# TURBIO: A System for Extracting Information from Restricted-domain Texts

J. Turmo, N. Català and H. Rodríguez

Dept. Llenguatges i Sistemes Informàtics*.
Universitat Politècnica de Catalunya
c/ Jordi Girona Salgado 1-3
E-08034 Barcelona - Spain

**Abstract.** The more extended way of acquiring information for knowledge based systems is manually. However, the high cost of this approach and the availability of alternative Knowledge Sources has lead to an increasing use of automatic acquisition approaches. In this paper we present TURBIO, a Text-Based Intelligent System (TBIS) that extracts information contained in restricted-domain documents. The system acquires part of its knowledge about the structure of the documents and the way the information is presented (i.e. syntactic-semantic rules ) from a training set of them. Then, a database is created by means of applying these syntactic-semantic rules to extract the information contained in the whole documents.

## 1 Introduction

The more extended way of acquiring information for knowledge-based systems is manually, frequently by means of a dialog between the system and the human expert (sometimes with the intervention of a knowledge engineer). However, the high cost of this approach, together with the availability of alternative Knowledge Sources has lead to an increasing use of automatic acquisition approaches.

Special interest present the Text-Based Intelligent Systems (TBIS), in which the knowledge to be extracted is contained in documents and must be extracted from them. When these documents have been produced for computing use they use to be highly structured and extracting information from them can be carried out quite straightforwardly, but frequently the documents have been produced for human use and lack an explicit structuring. In this case they consists of an unrestricted Natural Language text and the task of extracting information involves a great deal of linguistic knowledge in order to be performed. Sometimes, and it is the case of our proposal, documents present a mixed structure where chunks of NL text appear together with more structured codified pieces of information.

* This work has been partially funded by CICYT (TIC96-1243-c03-02, ITEM project) and by CIRIT (1997SCR51)

The techniques and methodologies for extracting information from unrestricted NL text conforms what is called Information Extraction (IE), (see [9] for an in depth survey or [3] for an introductory overview).[2]

Most IE systems are related to the MUC competitions (see [7] for a survey of MUC-6) although there are also notable examples outside. JASPER [1], FASTUS [2], LASIE [5], SRA [8], PLUM [12], NYU [6] and UMass [4] are some of the most known systems. In last years several programs have grown funded by EU. Between them ECRAN[3], AVENTINO[4] and SPARKLE[5].

TURBIO is a system, including both a methodology and a computer environment supporting it, that extracts information from semi-structured texts. TURBIO can deal with unrestricted text but also takes profit of codified pieces of information both for extracting this information and for guiding the extraction process elsewhere. The basic requirement is that a grammar could be built for allowing the extraction of codified parts and the location of chunks of NL text. TURBIO has been applied in the domain of mycology for extracting information from cards [10] describing mycological species[6]. The extracted information has been then used as knowledge base of the expert system KINOKO[11] that classifies unknown species from their features.

## 2 Functionalities

A main issue in IE systems is the definition of the extraction rules. Generally, extraction rules are represented by pairs <keyword,template-set>, where *keyword* refers to a domain concept and each template of the *template-set* represents the set of modifier features for that concept. Extraction rules are usually defined manually. TURBIO proposes, however, the acquisition of extraction rules using a learning process based on an analysis of a training corpus. The resulting rule set (containing the knowledge about corpus structure and the way of describing information) will be then used to perform the information extraction process.

The first functionality of TURBIO is then to extract patterns of relevant information contained in texts belonging to specific domains using a domain structured representation. The basic units to be extracted are triples <entity attribute value>.

The second functionality consists of applying these rule sets for extracting information from documents. Briefly, once the text has been preprocessed and shallow-parsed, the system looks for a keyword and its

[2] IE is an emerging technology and must not be confused with the more mature area of Information Retrieval (IR), that giving a query, tries to select a relevant subset of documents from a larger set.

[3] http://www2.echo.lu/langeng/en/le1/ecran/ecran.html

[4] http://www2.echo.lu/langeng/en/le1/aventinus/aventinus.html

[5] http://www2.echo.lu/langeng/en/le1/sparkle/sparkle.html

[6] Although our corpus consists of bilingual (Spanish/Catalan) non-parallel descriptions, only texts in Spanish have been used here.

modifiers in the parse forest and activates the associated template covering more modifiers.

Following, we explain the TURBIO methodology to get both functionalities.

## 3 Architecture

Figure 1 presents an overview of TURBIO architecture. TURBIO builds a structured representation of the documents (DB)[7] using a grammar for describing the document structure. A representation of the domain must be provided too.

The result of TURBIO performance is a domain knowledge base (DKB) represented in a typed feature structure formalism containing instances of entities of the domain owning the information extracted from texts.

The first functionality of TURBIO -extraction rule set acquisition- is performed by three modules of the system: S-BUILDER, P-BUILDER and KERNEL. S-BUILDER gets chunks of shallow-parsed trees, generalizes them in *syntactic-semantic pattern schemata* using GENERALIZER module and produces the set of relevant pattern schemata in the corpus. P-BUILDER is used to build relevant *syntactic-semantic patterns* from schemata trying to fix variables in a schema with common information of the chunks it represents. Finally, KERNEL builds the extraction rule set.
The extraction functionality of TURBIO is performed by KERNEL. It produces feature structures by applying extraction rules over the shallow-parsed chunks.

Next sections gives details of all these modules.

## 4 S-BUILDER module

The aim of this module is to extract all relevant syntactic-semantic pattern schemata occurring in the corpus. An schema means a representation of the set of chunks and subchunks having the same syntactic-semantic parse tree. S-BUILDER runs in three steps: 1) a pre-process in which morphological and shallow syntactic parsing is performed for getting shallow-parsed chunks of phrases, 2) a process to find syntactic-semantic pattern schemata from those chunks (GENERALIZER module) and 3) a process to select from them relevant pattern schemata. For instance, in the pre-process, the phrase:

> *"Crema amarillento y de carne oscura."* (Yellowish cream with dark flesh)

is morphologically parsed as:

> n( *"crema"*) a( *"amarillento"*) c0c( *"y"*) r0a( *"de"*) n( *"carne"*) a( *"oscura"*) zpunt( *"."*)

[7] Part of DB is used as training corpus during the learning process.

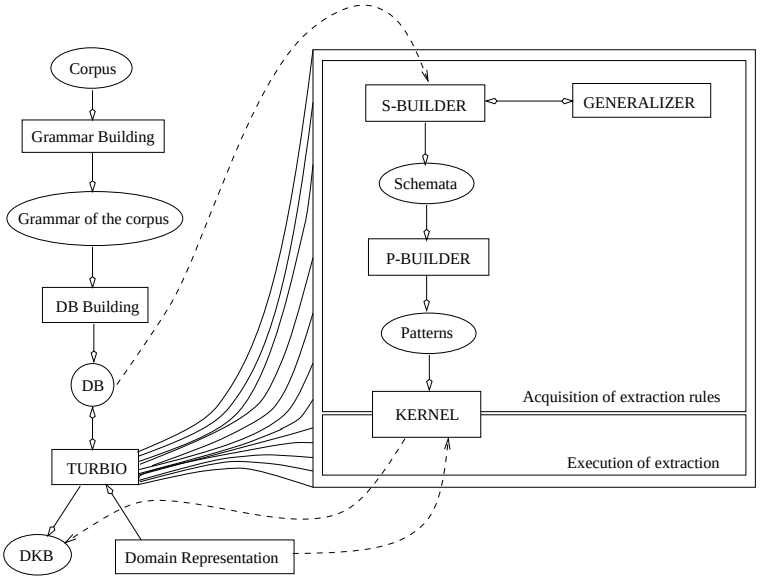


**Fig. 1.** Environment and architecture of TURBIO

and shallow parsed as:

1. (gnom ((n *"crema"*)(a *"amarillento"*)))
2. (c0c *"y"*)
3. (sp ((r0a *"de"*)(gnom ((n *"carne"*)(a *"oscura"*)))))
4. (zpunt *"."*)

where each line is a chunk and each subtree of a chunk is a subchunk. Analyzing these chunks we find the pattern schemata:

a. (gnom ((n [$R_1$:$S_1$]-$W_1$) (a [$R_2$:$S_2$]-$W_2$)))
b. (sp ((r0a $W_1$) (gnom ((n [$R_1$:$S_1$]-$W_2$) (a [$R_2$:$S_2$]-$W_3$)))))

(a) represents both the chunk (1) and the second subchunk of chunk (3), where $W_i$ contains the possible words in the schema (e.g. $W_1$={ *"crema" "carne"*}), $S_i$ the list of semantic values for $W_i$ and $R_i$ relations between semantic values and words[8].

### 4.1 Defining $\delta$-sets

In order to find relevant patterns some relationships between schemata must be introduced. Schemata have been coded assigning to each syntactic label a prime number and multiplying the codes of each syntactic label in the schema ($\delta$-code). For example, the schema:

$r$ =(gnom ((n [$R_1$:$S_1$]-$W_1$) (a [$R_2$:$S_2$]-$W_2$)))

has $\delta(r) = 30$ when we codify *gnom* as 2, *n* as 3 and *a* as 5. This coding allows to define the equivalence relation $=_\delta$ between schemata as follows:

[8] Only names, adjectives and verbs own $R_i$ and $S_i$.

*Let* $\mathcal{S}$ *the schemata set,*
$r, s \in \mathcal{S}$, $\delta$(r)= $\delta$(s)$\rightarrow$r=$_\delta$s

Now it is possible to define the quotient set $\mathcal{S}|_{=_\delta}$. In this way, a $\delta$-set will be an element of $\mathcal{S}|_{=_\delta}$. For example, both schemata:

$r$ =(gnom ((n [$R_1$:$S_1$]-$W_1$) (a [$R_2$:$S_2$]-$W_2$)))
$s$ =(gnom ((a [$R_1$:$S_1$]-$W_1$) (n [$R_2$:$S_2$]-$W_2$)))

belong to the same $\delta$-set.

### 4.2 Finding relevant pattern schemata

This module finds the relevant set of pattern schemata. As we can see, schema (1) of the example of this section is included in schema (2). It is possible to define a relation between schemata, called *covering* relation:

$r, t \in \mathcal{S}$ , *t covers* $r \leftrightarrow$ *r covered by* $t \leftrightarrow t \sqsubseteq r \leftrightarrow$ *t includes r*

We can classify schemata in two classes: those derived from chunks and those derived from subchunks (Fig. 2). The intersection of both sets is not empty. In order to get the set of relevant schemata we must study those pairs of high frequency schemata covering-related. Then, all high frequency schemata not covered by any other will be relevant.

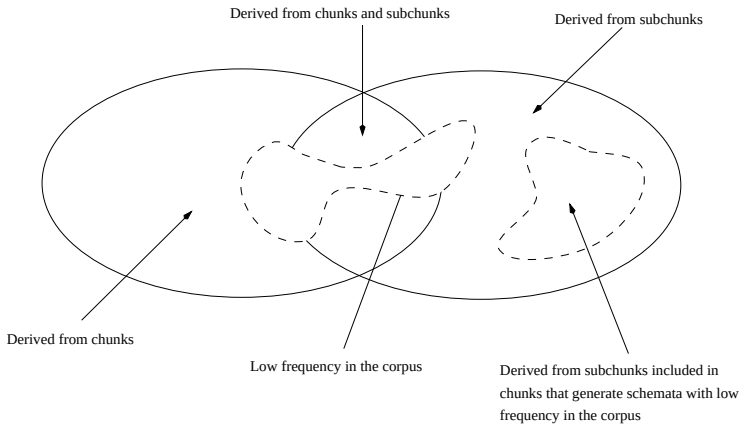


**Fig. 2.** Classification of schemata according to their derivation

We can represent covering relation between schemata using the $\delta$-sets defined in section 4.1. A link relation exists between pairs of $\delta$-sets when there is, at least, one schema in one of them covered by another schema of the other $\delta$-set (linked $\delta$-sets). This link is verified by: two $\delta$-sets $\delta_a$, $\delta_b$ are linked-related iff $\delta_a \equiv_{\delta_b}$.
In figure 3 $\delta$-sets 30 and 210 are linked because schema $t$ covers schema $r$ being 210 module 30 equal to zero. So, the method to get relevant schemata consists of:

- Drop out schemata without nominal, verbal or adjectival labels.
- Drop out schemata with low frequency in the corpus ($f_s < 10$).
- Lessening of $\delta$-sets: residual frequency of a schema is the difference between its frequency in the corpus and the sum of frequencies in the corpus of all other schemata covering the first one. The lessening of $\delta$-sets means to drop out all schemata having null residual frequency. We are selecting only schemata uncovered by more specific ones.

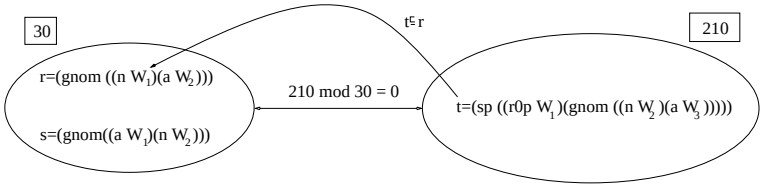


**Fig. 3.** Linked $\delta$-sets

### 4.3 GENERALIZER

Chunks found can be seen as specific syntactic patterns. Each constituent of the pattern matches specific words (e.g., *"pie"*, *"blanco"*), and specific syntactic properties of these words (e.g., n(ame), a(djective)).

Observing the set of specific patterns we note that two or more patterns, having the same syntactic restrictions, have also semantic properties in common but different specific words to apply to. This suggests that these patterns can be generalized using semantics.

Such assumption implies that we have semantic knowledge about our language or, at least, about the sub-language the application domain deals with. Our approach makes use of an Spanish WordNet (part of EuroWordNet)[9] as a basis of semantic knowledge because its representation as a semantic net allows us to reason about different types of semantic relationships between concepts. In addition, for simplicity, we consider that a conceptual representation corresponding to the domain vocabulary is available.

Figure 4 shows, partially, the conceptual representation in WordNet and the label meanings (relationships) of an abbreviated mushrooms vocabulary.

The generalization of two specific patterns starts looking for a more general concept covering both concepts. We need that all constituents could be generalized at some level of the hierarchy. If it is not possible, the specific patterns are maintained.

[9] http://www.let.uva.nl/~ewn

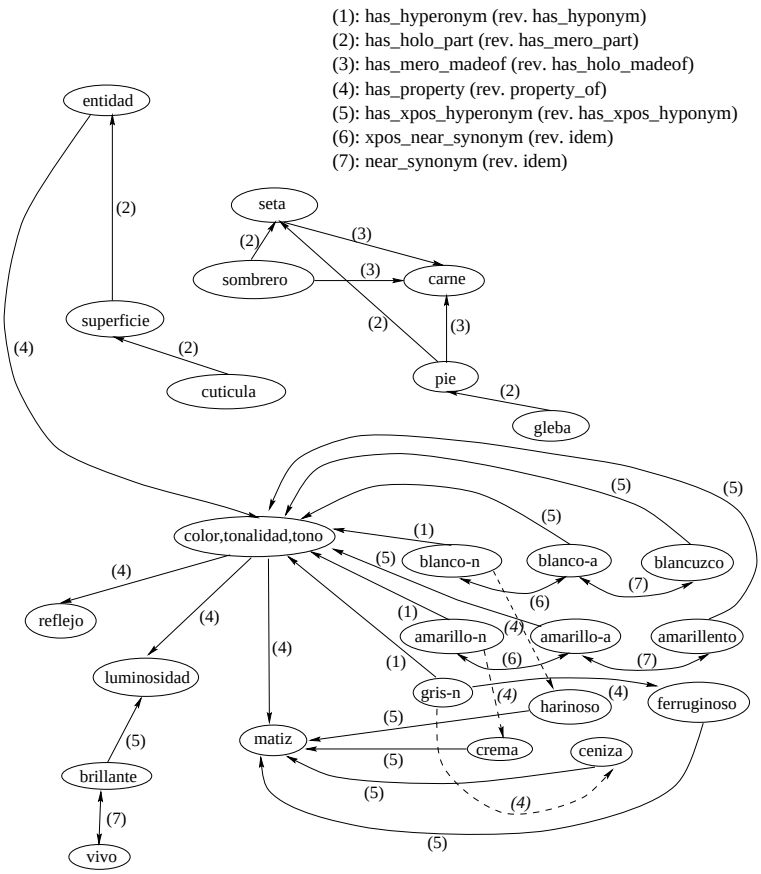


**Fig. 4.** Spanish WordNet (abbreviated mushrooms sub-language).

Since each specific concept may have several generalizations using a variety of relationships, a method for reducing the problem is needed. The search of the ancestor concept moving up in the hierarchy is guided by different rules depending on the relationships present in the domain semantic representation.

The easiest way of obtaining ancestors is provided by the *hyperonymy* relationship (and its homonym relation across different *part of speech*, *xpos-hyperonymy*) but there are also other relationships, such as the variety of types of *meronymy*, that allow generalization if they are controlled. Some relations shown in Fig. 4, play the role of restrictions with regard to the application of generalization rules. For instance, *hyperonymy* relationship is restricted by the existing *has-property* relationship between two concepts, while generalizing them.

The next example shows partially the generalization process applied to a set of specific patterns provided by the previous module (S-BUILDER).

(gnom ((n *"crema"*) (a *"amarillento"*)))
(gnom ((n *"gris"*) (a *"ferruginoso"*)))
(gnom ((n *"pie"*) (a *"blanco"*)))
(gnom ((n *"blanco"*) (a *"harinoso"*)))
(gnom ((n *"ceniza"*) (a *"blancuzco"*)))
(gnom ((n *"sombrero"*) (a *"rosado"*)))

Suppose that the chunk (specific pattern) list above is a complete list of all chunks built from the test text. Starting with the first two chunks there is no generalization possible because semantics of *"crema"* (that is a MATIZ) and semantics of *"gris"* (that is a COLOR) doesn't have a common hypernym concept in any level of the hierarchy.

Between the first three chunks there is no generalization possible for similar reasons. But when we consider the fourth one, a generalization is possible between chunks two and four. Semantics of *"gris"* (that is a COLOR) and semantics of *"blanco"* (that is a COLOR) have a common concept in the hierarchy, and semantics of *"ferruginoso"* (that is a MATIZ) and semantics of *"harinoso"* (that is a MATIZ) have also a common concept in the hierarchy. We represent the resulting generalized pattern as follows:

(gnom ((n [HYPONYM:COLOR]-$W_1$) (a [HYPONYM:MATIZ]-$W_2$)))

Following the example it is also possible a generalization between chunks one and five, and between three and six. The former generalization results in:

(gnom ((n [HYPONYM:MATIZ]-$W_1$) (a [HYPONYM:COLOR]-$W_2$)))

and the last generalization results in:

(gnom ((n [ISPARTOF:SETA]-$W_1$) (a [HYPONYM:COLOR]-$W_2$)))

Finally, each generalized pattern will be converted into its representation as a syntactic-semantic pattern schema, i.e. changing concepts and relationships for variables and maintaining the original information using linked sets (LSs) as explained above.

Different rules are applied on different specific patterns in order to generalize them and the resulting generalizations could be too general for the purposes of the type of information to be extracted. We need a domain-expert detecting the over-generalization and setting a limitation on the application of rules.
Once finished the S-BUILDER process, relevant schemata are used by P-BUILDER module.

## 5 P-BUILDER module

The aim of P-BUILDER module is to get syntactic-semantic patterns from the relevant schemata found by S-BUILDER. Patterns are defined as specializations of schemata.

As we saw in section 2, the basic units to extract are <entity attribute value> triples. Patterns can be classified into generic and specific ones, depending on the information they own, and into simple and compound according to the way they have been generated.

Extraction of simple patterns is done by analyzing LSs associated with schemata:

- All schemata with invalid LSs for the domain are dropped out.
- Some patterns are generated combining words, relations and semantics with high frequency in the corpus using previously selected schemata. In the case of the schema:

  (gnom ((n [$R_1$:$S_1$]-$W_1$)(a [$R_2$:$S_2$]-$W_2$)))

  it generates patterns like:

  $p_1$=(gnom ((n [ISA:CARNE]:$W_1$)(a [HYPONYM:COLOR]-$W_2$)))

  where $p_1$ is a pattern for *"carne amarillenta"* ( *"yellowish flesh "*).
- The rest of possible specializations are rejected because, in general, they represent patterns out of the restricted domain or false patterns because of the ambiguity of words. GENERALIZER may produce more than one generalization. But only the most frequent patterns for a schema will be selected. For example *"castaño"* has two senses: the chestnut color and the chestnut tree. The presence of the second sense of *"castaño"* in our domain and in the Spanish WordNet as a hyponym of the concept *habitat* could produce the pattern:

  $p_3$=(gnom ((n [ISA:CARNE]:$W_1$)(a [ISA:CHESNUTTREE]-$W_1$)))

  This false pattern must be rejected.
- Finally, simple patterns used only by finding compound ones are rejected.

Once simple patterns have been extracted, P-BUILDER generates compound patterns taking into account combinations of simple ones. For example, it is frequent to find next simple patterns together in the corpus:

$p_4$=(sp ((r0a *"de"*)(gnom((n [HYPONYM:COLOR]-$W_1$))))
$p_5$=(sp ((r0a *"a"*)(gnom((n [HYPONYM:COLOR]-$W_1$))))

So, they produce the compound pattern $p_4 + p_5$ to represent linguistic expressions of the *color* concept as interval value (e.g. *"de verde a amarillo"*).

## 6 KERNEL module

With KERNEL the system builds extraction rules using two sub-modules: PC (Priority Classifier) and ERG (Extraction Rules Generator) (Fig. 5).

PC module classifies patterns in a hierarchy of priorities according to: 1) their specificity and 2) the length of the pattern. Then ERG module

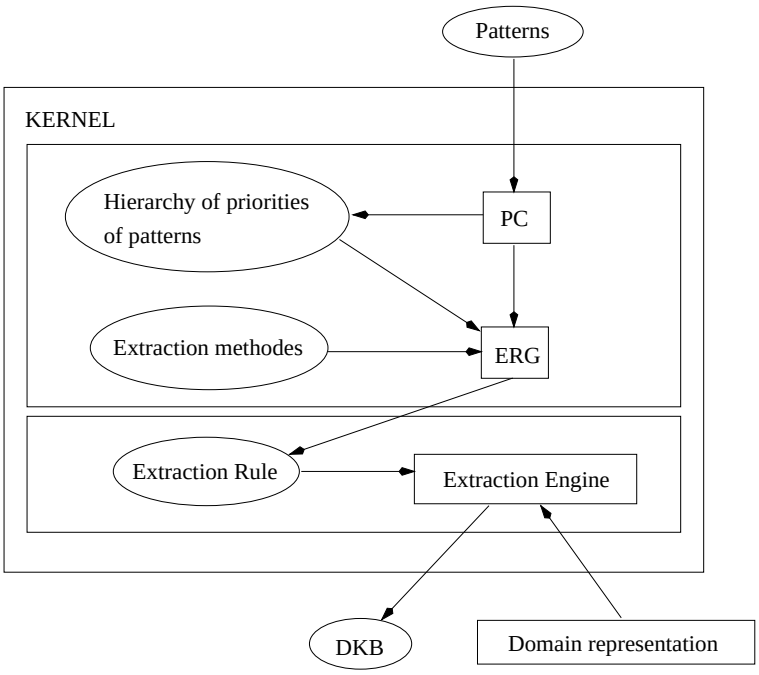


**Fig. 5.** Architecture of KERNEL module

builds extraction rules. The condition of a rule is a pattern and the action is the extraction method needed for that pattern. A set of methods, one for each class of pattern, has been built. Currently KERNEL includes five methods.

The third component of KERNEL, the Extraction Engine, can operate in isolated way in order to carry out the extraction task.

## 7 Results

TURBIO has been tested using 150 mycological cards as training corpus. This represents a total of 21609 words (2991 different lemmas). We have reduced the test to the study of the *color* attribute. Results can be generalized because the color comprises all possible classes of linguistic expressions supported by other attributes. Being Spanish WordNet under construction we have applied the GENERALIZER module only to selected examples. Results reported here have been obtained using only syntactic information.

The S-BUILDER module generated 159 schemata when obtaining $\delta$-sets. 35 of them were refused because they did not contain any valid label, 73 were dropped out due to low frequency in the corpus and 2 in lessening of $\delta$-sets. So, we obtain 49 relevant schemata.

In P-BUILDER module 4 schemata were rejected due to invalid LSs for the domain. From the rest of schemata only 28 were referred to the *color* attribute. With the P-BUILDER methodology we found 53 simple patterns and 68 compound ones, 7 simple patterns of them were dropped out because they only were used to generate compound ones.

In total we found 114 relevant patterns. All of them were used to get extraction rules by the KERNEL module.

The results of the extraction were 48.30% for *recall* and 87.14% for *precision*[10]. Uncoverage is analyzed in the next table:

| Cause | Uncovered cases | Partial Cause |
|---|---|---|
| No extraction rule | 11.49% | Not extracted |
| | 16.26% | Partially extracted |
| Erroneous POS tagging | 12.46% | |
| No coreferent found | 6.22% | Elision without coreferent in the context |
| | 5.27% | Value with reference to other entity |

Results are acceptable but not easy to evaluate because there is no valid benchmark for Spanish (as MUC for English).

## 8 Conclusions and future work

In this paper we have presented a system for extracting information from restricted domain texts using extraction rules learnt from a sample subset of the corpus.

In most existing IE systems the extraction rules are manually provided. Some others use a semi-automatic approach allowing a human expert the selection of the proper level of syntactic-semantic generalization from parse-chunks. Our approach minimizes human intervention using shallow parsing over the whole document and limiting the manual task to validate the automatic semantic generalization.

TURBIO has been tested in mycological domain getting acceptable levels of precision and recall.

Future work includes two main lines:

- The semantic component of TURBIO must be extended for covering a substantial amount of the vocabulary.
- The high error-rate of our tagger is mainly due to the specificity of the sub-language and the high number of unknown words. These problems will be approached tuning the tagged with a specific domain corpus and including a module for dealing with unknown words.
- A bidirectional island-driven shallow parser will be used to deal with the reference problem when necessary.
- We plan to apply TURBIO to English texts in order to compare its performance with widely used benchmarks.

[10] *Recall* is the percentage of possible answers which were correct. *Precision* is the percentage of actual answers given which were correct.

## References


1. P.M. Andersen, P.J. Hayes, A.K. Huettner, I.B. Nirenburg, L.M. Schmandt, S.P. Weinstein. *Automatic extraction of facts from press releases to generate news stories.* In Proceedings of Third Conference on Applied Natural Language Proceeding, pages 170-177. Association for Computational Linguistics, 1992.
2. D. Appelt, J. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Meyers, M. Tyson. *SRI International FASTUS system: MUC-6 test results and analysis.* In Proceedings of 16th MUC, Columbia, MD. 1993.
3. Ed. J. Cowie, W. Lehnert. *Information Extraction. Special Issue.* Communication of ACM. Vol.39(1). 1996.
4. D. Fisher, S. Soderland, J. McCarthy, F. Feng, W. Lehnert. *Description of UMass system as used for MUC-6.* In Proceedings of 16th MUC, Columbia, MD. 1996.
5. R. Gaizauskas, T. Wakao, K. Humphreys, H Cunningham, Y. Wilks. *Description of the LaSIE System as used for MUC-6.* In Proceedings of 16th MUC, Columbia, Maryland, USA. 1995.
6. R. Grishman. *The NYU system for MUC-6 or where's the syntax?.* In proceedings of 16th MUC, Columbia, Maryland, USA. 1995.
7. R. Grishman, B Sundheim. *MUC-6. A brief history.* In Proceedings of 16th Int'l Conf. on Computational Linguistics. 1996.
8. G. Krupka. *Description of the SRA system as used for MUC-6.* In Proceedings of 16th MUC, Columbia, MD. 1995.
9. Ed. M. Pazienza. *Information Extraction. A Multidisciplinary Approach to an Emerging Information Technology.* Springer. 1997.
10. Societat Catalana de Micologia. *Bolets de Catalunya.* ISSN 0212-3460, Ed.: Societat Catalana de Micologia. Facultad de Farmacia de Barcelona. 1994.
11. J. Turmo. *KINOKO: Sistema Experto para la Clasificacion de Micetos basado en Hechos.* Master thesis in Computer Science for the Universitat Politècnica de Catalunya. 1996.
12. R. Weischedel. *Description of the PLUM system as used for MUC-6.* In Proceedings of 16th MUC, Columbia, MD. 1996.


This article was processed using the LaTeX macro package with LLNCS style