differently. The portability of an IE system will depend on the acquisition of all this knowledge, or part of it, automatically.

Usually, IE systems represent local context (that contains lexical, syntactic and, may be also, semantic information) needed to extract information, by *extraction patterns*, also known as *extraction rules* or *conceptual patterns*. An extraction pattern synthesizes the set of lexical, syntactic and semantic restrictions that a sentence must to satisfy in order to extract information from it, and also which sentence components will be extracted. When a pattern applies to a fragment of text, the information extracted from it is indicated by the pattern.

The set of extraction patterns pertaining to an IE system makes up its *pattern base*. When moving an IE system to a different domain a new extraction pattern base have to be built. If this work is manually done it requires a large human effort and a human expert habituated to IE tasks, and must be repeated for each new domain. To automatize the process of acquiring pattern bases is a solution to this problem.

3 Automatically Building Information Extraction Pattern Bases

One of the most difficult tasks the construction of an information extraction system has to face is the acquisition of knowledge needed for identifying relevant information in a document. In the last years, different systems have been developed intended to automatize this task, such as AutoSlog [19] and CRYSTAL [22]. These systems generate extraction patterns from annotated

[4]This technique is usually referred as pattern-matching IE.

training corpus where information to extract is semantically[5] tagged. Annotation is clearly easier than building an extraction pattern base, but has also problems such as to decide what to tag and how perform the annotation, taking into account the need of a domain expert to do the task.

Other approaches to automatically build extraction pattern bases have avoided corpus annotation by providing other solutions. For instance, the AutoSlog-TS [21] system doesn't require annotated corpus but instead preclassified corpus, i.e., the input texts have been classified as relevant or irrelevant according to the goal of extraction. A different approach is LIEP [8] system that allows an interactive user intervention to identify relevant entities and interesting relationships between them, significative of events to extract.

Approaches that generate extraction pattern bases from corpus, with or without annotations, obtain in the first phase extraction patterns that are very close to the training corpus structure showing its writing style. If the future texts presented to the system have similar characteristics to those from training corpus, the extraction patterns will be still useful; otherwise, a large training corpus to discover all necessary patterns possible will be needed to get a robust IE system.

The use of specific patterns is very limited. An alternative is to generalize the initial extraction patterns so that they cover similar instances while maintaining some specificity (i.e, with the necessary restrictions to avoid instances that could extract irrelevant information). For example, CRYSTAL constructs extraction pattern dictionaries by means of an algorithm close to the concept induction learning described by Michalski [12]. LIEP system also generalizes patterns with a point of view closer to Explanation Based Learning (EBL) described by Mitchell [14] without a complete domain theory.

The methodology proposed in present work starts with an unrestricted training corpus containing representative (positive) texts of the kind of information to extract. From the initial corpus we generate "specific" or "lexico-syntactic" patterns that represent segments of sentences in the corpus, therefore only covering information they hold. If need be to use specific patterns

[5] Usually, the semantic tags give the text segment meaning in the context it is found. Thus, they are domain specific.

to extract information from new texts we have to generalize them. The generalization process reduces the amount of specific patterns and makes easier the validation process at the same time.

To give a summary of main differences to ESSENCE with respect to systems quoted above, relevant issues are listed:

- The training corpus has no annotations, neither syntactic tags nor semantic tags, and has positive examples of information to be extracted.
- Human intervention is restricted to validating and typifying patterns. It is also possible, but no mandatory, that a human expert gives some clues in order to locate relevant information or to guide generalization process.
- For the generalization process a semantic hierarchy will be needed. ESSENCE makes use of an existent lexical database, WordNet [13], able to cover multidomain vocabulary instead of a hand built semantic hierarchy tailored for each domain.

Many specific patterns will not be generalized and some of the generalized patterns will extract irrelevant information. All this justify the need of a mechanism that determines which patterns are actually domain specific expression, i.e. a "filtering" process that selects frequently used patterns along with those extract relevant information too and rules out the rest. Next section describes the methodology to a major extent.

4 The Proposed Methodology: ESSENCE

Different approaches presented in the previous section have in common the presence of a human expert working on. The ESSENCE methodology proposed in this work, and presented in previous works [3] [23], is intended to reduce human expert intervention when acquiring IE patterns. This goal is achieved by means of a pattern generalization (learning) algorithm which delays as much as possible the expert involvement to reduce the amount of information he has to deal with. It is nevertheless true that a human expert is required after generalization process in order to validate the results and

specify the kind of information to extract. But the fact is delaying expert intervention after generalization process allows him to work with patterns instead of corpus[6].

This section aims to describe different stages that compose the ESSENCE methodology, depicted in Figure 2, while briefly detailing requirements, purposes and performance of each one.

4.1 Preliminary Steps

This introductory stage only points out some issues needed in later stages and also comments drawbacks and alternatives.

4.1.1 Avalaible Knowledge Sources and NLP Tools

As we said above, building an IE system requires considerable knowledge about the domain such as entities and events described in it. Prior to any extraction the system has to identify those entities and events appearing on texts considered to be relevant for the domain. But to identify all possible kind of entities in whatever domain it would be necessary a large knowledge source that provides such a capacity.

Referring to MUC competitions, many systems has showed a good deal when identifying person names, organization names, locations and numerical expressions (that may refer to either currencies or time expressions). An example is LaSIE system [5] which obtained high performance levels in this task. This system, and others, make use of gazetters, word lists and trigger words (such as Inc.) incorporated into a noun phrase grammar especially designed for named entity recognition. Just mention here the problem of bug propagation through tasks. To fail in identify an entity in NE task will cause an error chain propagation to later tasks resulting in low performance levels. Bug flow propagation across tasks is represented by:

$$\text{NE} \rightarrow \text{CO} \rightarrow \text{TE} \rightarrow \text{ST}$$

Even inside a task exists the problem of bug propagation, e.g. don't recognize "CEO" as "chief executive order" is not counted only as one error

[6]It is worthy of note that starting with untagged corpus the effort reduced to the expert is even more remarkable

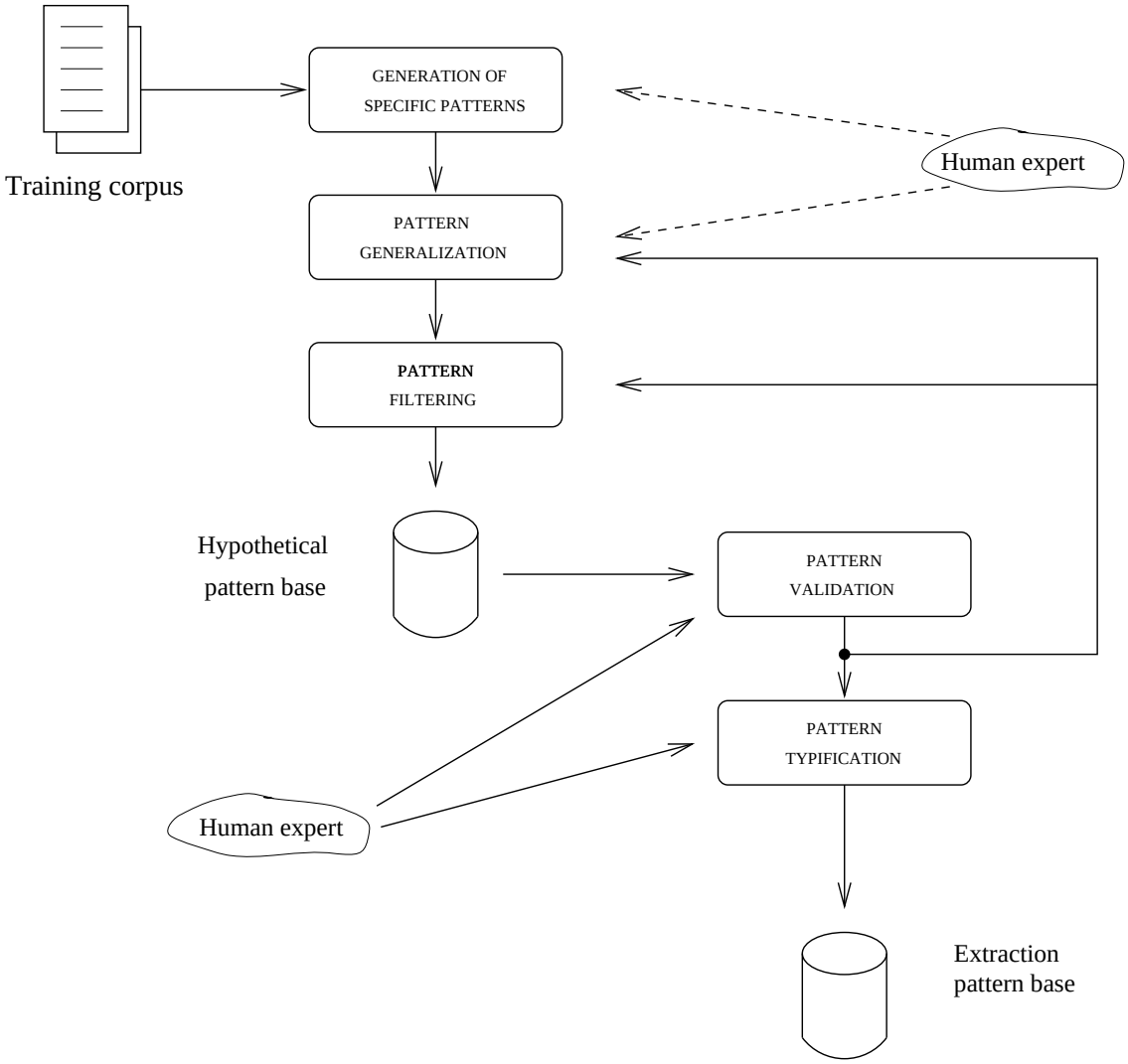

Figure 2: *Schema of* ESSENCE *methodology.*

because every repetition of "CEO" in text will produce an error in further tasks.

All NLP resources mentioned in this subsection will be avalaible in order to automatize the crucial task of entities identification, but also some existing NLP tools will be needed in further processings. For example, wide coverage lexicon, morphological analyser, POS tagger and syntactic parser, are some tools avalaible and more or less domain independent.

4.1.2 The Lexicon

A very important resource is the lexicon and the kind of information it contains. Usually we can find in a lexicon not only syntactic knowledge about words but also semantic knowledge and, in some cases, lexico-semantic relations. This is the case of WordNet, which is our basis knowledge source and it is decisive in the generalization process. In addition, the lexicon includes some multiword forms or collocations that represent a single concept

A wide-coverage lexicon avoids extending the lexicon when moving to a new domain and adding new syntantic and semantic knowledge, sometimes a difficult task to do. A drawback is the problem of ambiguity. Because that, several IE systems have semantic lexicons customized or tailored to each domain the system has to deal with.

4.1.3 Parsing

In a fully trainable IE system may be a grammar learning tool is used, but for lack of such a tool existent parsers like MARMOT[7] would be useful. At now, we are using MARMOT a sentence analyzer that identifies major syntactic constituents such as NP, VP or PP among others; we have extended its lexicon because its poor coverage.

[7]The MARMOT and BADGER software is provided by the NLP Laboratory, Univ. of Massachusetts Computer Science Department, Amherst, Massachusetts. Copyright 1990-1996 by the Applied Computing Systems Institute of Massachusetts, Inc. (ACSIOM). All rights reserved.

4.1.4 Ambiguity

Is it necessary some sort of WSD (Word Sense Disambiguation) tool? Initially, we thought that a WSD tool could reduce the amount of possible combinations when generalizing patterns. This idea has been discarded at the moment because the generalization process bounds itself unuseful sense combinations in context. On the one hand, different senses of a word not related to surrounding word senses will be discarded. On the other hand, senses retained through generalization that doesn't fit with domain context will be wiped out to be considered irrelevant during the filtering process.

4.1.5 Special Treatments

What about words not found in WordNet and that might be generalized?

After a study of WordNet coverage of nouns and verbs in the domain, we present two alternatives:

- If a word is not found in WordNet but is found in a specific gazetter that we know how to typify, we typify it with a known semantic label from WordNet or with a semantic label of our own make.
- Otherwise, can we associate it the top of the semantic hierarchy?[8]

4.1.6 Training Corpus

We have to set, by experimentation, the minimal size of the training corpus in order to obtain acceptable patterns. Always is possible to use new texts for training in addition to those provided by the organization (for instance, MUC). We can find texts of the domain somewhere and use them also as training texts to obtain new kind of patterns.

4.2 Generation of Specific Patterns

This is the first step of the ESSENCE methodology. The start point is an untagged text corpus with representative texts for the information to be extracted, that may also contain negative examples. The goal is to generate

[8] In study.

a specific pattern set that must be used as examples in the following step of generalization.

Obtaining specific patterns begins with a syntactic analysis of each sentence in the training corpus. For our purposes, we do not need any higher level parse structures since patterns will represent just local syntactic information about sentences or fragments of sentences, not distinguishing between main phrases and relative clauses. We make use of MARMOT, a component of CIRCUS system [9], a shallow parser that separates and segments sentences into noun phrases, verb phrases, and other high-level constituents. From segmented sentences, we collect parameter-sized context windows[9] do not cutting across sentence boundaries. Thus, a sentence may result in multiple context windows depending on how it is syntactically analized and the width of the window settled as parameter. Each context window is represented later as a specific pattern by considering constituents as lexico-syntactic restrictions. A new sentence, or part of it, must have exactly the same restrictions in order to a specific pattern could be applied to it. Figure 3 shows the set of specific patterns obtained from two simple sentences.

Optionally, an expert can give information (hints) about possible relevant sentences from which one might generate specific patterns. For instance, he can give a keyword list as a condition relevant sentences must contain to be considered. It is not necessary that the keyword list must be an exhaustive list but one can automatically enlarge it by using the semantic relationships, like synonymy or hyponymy, from WordNet. For instance, in the terrorism domain, the expert could give `kill` as a keyword without giving also other related words that could be obtained from WordNet. The word `kill` as a verb has 5 senses in WordNet but only the sense 2 is significative according to domain purposes.

```
Sense 2
kill -- (cause to die)
      => destroy, ruin, bust up, wreck, wrack
        Also See-> kill off
```

[9]Surrounding an occurrence of a keyword if we have a keywords list provided by an expert.

Sentence 72: At least one person has been killed in an avalanche in the Italian Alps.

```
(72
  ((NP ((NOUN AT_LEAST) (NOUN ONE) (NOUN PERSON)))
   (VP ((AUX HAS) (AUX BEEN) (VERB KILLED)))
   (PP ((PREP IN) (NPST AN) (NOUN AVALANCHE)))
   (PP ((PREP IN) (NPST THE) (NOUN ITALIAN) (NOUN ALPS))) )
)
```

Sentence 294: Five men were killed this weekend in helicopter crash in the Rocky Mountains of Southeastern B-C.

```
(294
  ((NP ((NOUN FIVE) (NOUN MEN)))
   (VP ((AUX WERE) (VERB KILLED) (ADV THIS_WEEKEND)))
   (PP ((PREP IN) (NOUN HELICOPTER) (NOUN CRASH)))
   (PP ((PREP IN) (NPST THE) (NOUN ROCKY)
        (NOUN MOUNTAINS) (NOUN OF) (NOUN SOUTHEASTERN) (NOUN B-C))))
)
```

Figure 3: Specific patterns from sentences 72 and 294 of MUC-7 texts for NE task.

Sense 2 has no synonyms but there is a great number of hyponyms, i.e particular ways to `kill`, useful to enlarge the initial list. Words printed using typewriter style represent the head of a synonym/hyponym list.

kill : `eliminate`, annihilate, extinguish, eradicate, wipe out, carry off, cancel out, `drown`, `massacre`, slaughter, mow down, `butcher`, slaughter, chine, `poison`, `stone`, lapidate, `poison`, `brain`, `put away`, put to sleep, `liquidate`, waste, knock off, do in, dispatch, `exterminate`, kill en masse, kill off, `smother`, asphyxiate, suffocate, `strangle`, throttle, garrotte, garotte, `decapitate`, behead, guillotine, `impale`, stake, `dismember`, cut to pieces, tear to pieces, quarter, draw and quarter, `hang`, string up, `murder`, slay, hit, dispatch, bump off, polish off, remove, burke, execute, murder execution-style, `assassinate`, `execute`, put to death, crucify, kill by crucifixion, electrocute, fry, burn, burn at the stake, hang, `lynch`, `shoot`, pick off, shoot one by one.

Even if the word related list obtained from WordNet may contain irrelevant words for the terrorism domain, there is no inconvenience for that because no sentence will contain them or, if they were, will be eliminated later in the filtering process due either to low frequency or its poor degree of relevancy.

4.2.1 Specific Pattern Representation

Figure 4 shows the syntax description used to represent specific patterns.

When the set of specific patterns is submitted to the generalization process we obtain a semantic pattern base. As we will see in section 4.3, generalized patterns have a syntax description similar to specific pattern syntax but semantic features have been linked to each word.

4.2.2 Syntactic Pattern Equivalence

Specific patterns have different kinds of lexico-syntactic constraints depending on each constituent. The set of applicability conditions of a pattern to a new sentence will be all lexico-syntactic contraints of all contituents. Without other extensions, if one would recognize the following two sentences, two different specific patterns would be needed.

```
(sentence-number
  (syntactic-constituent1 (POS word11) ... (POS word1n))
  (syntactic-constituent2 (POS word21) ... (POS word2m))
  ...
  (syntactic-constituentN (POS wordN1) ... (POS wordNz)))
```

Figure 4: Specific pattern format

```
''At least one person has been killed by an avalanche in the
Italian Alps.''
''An avalanche in the Italian Alps killed at least one person.''
```

When a syntactic pattern is generated, a set of equivalent syntactic patterns is obtained by creating syntactic variations of it which are able to recognize the same information as using the original pattern. For example, from a pattern in active voice, a variation on passive voice is created.

If semantic generalization is done, is it necessary or even useful creating syntactic variations? Probably it is only useful to create pasive/active syntactic variations in order to determine the agent and the object of the phrase[10]. The basis of this reasoning is that syntactic constituents doesn't enforce any specific order - it is only important the presence of each syntantic constituent along with its semantic feature. This means that, for instance, a pattern having two PP constituents semantically labeled as [time] and [location] respectively, it is equivalent to other one also having two PP constituents semantically labeled as [location] and [time], i.e presented in reverse order.

[10]We are studying this issue already discussed in NYU system for MUC-6 description [6].

4.3 Generalizing Patterns

One specific pattern recognizes one concrete sentence from text, i.e. it uses the same words found in text to represent the pattern applicability conditions. It is difficult to know crucial pattern characteristics, according to the goal extraction, in advance. The essential idea of the generalization process is to get extraction patterns able to extract the same information as specific patterns without adding irrelevant information. As a side effect extraction patterns base is reduced.

The input to the generalization process is the specific pattern base obtained in the first step. The algorithm is intended to find generalizations that cover a set of specific patterns, i.e. each generalized pattern must extract the same information as the collection of specific patterns it covers.

The generalization process takes into account different features of the elements of the patterns, that are presented as generalization rules as follows:

R.1 - Semantics of the syntactic constituents:
Generalizing from semantic features of syntactic constituents.

This task requires in addition to a lexicon some mechanism that links semantic features to each word[11] of syntactic constituents as well as a semantic hierarchy. A semantic feature is a semantic class that covers some subclasses while is covered by other classes. The set of relationships between semantic classes is given by the semantic hierarchy that allows rules over semantic class constraints. Once again WordNet is a valuable resource for this task.

The generalization of a syntactic constituent, that has a semantic class A in a pattern and B in another one, would be [A $\vee$ B .. T]. This expression means that generalized pattern has as possible semantic features A, B and all A and B hyperonyms found below T, where T is the first node in the hierarchy that subsumes both classes.

R.2 - The presence of syntactic constituents:

[11]Currently, we are tagging nouns and verbs appearing on each syntactic constituent because the determination of the head is an outstanding matter.

The idea is to remove those syntactic constituents that differs a very similar pattern set. From this, the resulting pattern will cover all similar patterns excluding variable features considered to be irrelevant at first.

The generalization algorithm will be incremental to allow dealing new documents of same domain without redoing the work done. It will allow handling negative examples also because, as we will see, the human expert could label the training corpus examples erroneously covered by a generalized pattern as negative examples.

Besides the structure of the generalized pattern being built the set of training corpus sentences that it covers is maintained. So that, for each pattern, together with its description we have the examples from corpus it covers.

As option the expert can give information to guide the learning algorithm pointing out the elements that should become part of a pattern. For instance, he knows that some verbal forms are highly related with the kind of information to extract. In Machine Learning this is known as a *bias* to accelerate the learning process.

4.3.1 Generalized Pattern Representation

The syntax description of generalized patterns is shown in Figure 5.

Semantic features linked to each word are represented by senses. A semantic feature correspond to a byte offset or address of a *synset*[12] extracted from WordNet. At this moment, semantic features are reserved to nouns and verbs.

4.3.2 Example of generalization

This subsection will serve us to describe to a major extent the generalization process by tracing an example. The starting point is a reduced specific pattern base that only contains two specific patterns.

Figure 6 shows the two specific patterns seen in Figure 3 once semantic tagging has been done. Only nouns and verbs have semantic features attached because generalization is mainly done by the hypernymy relationship,

[12] A synset or synonym set, is a list or synonymous word forms that are interchangeable in some syntax.

```
(sentence-number

  (syntactic-constituent1 (POS word11 sense111 ... sense11a)
                          (POS word12 sense121 ... sense12b)
                          ...

                          (POS word1n sense1n1 ... sense11z))

  ...

  (syntactic-constituentN (POS wordN1 senseN11 ... senseN1X)
                          (POS wordN2 senseN21 ... senseN2Y)
                          ...

                          (POS wordNz senseNz1 ... senseNzZ)))
```

Figure 5: Generalized pattern format

which has no sense in remainder syntactic categories (adjectives, adverbs and closed class words).

Once specific patterns has been tagged with semantic features from WordNet, rules of generalization process are applied over them. In Figure 7 we can see the generalized patterns obtained from the two specific patterns with semantic features seen in Figure6.

In this case `person` has a semantic feature [4311] and `man` has a semantic feature [3977410]. Generalizing from semantic features of syntactic constituents we have a semantic feature ranging over [4311 $\vee$ 3977410 .. 4311], because [4311] is the first node in the semantic hierarchy that subsumes both them. Reducing the range gives us to [4311]. From remaining syntactic constituents, the first PP has as semantic feature [4310904] (`Alps` and `Mountains` have it as ancestor in common) shared by two specific patterns, and also the second PP has a semantic feature [3551221] (`crash` and `avalanche` have it as ancestor in common) shared by two specific patterns. The semantic feature of VP constituent is not generalized because it's the same in the two patterns. In this case there are no exclusive syntactic features therefore the second rule doesn't apply.

4.3.3 Fitting Function for the Generalization Process

While general patterns can improve a better *recall*, specific patterns can improve a better *precision*. This leads us to define a parameter or function that contrains the degree of generalization of a concept besides the expert's considerations. The constraints move on different levels of the hierarchy (WordNet, in our case): neither too general levels nor too specific.

This kind of function is defined as a fitting or adjustment function to different levels of generalization. Its purpose is to close the system perfomance to the user's usage. But there are also some questions (not just answered) on the air like, it must be the fitting function adjustable by the user?, by feedback?. For the present we have studied the possibility of maintaining different degrees of generalization in the same pattern depending on concept to be generalized because there could be specific concepts that not are advisable to generalize while others surrounding them are.

Sentence 72: At least one person has been killed in an avalanche in the Italian Alps.

```
(72
  ((NP ((NOUN AT_LEAST NIL) (NOUN ONE (6101912))
        (NOUN PERSON (3180212 4311))))
   (VP ((AUX HAS NIL) (AUX BEEN NIL)
        (VERB KILLED (979306 539894 483406 202425 150771))) )
   (PP ((PREP IN NIL) (NPST AN NIL) (NOUN AVALANCHE (3587260))))
   (PP ((PREP IN NIL) (NPST THE NIL) (NOUN ITALIAN (4441685 3431667))
        (NOUN ALPS (4309198))))))
```

Sentence 294: Five men were killed this weekend in helicopter crash in the Rocky Mountains of Southeastern B-C.

```
(294
  ((NP ((NOUN FIVE (6102936 3936049)) (NOUN MEN (3977410))))
   (VP ((AUX WERE NIL) (VERB KILLED (979306 539894 483406 202425 150771))
        (ADV THIS_WEEKEND NIL)))
   (PP ((PREP IN NIL) (NOUN HELICOPTER (2070805))
        (NOUN CRASH (3583509 3557432 49423))))
   (PP ((PREP IN NIL) (NPST THE NIL) (NOUN ROCKY NIL)
        (NOUN MOUNTAINS (4310904)) (NOUN OF NIL) (NOUN SOUTHEASTERN NIL)
        (NOUN B-C NIL)))))
```

Figure 6: Semantic tagging of specific patterns from sentences 72 and 294.

```
(
  (72 294)
    ((NP ((NOUN (PERSON MAN) (4311))))
     (VP ((VERB KILLED (979306 539894 483406 202425 150771))))
     (PP ((PREP IN NIL) (NOUN (AVALANCHE CRASH) (3551221))))
     (PP ((PREP IN NIL) (NOUN (ALP MOUNTAIN) (4310904)))))
)
```

(4311: person, individual, someone, man, mortal, human, soul; "a human being")
(3551221: happening, occurrence, natural_event)
(4301051: mountain, mount; "a land mass that projects well above its surroundings; higher than a hill")

Figure 7: Generalized pattern for sentences 72 and 294.

4.3.4 Generalization of Different Types of Constituents

The essential idea of generalizing different types on constituents separately is explained with an example.

It doesn't make sense to generalize a sentence which has the keyword *crash* as a noun in a NP constituent, starting with its verb in a VP constituent. The following example illustrates this idea, where there's a *crash* as noun in a sentence and *crash* as verb in another one:

```
''The crash caused two victims.''
''A F14 has crashed in the Italian Alps.''
```

Clearly, the generalization of a keyword concept might take into account its POS along with its type of constituent. We will split up generalizations concerning different types of constituents (NP, PP, VP...) in different sets:

- NPs set: sentences containing the keyword in a NP group.
- VPs set: sentences containing the keyword in a VP group.
- PPs set: sentences containing the keyword in a PP group.
- and others.

4.3.5 Characterization of Components of a Pattern

Different components of a pattern aren't all equally relevant for extraction purposes. Some of them will produce useful generalizations while others will be maintained as specific as possible due to its key significance. Since it is no clear how to randomly apply the specificity criterion over a pattern component, there will be an special mark to characterize it. We are studying three values for this mark which are:

- Generalizable components
- Optional components
- Specific components

4.4 Filtering Patterns

The amount of extraction patterns even after the generalization process may be still very large. Some patterns will be useless for the extraction of domain relevant information and some others will be spurious. To solve this problem the generalized patterns are put on a filtering process.

Two possible filtering processes might be applied:

- Filtering by frequency: This process is intended to remove generalized patterns with reduced applicability. A minimum threshold of applicability is decided in order to throw out those patterns not surpassing this level. The idea is then keep up patterns that are useful at least in a minimum degree.

- Filtering by relevancy: This process only can be done if a set of erroneous or irrelevant extraction examples is available. As will be pointed out later, a human expert of domain will supervise the patterns setting which examples covered by a pattern should not. Thus, he notes that the pattern either is wrongly generalized or it extracts irrelevant information.

The filtering by relevancy aims at decide which extraction patterns are actually relevant ones and discard those not surpassing a given threshold of relevance. The relevance of a pattern is the percentage of actual examples it covers which were relevant.

4.5 Validating Patterns

At this point, the volume of initial patterns has been substantially reduced then the cost of review process as well. The hypothetical pattern set must be validated by a human expert. If he decides that a pattern is too general, he can note which examples related to the pattern should not be covered by it. These examples will serve as negative examples in the feed-back generalization process.

Moreover the expert can consider that the volume of pattern is still excessive and modify the parameters of filtering methods and/or the bias of generalization process. The process is repeated until the expert approval, in which moment begins the next step.

In the generalized pattern from example above, an expert can decides that the pattern is too general, because it doesn't allow to extract the instant of the accident. He can note the specific patterns wrongly covered by the generalized pattern (Figure 7) as negative examples and repeat the generalization process.

4.6 Typifying Patterns

Until this phase, once hypothetical patterns base have been validated, the expert has yet fixed the concrete kind of information to extract from each pattern.

Typifying patterns lies in "give names" (that, in fact, are roles they play) to different pattern components, indicative of kind of information they will extract. For instance, coming back to terrorism domain, a pattern representing the set of sentences [person] `was assassinated` will typify [person] as [VICTIM]. [VICTIM] is the role played by [person] in the pattern and represents the kind of information to extract in this domain.

Almost all approaches presented in this work do typification manually, either from semantically annotated corpus, from answer keys, by interactively defining events or typifying patterns a posteriori. The issue is the same: an expert must be required for this task. In section 6, we propose an alternative to manual typification of patterns as a future work matter.

The belated typification of patterns has an advantage, added to the smaller volume of information to deal with, because it makes easier reusing

patterns to other extraction related tasks. Text classification [20], text summarization [24], constructing specific lexicons that includes contextual information or building word sense disambiguation tools [10], are some examples.

4.6.1 Typification: Usage and Limitations

Typification[13] of patterns guides the process to extract the required information. It determines which component of a pattern will fill which slot of the final output template. Information extracted says exactly what the text says (using exactly the same words) but is structured in the form of a template.

From this point view typification is a good method, but it also restrains the domain of the text and the kind of information to extract. So that, it is very important do typification as later as possible.

Typification is also a previous stage to transform patterns into templates. A pattern will be linked to a template-like concept to capture the semantic content of a pattern and transformed into an extraction rule. For each pattern, the extraction rule specify which slots should be filled by which kind of information.

To cite here some other example of typification Figure 8 shows the pattern representation used by the CRYSTAL system. A pattern in CRYSTAL, named *concept node*, not only has syntactic and semantic constraints, but also includes typification. Typification is manually done on the training corpus by means of human expert annotations. Every concept being learned has been explicitly marked in the text and these marks are labels representing a phrase's role in a target concept (shown as Extract slot in the figure).

This concept definition for the Management Succession domain has constraints to identify instances that have a `Person_In`, found in the subject, to a `Position`, found in the direct object. Applied to a sentence like "Paul Herold was recently named chairman of this major farmaceuticals concern", will identify "Paul Herold" as a `Person_In` to `Position` "chairman of this major farmaceuticals concern".

The CRYSTAL's extraction rules signal the concrete template or case frame slot to fill in. The value to fill the slot will be the syntactic constituent where the rule is found.

[13] In some systems typification is known as "logical labeling" because it assigns so-called logical labels like [VICTIM] in the above example.

```
Concept type: Succession event
Constraints:
   SUBJ::
     Classes include: <Person Name>
     Extract:         Person_In
   VERB::
     Root:            NAME
     Mode:            passive
   OBJ::
     Classes include: <Corporate Post>
     Extract:         Position
```

Figure 8: CRYSTAL concept node definition.

4.7 Pattern Instantiation and IE Systems Output

During scanning of new information, with the help of a pattern matching procedure, the system applies general patterns on a large number of unseen articles from the domain.

The output generated by an IE system may consist in a *mark-up*, i.e. textual items in the document are bracketed or attached with semantic labels, or in a set of templates to represent complex logical structures according to a predetermined format.

Coming back to MUC competitions, different tasks done in them have variable representations.

- NE, CO, TE: first are marked with Tipster annotations[14] and then converted to SGML.
- ST: this task requires a certain amount of inferencing to extract the actual events from those explicitly stated in the document, e.g. "Ford was the president of Lott Inc. Ford was succeeded by Heint" we need to infer that Heint is becoming the president of Lott. Thus, we need

[14] A Tipster annotation includes a type, a set of start/end byte offsets, and a set of attributes.

some kind of mechanism to do so like predicates (e.g. "succeds(person1, person2)") used in [6].

4.8 Postprocessing

After patterns and extraction rules are generated and instantiated there is some more processing needed.

- To cut out extraneous words: some irrelevant words or processing marks, like commas or punctuation marks, must be removed from patterns.

- To recognize different references to the same entity: an entity can be referred using different names or alias. A postprocessing has to recognize them and substitute multiple references by the actual entity.

- To replace pronouns with the actual names: pronouns substitutes persons or entities in text and they are maintained in patterns without changes. It is necessary identify which person or entity represent each pronoun to replace it with the actual name.

- To merge template outputs conveniently: a template relationship covers relationships among template elements captured in the form of template "relations" consisting of a relationship and the template elements participating in that relationship. A scenario template requires identifying instances of a task-specific event and identifying event attributes, including entities that fill some role in the event. Both tasks need merging mid-level template outputs conveniently to generate the final template output.

4.9 Further Improvements

Each step of the ESSENCE methodology assumes the existence and makes use of different knowledge resources and tools. The selection of one or another may significantly modify IE system performance. This choice must keep in mind robustness even if portability is more important than other issues. In this sense, we have stated that, some exhaustive linguistic knowledge resource such as WordNet is needed. We require: 1) large lexical knowledge

covering whatever domain vocabulary, 2) lexico-semantic relations to make up semantic generalization and 3) low level syntactic features.

On the other hand, WordNet (or any other lexical resource) is not the unique participant when building an IE system up. In our case, we need a sentence analyzer, that depends strongly not only on its lexicon but also on its grammatical rules. A bad tagging induces an erroneous sentence analysis that will generate wrong extraction patterns, and so on. Our experiments reveal too much errors in the first steps of the sentence analysis due to poor flexible syntactic tagging. This leads us to think about replacing or improving our syntactic analysis module.

Other improvements refer to the *right* determination of the head of each syntactic component of a pattern in order to restrain the generalization process, and a mechanism to filter out irrelevant senses given a specific domain.

5 Extensions

This section extends further improvements and future works but can be included as a part of any of them.

5.1 Customization of IE Systems

First of all, we have to examine which components of a system need to be modified when an existing system is to be customized for a new domain. The modularity presented by an IE system is an important issue when moving the domain of a system.

Generally, customization is restricted to the knowledge resources. Among these, the pattern base is by far the most complex and will require the greatest effort to customize (fitting and revision will be needed). The lexicon and semantic hierarchy, using existing general tools like WordNet, will be useful to many domains. Customizing the pattern base depends on the complex structure patterns will be represented and on manual work. Several groups have attempted to face up the portability of a system making it human expert independent, i.e. trying to make the domain adaptation by non-expert users possible. This is done by keeping the interaction as high-level as possible like in CRYSTAL.

Looking-up developed approaches we have stated two solutions:

- To use unsupervised learning: but is known to be dependent on the availability of the large amount of manually prepared training data.
- To use an example-based strategy for pattern building

Not so far of customization of IE systems is the reusability property of a pattern base. We have to study if a pattern base can be organized so that one can distinguish between patterns specific for a domain and patterns applicable to whatever domain. It is intended to have some sort of pattern repository in which:

- Specific patterns (at the top): able to capture determinated events.
- Libraries of patterns (in the middle): it should be created depending on the domain of application.
- More general patterns (at the bottom): fixed for all domains, syntactic patterns are included in this level.

5.2 Preclassification of Relevant Text for Training

At text level, a filtering is done to determine relevance of text or parts of text based on word statistics or the ocurrence of keywords.

Classification of relevant text searching by keywords has a drawback because two synonym or partially synonym keywords are found to be different words and therefore its statistics are separated instead of counted together.

5.3 Problems of MUC Oriented IE

MUC oriented IE aims to identify and extract all information pertaining to a set of prespecified events, within a limited domain, and store that information in prestructured templates.

In many domains, factual information is often all that is needed. However in other domains, non-factual information is often just as important. For example, in traditional IE one should extract factual information from finance and military domains; on the contrary, one should extract non-factual information from culture and entertainment domains (e.g., about movies) [11].

6 Conclusions and Future Work

This work proposes a methodology, named ESSENCE, to automatically acquiring extraction pattern bases for IE systems. The purpose is twofold: 1) to avoid the effort in preparing a training text corpus and 2) to reduce the human expert intervention when acquiring general extraction patterns.

The ESSENCE methodology comprises five basic steps to which one may add different options to guide the learning strategy. From first step a set of specific pattern set is available being used as examples in the following learning or generalization process. The generalization process is automatic and neither the relevance of patterns nor its utility is ensured, for that reason the patterns are submitted to a filtering process. The relevance of a pattern depends on the extraction purposes and who determines what extract is a human expert. Thus, the expert is required for patterns validation, iterating the generalization process if need be, and for pattern typification, stating the kind of information to extract.

The goal of extraction patterns is identify and extract relevant information from a document. As we have seen, the decision about what a pattern has concretely to extract is not taken until typification process. But typification is nothing more than mapping each element of a pattern to its role in context (other elements of pattern surrounding it). To find out a role in context requires deep knowledge but is possible acquire it if a system that could express the concept the role represents is available. Conceptual representation of a role in context along with a mechanism that allows instance classification (pattern classification, in this case) above a suitable concept, could be enough to achieve automatic typification. A system having these features is YAYA [1] and with it we will address the automatization of typifying process.

References

[1] Jordi Àlvarez. Yet another yet another (YAYA). Technical Report LSI-96-15-T, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 1996.

[2] J. Atserias, N. Castell, N. Català, H. Rodríguez, and J.Turmo. Del texto a la información. *Novatica*, 133, May-June 1998.

[3] Neus Català and Núria Castell. Construcción automática de diccionarios de patrones de extracción de información. *Procesamiento del Lenguaje Natural. Actas de SEPLN'97*, 21:123–135, July 1997.

[4] Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39(1):80–91, 1996.

[5] R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. University of Sheffield: description of the LaSIE system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 207–220, San Francisco, California, 1996. Morgan Kaufmann.

[6] Ralph Grishman. The NYU system for MUC-6 or where's the syntax? In *Proceedings of Sixth Message Understanding Conference (MUC-6)*, pages 167–175, Columbia, MD, November 1995. Morgan Kaufmann.

[7] Ralph Grishman and Beth Sundheim. Message Understanding Conference - 6: A brief history. In *Proceedings of 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen, August 1996.

[8] Scott B. Huffman. Learning information extraction patterns from examples. In *IJCAI-95 Workshop on New Approaches to Learning for NLP*, 1995.

[9] W. Lehnert, J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan, and S. Goldman. UMass/Hughes: Description of the CIRCUS system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference*, 1993.

[10] Xiaobin Li, Stan Szpakowicz, and Stan Matwin. A wordNet-based algorithm for word sense disambiguation. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

[11] Alpha Luk, Barbara Vauthey, and Olivier Ansaldi. Acquisition of domain specific salient semantic features for information extraction. In

Proceedings of the International Workshop on Lexically Driven IE, Frascati (Italy), July 16th 1997.

[12] R.S Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, 20:111–161, 1983.

[13] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to wordNet: An on-line lexical database. Technical report, Cognitive Science Laboratory, Princeton University, 1993.

[14] T.M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[15] *Proceedings of the Third Message Understanding Conference (MUC-3).* Morgan Kaufmann Publishers, 1991.

[16] *Proceedings of the Fourth Message Understanding Conference (MUC-4).* Morgan Kaufmann Publishers, 1992.

[17] *Proceedings of the Fifth Message Understanding Conference (MUC-5).* Morgan Kaufmann Publishers, 1993.

[18] *Proceedings of the Sixth Message Understanding Conference (MUC-6).* Morgan Kaufmann Publishers, 1995.

[19] Ellen Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1993.

[20] Ellen Riloff. Using learned extraction patterns for text classification. In G. Scheler S. Wermter, E. Riloff, editor, *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pages 275–289. Springer-Verlag, 1996.

[21] Ellen Riloff and Jay Shoen. Automatically acquiring conceptual patterns without and annotated corpus. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 148–161, 1995.

[22] Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

[23] Jordi Turmo, Neus Català, and Horacio Rodríguez. TURBIO, a system for extracting information from restricted-domain texts. In *Proceedings of the Eleventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA-98-AIE)*, Lecture Notes in Artificial Intelligence. Springer Verlag, 1998.

[24] Klaus Zechner. A literature survey on information extraction and text summarization. Paper for Directed Reading, Fall 1996.