Patrixa: A unification-based parser for Basque and its application to the automatic analysis of verbs

Izaskun Aldezabal, Maxux Aranzabe, Atziber Atutxa, Koldo Gojenola, Kepa Sarasola

PATRIXA: a unification-based parser for Basque and its application to the automatic analysis of verbs

Izaskun Aldezabal <jibalroi@si.ehu.es>
Mª Jesús Aranzabe <jibarurm@si.ehu.es>
Aitziber Atutxa <jibatsaa@si.ehu.es>
Koldo Gojenola <jipgogak@si.ehu.es>
Kepa Sarasola <KSarasola@si.ehu.es>

Ixa Taldea. University of the Basque Country
649 Postakutxa. E-20080 Donostia. Basque Country
http://ixa.si.ehu.es

ABSTRACT

In this chapter we describe a computational grammar for Basque, and the first results obtained using it in the process of automatically acquiring subcategorization information about verbs and their associated sentence elements (arguments and adjuncts).

The first part of this chapter (section 1) will be devoted to the description of Basque syntax, and to present the grammar we have developed. The grammar is partial in the sense that it cannot recognize every sentence in real texts, but it is capable of describing the main syntactic elements, such as noun-phrases (NPs), prepositional phrases (PPs), and subordinate and simple sentences. This can be useful for several applications.

Next, the syntactic grammar will be used by a syntactic analyzer (or parser) to automatically acquire information on verbal subcategorization from texts (section 2). The results will later be used by a linguist or processed by statistical filters.

This work has been done by the IXA Natural Language Processing research group, centered on the application of automatic methods to the analysis of Basque. Comparing to other languages (English, German, French, …) Basque can be considered as a minority language due to the following constraints:

- Limited number of language users. This fact implies a reduced number of researchers/developers of computational linguistic tools.

- Limited number of language resources, in the form of computational lexicons, grammars, corpora, annotated treebanks or dictionaries.

These are the main reasons that have compelled the IXA group to the development of automatic methods for the analysis of linguistic data. The work described in this chapter is a part of this effort.

1 THE SYNTACTIC ANALYZER

1.1 A BRIEF INTRODUCTION TO COMPUTATIONAL SYNTAX

The computational treatment of syntax has long been an area of research. From 1950, when the first automatic translation systems were created, many researchers have studied the syntactic relationships among words and the way they are combined to form sentences. However, the task was more difficult than expected. Nowadays, there is no system capable of syntactically analyzing any sentence in real texts, such as newspapers. At the moment, the best syntactic analyzers have been developed for English, but they find an unsolvable obstacle in the form of ambiguity, because many common sentences can produce tens or even hundreds of different syntactic analyses. In this context, we can distinguish two approaches to computational syntax, according to their main objective:

- Full parsing. The aim is to construct more accurate and complete grammars and parsers, with the objective of syntactically analyzing any sentence. As we have noted earlier, the state of the art is still far from this objective.
- Partial parsing. In many systems the objective is not to completely analyze a sentence, but to detect several syntactic elements, such as NPs, verb chains or simple sentences. These pieces of information, also called *chunks* (Abney 1997), are useful for several linguistic applications, as information retrieval or speech synthesis.

Regarding the main kind of knowledge employed, we can classify syntactic analyzers in four groups:

- Unification-based analyzers (Shieber 1986). These systems are based on context-free grammars (Chomsky 1957) with the addition of information to syntactic elements and rules by means of feature structures (see subsection 1.2).
- Finite state analyzers (Karttunen et al. 1997). They are mainly dedicated to partial parsing, that is, they typically distinguish the different components of a sentence. Grammars are defined using regular expressions.

- Constraint grammar (Karlsson 1995). To analyze a sentence, this formalism begins with all the options to analyze each individual word-form, and the task of the grammar is to discard as many options as possible until each word contains a single analysis that gives information about number, case, person and syntactic category. This formalism is called reductionistic because it starts from all the possibilities and it ends only when the correct one is selected.
- Statistical methods. These systems automatically acquire syntactic information (in the form of context-free grammars or regular expressions) from big corpora. The information thus obtained is used to analyze new sentences. Usually, statistical methods are not used in isolation, but combined with other methods (Collins 1997).

The IXA natural language processing group has developed two syntactic analyzers for Basque, one using a unification-based formalism and another one based on a Constraint Grammar. Work on this second formalism is described in (Aduriz et al. 1997; Arriola 2000; Aduriz 2000; Aduriz and Arriola 2001). In this chapter we will describe a unification grammar for Basque together with its application to the task of automatically extracting verbal information from text corpora.

Regarding computational grammars and syntactic analyzers for languages other than Basque we can cite the following:

- Natural Language Software Registry: http://registry.dfki.de
- Computational Linguistics (on-line presentations): http://www.ifi.unizh.ch/CL/InteractiveTools.html#as-h2-3296

Or else, if we want to experiment directly with a syntactic analyzer:

- Syntactic analyzer for English: http://www.conexor.fi
- Syntactic analyzer for Spanish (CliC): http://clic.fil.ub.es/equipo/index_en.shtml

1.2 UNIFICATION-BASED GRAMMAR FORMALISMS AND PATR

Unification-based grammar formalisms are based on context-free grammars (CFG). CFGs were formalized by Chomsky (1957), and they define a grammar as shown in Table 1.

English grammar			*Basque grammar*		
S	→	NP VP	S	→	NP VP
VP	→	Verb NP	VP	→	NP Verb
NP	→	Noun	NP	→	Noun
NP	→	Det Noun	NP	→	Pronoun

Table 1. Two examples of context-free grammars.

Context-free rules are of the form 'a → b' or 'a → b c', where *a* is a non-terminal syntactic category and *b*, *c* are terminals (lexical elements) or non-terminals. Non-terminal symbols (S, NP, PP, ...) are syntactic categories, while terminals are words or morphemes from a lexicon. The chains of terminal symbols that can be derived from the first symbol (or axiom) of the grammar (*S* or sentence in the example) will be the sentences of the language. A sentence belonging to the grammar will be typically described by a tree. For example, Figure 1 shows an analysis tree of a sentence derived using the rules for the Basque grammar in Table 1.

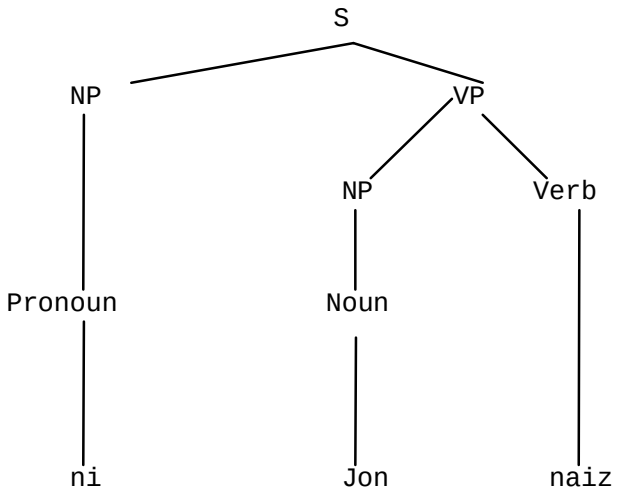

Figure 1. Analysis tree for the sentence '*ni Jon naiz*' (My name is Jon).

The formalism of context-free grammars is simple, but there are problems to describe many linguistic phenomena. For example, if we want to specify the agreement between subject and verb in number and person, then the 'S → NP VP' rule would have to be replaced by a number of similar rules, such as 'S → NP_subj_sing_3 VP_subj_sing_3_abs' or 'S → NP_subj_pl_3 VP_subj_pl_3_abs', and many others.

Unification-based formalisms (Shieber 1986) were defined to overcome this problem. The main idea is to add information to each syntactic element of context-free grammars by means of feature-structures, and to express the syntactic relationships and constraints using equations on that information. Unification is a useful mechanism for the treatment of Basque syntax, due to its rich word-level information and also to the complexity of the syntactic structures that must be dealt with.

This is an example of a rule, given by Shieber (1986):

S → NP VP
S head = VP head
S head subject = NP

The base is a context-free rule that expresses one way of forming a sentence. Two unification equations are used to specify constraints among the sentence components. The first equation states that the head of the sentence is that of the VP, while the second one says that the subject of the sentence corresponds to the NP appearing before the VP. The application of these equations will create a feature structure describing the information in the sentence, as in Figure 2, which corresponds to the sentence "The man runs".

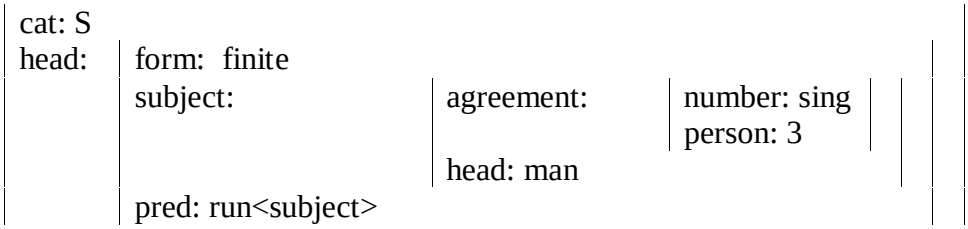

Figure 2. Example of a feature structure.

Several kinds of unification-based formalisms have been defined, such as PATR (Shieber 1986), Generalized Phrase Structure Grammar (GPSG, Gazdar et al. 1985), Lexical Functional Grammar (LFG, Bresnan 1982) and Head-Driven Phrase Structure Grammar (HPSG, Pollard and Sag 1994). When developing a computational grammar, there is always a compromise between depth and breadth of analysis. Sometimes the objective is to develop a formal theory of complex linguistic phenomena. The resulting grammar can serve as a tool for the investigation of linguistic phenomena, but will not be very helpful to analyze real texts, because many linguistically interesting sentences do not appear often in common texts. For example, Abaitua (1988) and Zubizarreta (1992) described several kinds of linguistic phenomena of Basque using the LFG formalism. On the other hand, there is another approach, named shallow parsing (Abney 1997), that is based on the analysis of the most frequently occurring phenomena. This allows, using limited resources, to obtain automatic tools capable of doing several tasks, such as information extraction or machine translation.

We opted for this second option, choosing PATR for the description of Basque syntax, mainly for two reasons:

- To build a computational grammar, we must use the lexical database of Basque (EDBL, Agirre et al. 1995; Aduriz et al. 1998), and this database does not contain all the information required by rich formalisms such as GPSG, LFG or HPSG.
- PATR is a flexible and simple formalism, which can serve in the first attempt to develop a computational syntactic analyzer for Basque. More complex formalisms as LFG and HPSG will be left for future developments.

We will illustrate the main characteristics of the PATR formalism with the grammar in Table 2.

R1. X0 → X1 X2 X0 cat = S X1 cat = NP X2 cat = S X1 case = erg X2 subcat erg agr = X1 agr X0 = X2 X0 subcat erg head = X1	R2. X0 → X1 X2 X0 cat = S X1 cat = S X2 cat = NP X2 case = erg X1 subcat erg agr = X2 agr X0 = X1 X0 subcat erg head = X2
R3. X0 → X1 X2 X0 cat = NP X1 cat = noun X1 type = common X2 cat = case-morpheme X0 head = X1 X0 case = X2 case X0 agr = X2 agr	R4. X0 → X1 X0 cat = S X1 cat = sv X0 subcat = X1 subcat X0 root = X1 root

Table 2. Example PATR grammar of Basque.

The first rule (R1) combines a sentence (S) with an NP, giving an S (in a context-free grammar it would correspond to the rule 'S → NP S'). The X0 component (parent) is formed combining X1 and X2. The unification equations serve two purposes:

- They express syntactic constraints among the sentence elements.
- They also tell how to combine the information from the sentence components (NP and S in the right part of the rule) to form a new element (S at the left of the rule).

The first three equations of rule R1 define the categories of the syntactic elements participating in the rule. The fourth equation ('X1 case = erg') is a constraint imposing that the subject NP must be in the ergative case. The fifth equation ('X2 subcat erg agr = X1 agr') determines whether the NP and the S agree in number, definiteness and person. The sixth equation ('X0 = X2') asserts that the sentence (X0) is a projection of the simpler S appearing in the right hand of the rule, that is, they share the same information. Finally, the last equation

('X0 subcat erg head = X1') of rule R1 states that the NP corresponds to the subcategorized ergative argument.

Rule R2 expresses the same phenomenon as in R1, but changing the order of the sentence components ('S → S NP'). This is how the grammar reflects the free order of Basque. Similar rules must be defined for NPs in absolutive and dative cases, and for subordinate sentences and PPs as well (in our grammar PPs have the same syntactic structure as NPs, differing only in the grammatical case: absolutive, dative and ergative in NPs, and the remaining ones for PPs).

The second line of the table shows rule R3, which defines that an NP can be composed by a noun followed by a case-morpheme ('NP → noun case-morpheme'). This rule links a noun with a morpheme containing information about number, definiteness and case. For example, "*etxe*(house) + *-ari*(to)" (to the house).

Rule R4 defines that, in its simplest form, an S is formed by a synthetic verb (sv). Beginning from this basic S, a sentence is formed linking NPs and PPs to it (either to the right or to the left of the verb).

Table 2 shows an example lexicon. The L1 and L2 entries define verbal forms: *dakarte* ((they) bring (it)) and *dakartza* ((he) brings (them)). For each verb the lexicon defines its category (synthetic verb, abbreviated to *sv*) and information about subcategorization. L1 is defined as a subject-object verb (ergative + absolutive) where the NP in ergative case must be the third person plural (3p) and the absolutive NP must be third person singular (3s). L2 defines that the ergative and absolutive NPs must be respectively third person singular and plural. L3 and L4 describe case-marking morphemes: absolutive-plural (*-ak*) and ergative-plural (*-ek*). The last line of Table 2 defines two noun entries: *gizon* (man) and *txakur* (dog).

L1. X0 entry = ***dakarte*** X0 cat = sv X0 root = ekarri X0 subcat erg agr num = 3p X0 subcat abs agr num = 3s	L2. X0 entry = ***dakartza*** X0 cat = sv X0 root = ekarri X0 subcat erg agr num = 3s X0 subcat abs agr num = 3p
L3. X0 entry = ***-ak*** X0 cat = case-morpheme X0 case = abs X0 agr num = 3p X0 agr def = d	L4. X0 entry = ***-ek*** X0 cat = case-morpheme X0 case = erg X0 agr num = 3p X0 agr def = d
L5. X0 entry = ***gizon*** X0 cat = noun X0 type = common	L6. X0 entry = ***txakur*** X0 cat = noun X0 type = common

Table 3. Example of a lexicon in the PATR formalism.

Taking this lexicon and the grammar in Table 2, the syntactic analyzer can determine that *gizonek dakarte* (the men bring (it)) or *dakartza txakurrak* ((he) brings the dogs) are correct sentences and, conversely, that sentences such as **gizonek dakartza* (*the men brings (them)) are incorrect, because in this case it does not obey the agreement constraint in R1. Figure 3 presents the syntactic tree representing the analysis of the sentence *gizonek dakarte*.

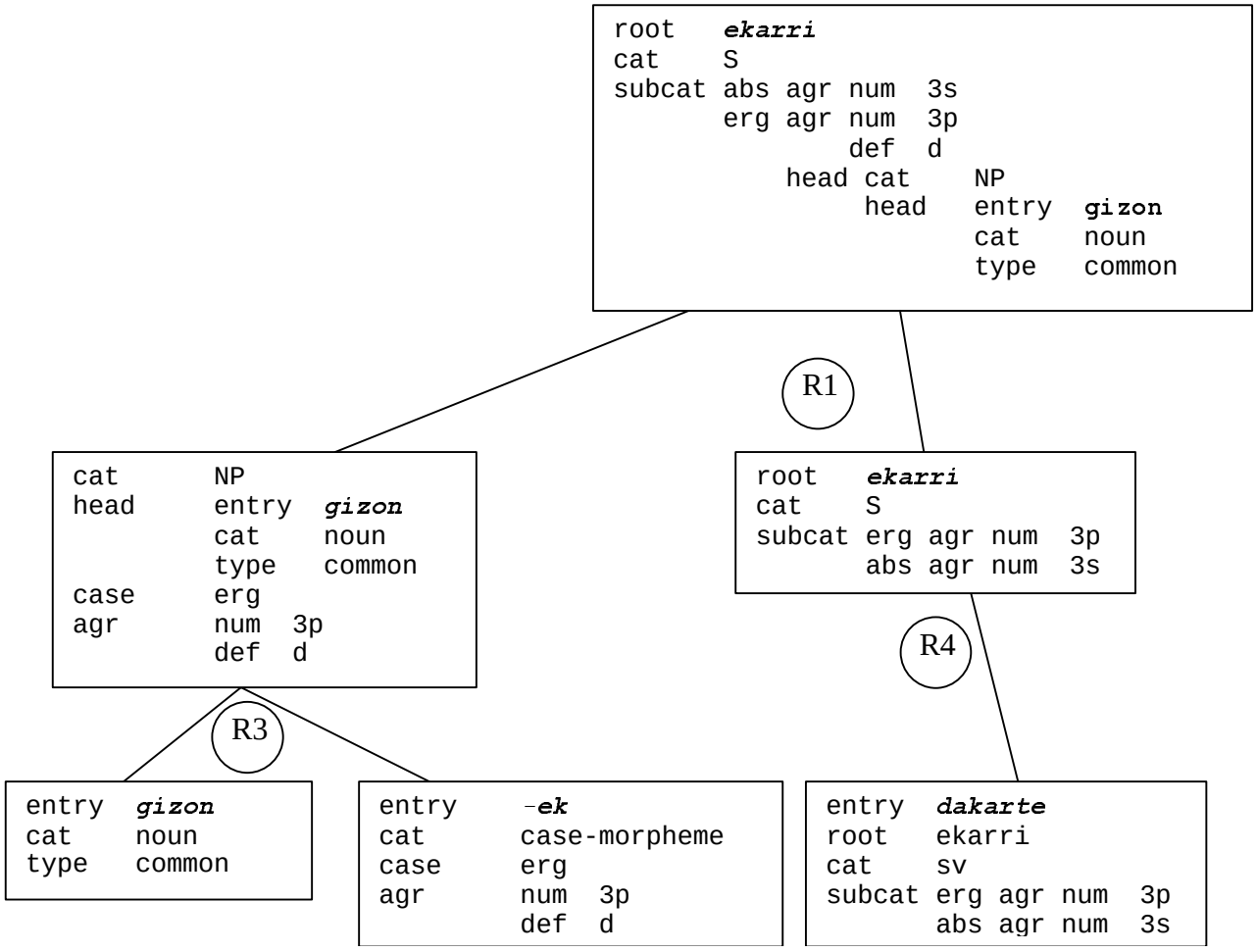

Figure 3. Analysis of *gizonek dakarte* (the men bring (it)).

After explaining the basics of the PATR formalism by an example grammar we will, in the next section, describe the grammar we have developed for Basque.

1.3 A COMPUTATIONAL GRAMMAR FOR BASQUE USING THE PATR FORMALISM

If we want to describe Basque syntax, we must take the following facts into account:

- The morpheme is the basic unit of analysis (Goenaga 1980; Abaitua 1988; Abaitua et al. 1992). This implies that both morphology and syntax will be integrated in the

grammar, without a sharp limit between them, as it happens in agglutinative languages. This will differ from most European languages, such as English or French. For example, in the NP "*gizon* + *handi* + *-a*" (the big man), the case-morpheme "*-a*" at the end is not syntactically linked to the adjective "*handi*" (big) but to the whole noun phrase ("*gizon handi*"). This way the syntactic description is more general and simpler.

- Lexical information is rich. Every lexical entry (and the syntactic elements projected from it) contains information about number, definiteness, case or syntactic functions. The main objective of the grammar will be to adequately combine all this information.
- The lexicon does not contain full subcategorization information. Verbs are the central elements in syntax, both in syntactic theories and in applied systems. From the verbal information, subcategorization is the most complex, specifying how each verb combines with other kinds of elements. In Basque the auxiliary verb conveys information about the subject, object and indirect object (case, number and person), but the lexical database we are using lacks information about main verbs.
- There is agreement between the verb and subject, object and indirect object (corresponding roughly to the ergative, absolutive and dative cases).
- Free order of sentence components. In Basque the order of the main sentence elements (NPs and PPs) is relatively free. This means that in the following example changing the order of subject, object and indirect object gives 24 possible permutations, which are correct sentences in some context:

```
  Txakurrak       egunkaria          ahoan          zekarren.
   The-dog      the-newspaper    in-his-mouth      brought
ergative-3-s   absolutive-3-s   inessive-3-s
   subject         object          modifier          verb
(The dog brought the newspaper in his mouth)
```

  We must also say that this flexibility at sentence level is much more restricted for other syntactic elements (for example, inside NPs or subordinated sentences).

Next, we will begin a description of the grammar, showing the structure of NPs and PPs, and then we will continue with the sentence structure.

We have described three main types of NPs (PPs):

1. NPs and PPs with a common noun as head. NPs and PPs end with a case-morpheme (it contains information about case, number and definiteness). Before the noun there could be optional genitive NPs (similar to PP-of in English) and determiners. After the

noun there could be one or more adjectives and determiners (optional). Unification equations are in charge of checking constraints on order or number:

(NP-gen)	+ **(det)**	+ **noun**	+ **(adj)**	+ **(det)**	+ **case-morpheme**
etxeko		*gauza*	*zahar*	*hori*_____	*ekin*
of-the-house		thing	old	those_____	with (3rd-pl)
(with those old things of the house)					

etxeko	*lau*	*gauza*	*zahar*_____	*etan*
of-the-house	four	thing	old_____	in (3rd-pl)
(in four old things of the house)				

etxeko	*gauza*	*zahar*_____	*ari buruz*
of-the-house	thing	old_____	concerning (3rd-sg)
(regarding the old thing of the house)			

2. NPs (or PPs) with a proper noun as head. There are optional genitive NPs, but neither adjectives nor determiners are accepted:

(NP-gen)	+ **proper-noun**	+ **case-morpheme**
Donostiako	*Jon*_____	*ri*
of-Donostia	Jon_____	to
(to Jon of Donostia)		

3. NPs with a pronoun as head. They only admit the case morpheme:

pronoun	+ **case-morpheme**
*ni*_____	*ri*
I_____	to
(to me)	

These descriptions are relatively simple but not 100% complete, because there are exceptions to some of the principles stated. For example, in NPs formed by a proper noun it could be correct to use adjectives in some contexts, but the inclusion of this fact would have several disadvantages:

- The grammar would be considerably more complicated.
- The resulting ambiguity would increase. It is usual to have tens of analysis for many sentences, due to lexical ambiguity (several analysis per word-form) and syntactic ambiguity (when a part of a sentence can be analyzed by different rules). The inclusion of exceptional cases has the effect of dramatically increasing ambiguity.
- The introduction of new possibilities, although correct in some context, only would account for a very small fraction of sentences in real texts. As our objective is to use the analyzer as a tool for the analysis of written texts, we decided not to include the

special rules in the grammar, as most of them would describe phenomena that do not have even a single instance in the corpora we have studied.

In order to accept the described kinds of syntactic structures, we have defined several auxiliary syntactic categories np1, np2 and np3, starting from the simplest categories to the most complex ones. Finally, adding a case-morpheme to the highest-level structure (np3) forms the category npc (NP + case), that corresponds to an English NP or PP (in fact, they are distinguished by their case: absolutive, ergative and dative for NPs, and the rest of the cases for PPs).

We have taken a broad definition of a case-morpheme. It will describe a suffix containing information about number, case and definiteness. Moreover, we have defined complex suffixes (postpositions) formed by the combination of a suffix with a different word (for example, we take *–ri_buruz* as a suffix, as in *zinemari buruz* (about the cinema)).

The following rules show the structure of NPs and PPs[1]:

	Rule	Examples
1	np1 → noun adj	*etxe EDER* (NICE house)
	| noun	*etxe* (house)
2	np2 → det np1	*ZENBAIT etxe eder* (SEVERAL nice houses)
	| det np1	*HIRU etxe eder* (THREE nice houses)
	| np1 det	*etxe eder BAT* (ONE nice house)
	| proper-noun	*JOHN*
	| np1	*etxe eder* (nice house)
3	np3 → np-gen np2	*MENDI HORRETAKO zenbait etxe eder* (several nice houses OF THAT MOUNTAIN)
	| pronoun	*ZU* (you)
	| np2	*zenbait etxe eder* (several nice houses)
4	npc → np3 case-morpheme	*etxe ederrEKIN* (WITH the nice houses) *mendiko zenbait etxe ederrAK* (several nice houses of the mountain)
		mendiko zenbait etxeRI BURUZ (REGARDING several nice houses of the mountain)
5	np-gen → np3 case-morpheme(gen/gel)	*mendi horretaKO* (OF that mountain)

Table 4. Grammar rules for NPs.

[1] The example rules are a simplification of the actual rules. As we have explained before, each rule will have an associated set of unification equations describing syntactic restrictions among its components.

The structure of the genitive NP (np-gen in rule 5) is the same as for a general NP, where the case must be one of the two genitives (*gen* (possessive) and *gel* (locative)).

In the analysis of a sentence, we do not distinguish the subject from other NPs. A sentence will be a projection of a verb-phrase (VP). The simplest VP is formed by a verb (synthetic or formed by a main verb plus an auxiliary verb). After recognizing the verb, its dependents will be added one by one either to the left or to the right, using the rules in Table 5.

	Rule	Examples
6	vp → synthetic-verb	*dakartza* ((he) brings (them))
7	vp → main-verb aux-verb	*ikusi dute* ((they) have seen (him))
8	vp → npc(erg) vp	*GIZONEK ikusi dute* (THE MEN have seen (it))
	| npc(abs) vp	*GIZONAK ikusi dituzte* ((they) have seen THE MEN)
	| npc(dat) vp	*GIZONARI eman dio* ((he) has given (it) TO THE MAN)
	| vp npc(erg)	*ikusi dute GIZONEK* (THE MEN have seen (it))
	| vp npc(abs)	*ikusi dituzte GIZONAK* ((they) have seen THE MEN)
	| vp npc(dat)	*eman dio GIZONARI* ((he) has given (it) TO THE MAN)
9	vp → npc(*not abs, erg or dat*) vp	*GIZON HORREKIN ikusi dute* ((they) have seen (him) WITH THAT MAN)
10	vp → adb vp	*GAUR egin dut* ((I) have done (it) TODAY)
11	vp → subord-modal-temp vp	*HONA NENTORRELA ikusi dut* ((I) saw (it) WHILE COMING HERE)
	| subord-ind-interrog. vp	*EA JOAN DEN galdetu du* ((he) asked WHETHER HE WAS GONE)
	| subord-completive vp	*ETORRI DIRELA jakin da* ((it) has been known THAT THEY HAVE COME)
12	subord-completive → vp subord-suffix	*Hona nentorrELA* (THAT (I) was coming here)

Table 5. Grammar rules for sentences.

1. Rules 6 and 7 express the simplest way to form an VP, that is, a sentence. It is formed either by a synthetic verb or by the combination of a main verb with an auxiliary verb.
2. Rules for analyzing the grammatical cases (rule 8 in Table 5). NPs in the ergative, absolutive and dative case must agree with the verb in number, case and person. The three rules are duplicated in order to account for free constituent order.
3. Rules for adjuncts (rule 9). These rules account for all the cases (instrumental, inessive, …) apart from the grammatical ones.

As before, there will be a corresponding rule that accepts an adjunct after the verb.

4. Rules for adverbs (see rule 10).
5. Rules for linking subordinated sentences to a verb: completive, indirect interrogative, modal and temporal (see rule 11).
6. Rules for subordinated sentences. They are formed by adding a subordination suffix to a sentence (see rule 12).

The grammar contains a total of 90 rules, each one with an average of 15 equations. As we have explained before, the rules are more complex than the ones presented. Example 1 shows a part of the rule "np3 → np-gen + np2".

```
X0 ---> X1, X2
        X0 cat                      = np3
        X1 cat                      = np-gen
        X2 cat                      = np2
        X0 sint agr                 = X2 sint agr
        X0 lexhead                  = X2 lexhead
        X0 sint elements np-gen     = X1 sint np-gen
        X0 sint elements adj        = X2 sint elements adj
        X0 sint elements determiner = X2 sint elements determiner
        X0 sint head agr            = X2 sint head agr
        ...
```

Example 1. Grammar rule.

As the resulting grammar uses a broad-coverage lexical database, we can say that the analyzer is capable of analyzing any NP (or PP) in real texts, also verifying agreement among the component elements, added to the proper use of determiners. This also happens with sequences of the following syntactic elements not separated by punctuation marks:

- Verbs and verb chains.
- NPs (grammatical cases: ergative, absolutive and dative).
- Adjuncts (NPs in cases other than the three grammatical ones).
- Adverbs.
- Nominalized verbs.
- Relative, completive and modal subordinate clauses.
- Temporal subordinate clauses.
- Indirect interrogatives.
- Simple sentences using all the previous elements. The rich agreement between the verb and the main sentence constituents (subject, object and second object) in case, number and person is verified. As we explained before, sentence analysis is performed up to the level of phenomena that can be described using only syntactic information now included in the lexicon.

1.4 EXAMPLES

Figure 4 shows the analysis of the NP '*gure etxe polit hark*' (that nice house of us). The union of np-gen (of us) and np2 (that nice house) gives an element of category np3, and adding the final case-morpheme (*-ak*) gives the final NP (npc).

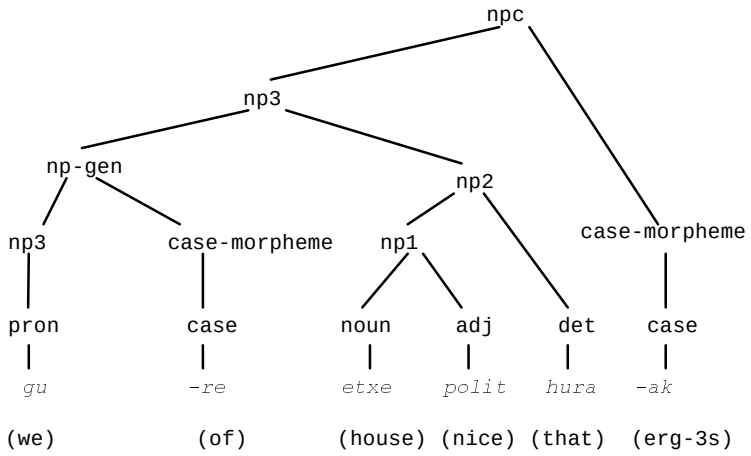

Figure 4. Analysis of *'gure etxe polit hark'* (that nice house of us).

Figure 5 shows the analysis of the sentence '*etxera zetorrela jakin du*'. In this example, a completive subordinated sentence '*etxera zetorrela*' (that he came to the house) is linked to the main sentence ('*jakin du*').

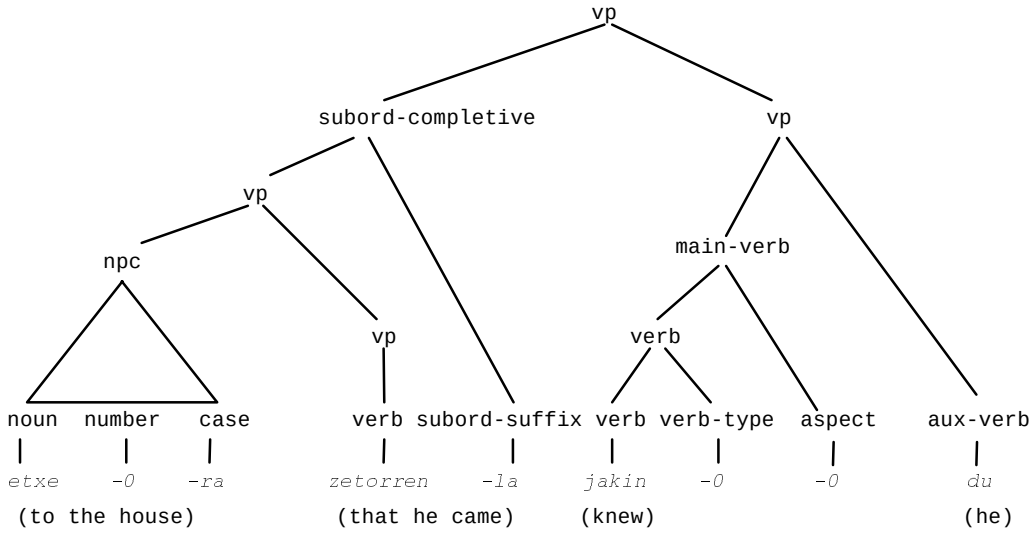

Figure 5. Analysis of *'etxera zetorrela jakin du'* ((he) knew that he came to the house).

1.5 SUMMARY

In the first part of this chapter we have presented the core of PATRIXA, a computational syntactic grammar of Basque. As the lexical coverage is very robust (more than 70,000 lexical entries from the Lexical Database of Basque are used), we can say that the syntactic analyzer provides a good coverage of syntactic elements for the analysis of real texts (newspapers or written texts). The grammar describes extensively NPs, PPs, subordinate sentences and simple sentences.

The grammar can be useful from two perspectives. First, it can be a tool for linguists, helping them in the examination of corpora. The analyzer will give the possibility of finding the syntactic structures present in written texts. Second, it can also be useful for several applications, such as information retrieval or machine translation, where it is crucial the determination of basic syntactic units as the ones found by the analyzer.

In order to obtain deep syntactic analysis of sentences, we think that the next step should be the inclusion of verbal subcategorization information in the grammar. For that reason, we have used the syntactic analyzer to automatically acquire information about verbs and their complements from text corpora. These experiments will be described in section 2.

2 APPLICATION TO THE AUTOMATIC ANALYSIS OF TEXT CORPORA

In this section we will describe the application of the syntactic analyzer to the extraction of information about 1,400 verbs from a newspaper corpus, followed by a preliminary evaluation of the results. These results will be used for both manual and automatic examination (Aldezabal et al. 1998, 2000, 2001).

The acquisition of lexical information is an ineludible step in many applications, ranging from lexicography (construction of dictionaries) to automatic systems, such as machine translation or automatic text understanding. Most of the recent syntactic theories project syntactic structure from the lexicon, where every verbal entry will contain information about predicate subcategorization, including the number and type of arguments, semantic selectional preferences, and so on (Briscoe and Carroll 1997). Manual acquisition of lexical information is reliable and accurate in general, but it is also a costly enterprise, because of the need of highly specialized experts (linguists) in a very time-consuming process. Moreover, manual encoding also faces the problems of errors, such as omission of relevant information or, conversely, adding information based on a linguist’s intuitions which do not match with real occurrences. To that we must also add that predicate subcategorization is associated with

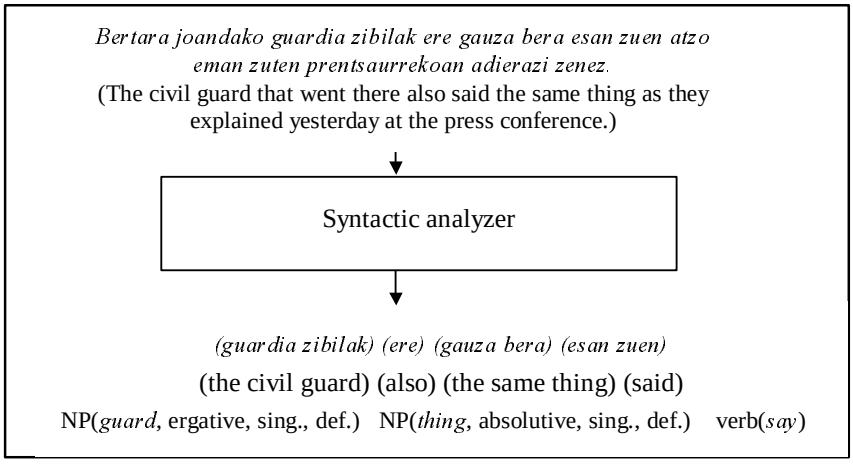

Figure 6. Input sentence and result for the verb *esan* (say).

lexical senses, which vary with the corpus or domain. The huge size of the now available corpora demands successive extensions of the lexicons, to include corpus-specific information or to augment the available lexical information.

For that reason, we have explored the possibility of using computers to help in the process of lexical acquisition. Automatic methods will never get the reliability of a linguist expert, but they can be helpful in several cases:

- The information gathered automatically can be validated by experts. This way, the linguist gets rid of the most mechanical task of examining hundreds of text sentences.
- In cases where it is not feasible to dedicate people to the task of lexical acquisition, automatically collected information could serve as an approximation useful for several applications. The reliability of the approximation can be evaluated examining a small fraction of the extracted information.

In our experiment, we have automatically examined more than 1,000,000 words of newspaper text obtaining, for each of 1,400 verbs, the set of sentences containing each verb and the elements associated with it (arguments and/or adjuncts), marking each element with information about case, number or type of subordinated sentence. Figure 6 shows the result obtained by the system when examining the verb *esan* (say). The syntactic analyzer first tries to analyze the whole sentence. As the grammar is partial and the sentences long, many times the analyzer does not find an analysis for all the sentence, but it can obtain the main syntactic components. In a second phase of the process, the analyzer looks up the syntactic elements surrounding the target verb (*esan*) and determines which of them are the most plausible arguments or adjuncts. This way, the result is the last line in Figure 6, where the verb is linked

with two NPs (ergative and absolutive). This kind of information can be useful for an ulterior manual or automatic determination of subcategorization frames.

Subsection 2.1 will review previous works on the automatic acquisition of subcategorization information. Next, we will describe the architecture of the system (subsection 2.2), together with the linguistically relevant aspects of the experiment. In subsection 2.3 we will examine the results.

2.1 PREVIOUS WORK ON THE ACQUISITION OF SUBCATEGORIZATION INFORMATION

Concerning the acquisition of verb subcategorization information, there are proposals ranging from manual examination of corpora (Grishman et al. 1994) to fully automatic approaches. (Briscoe and Carroll 1997; Carroll et al. 1998) describe a grammar based experiment for the extraction of subcategorization frames with their associated relative frequencies, obtaining 76.6% precision and 43.4% recall.

(Kuhn et al. 1998) compare two approaches for the acquisition of subcategorization information: a corpus query pattern based approach (no grammar, using regular expressions on morphologically analyzed word forms) and a grammar based approach (in a way similar to (Briscoe and Carroll 1997)). Both are applied to the problem of acquiring subcategorization instances of 3 subcategorization frames, showing that the grammar based approach improves results specially in recall, due mainly to the higher-level knowledge encoded in the grammar. Comparing with our work, we think that our system is situated between the two approaches, as we will use a partial parser. Our objective is more ambitious in the sense that we try to find all the subcategorization instances, rather than distinguishing among 3 previously selected frames.

On the statistical side, (Carroll and Rooth 1998) present a learning technique for subcategorization frames based on a probabilistic lexicalized grammar and the Expectation Maximization algorithm using unmarked corpora. The results are promising, although the method is still computationally expensive and requires big corpora (50 million words).

2.2 DESCRIPTION OF THE PROCESS

We have developed a parsing system divided in several main modules: the unification-based parser that we have presented in section 1 is the core of the system (see Figure 7). Prior to parsing, there is another step concerned with morphological analysis and disambiguation,

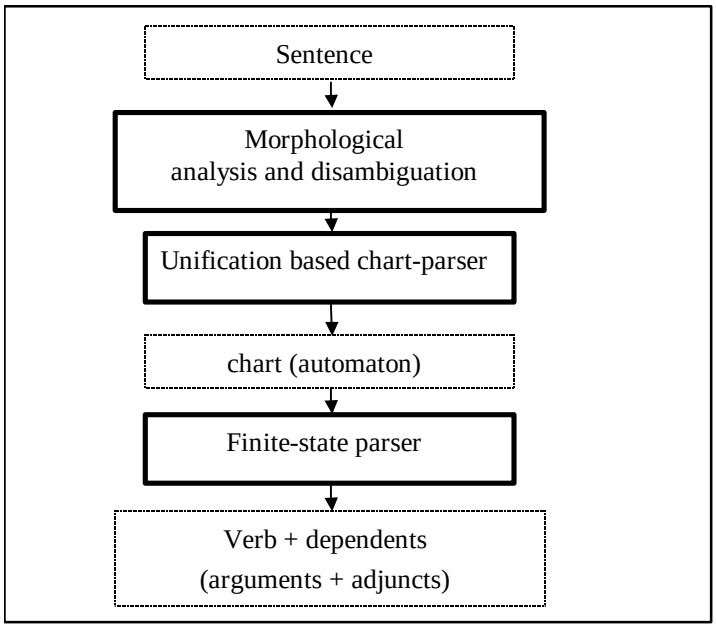

Figure 7. Description of the system.

using the basic tools for Basque that have been developed in previous projects. These are the main modules of our system:

- The lexical database. As we have commented earlier, it is a large repository of lexical information, with about 70.000 entries (including lemmas and declension/derivational morphemes), each one with its associated linguistic features, like category, subcategory, case and number, contained in a commercial database management system.
- Morphological analysis and segmentation. Inflectional morphology of Basque was completely described in (Alegria et al. 1996). This system applies Two-Level Morphology (Koskenniemi 1983) for the morphological description and obtains for each word its segmentation(s) into component morphemes, where each morpheme is associated with its corresponding features in the lexicon. The segmentation module has full coverage of free-running texts in Basque, and it is capable of treating unknown words and non-standard forms, such as dialectal variants and typical errors (Aduriz et al. 2003).
- Morphological disambiguation. A disambiguation system was implemented for the assignment of the correct lemma and part-of-speech to each token in a corpus (Ezeiza et al. 1998) taking the context into account, by means of statistical (Hidden Markov Models) and hand-crafted rules in the Constraint Grammar formalism (Samuelsson and Voutilainen 1997, Karlsson et al. 1995, Aduriz et al. 1997). This tool reduces the

high word-level ambiguity from 2.65 to 1.19 interpretations, still leaving a number of interpretations per word.

- Unification-based chart parsing. The syntactic analyzer presented in section 1 recognizes the main syntactic units of the sentence, described in the unification-based PATR grammar of Basque.
- After the partial parser has obtained the main syntactic components of the sentence, there are multiple readings for each sentence, as a result of both morphological ambiguity (1.19 interpretations per word-form after morphological disambiguation) and syntactic ambiguities introduced by the partial parser. For this reason, we have also developed a finite-state grammar that performs syntactic disambiguation and filtering of the results. This grammar consists of a set of regular expressions and transducers for both disambiguation and determination of clause boundaries, in order to exactly delimit the syntactic elements corresponding to each verb (Aldezabal et al. 2003). The finite state filter has been implemented using the *Xerox Finite State Tool* (XFST, Karttunen et al. 1997).

2.2.1 Size and type of the corpus

In the present work we have used newspaper texts from "Euskaldunon Egunkaria", ranging from January 1999 to May 2000. This corpus offers a rich variety of text types, using standard Basque. It contains 111,000 sentences (more than one million words). In a preliminary stage of this work we also used the *Statistical corpus of 20th Century Basque* (UZEI 2003).

2.2.2 Number of verbs

We selected a preliminary set of 1,400 verbs appearing in the corpus. From them, 400 had more than 50 occurrences in the corpus, which we have taken as the minimum for the results to be representative.

2.2.3 Data extraction method

After doing some preliminary tests and a manual verification of the results, we defined several procedures to be applied, related with specific features of Basque, with the aim of improving the reliability of the results. The resulting procedures are the following:

1. Grouping of cases and subordination suffixes. Basque has a high number of cases and subordination suffixes. In our grammar we have described 61 different types. Concerning the verb, however, several of them perform a similar function. We will not go into details about what we have defined as a "similar function". The grouping was

made based mainly on the syntactic function (subject, object, …), also taking into account semantic relationships. So, for example, we have grouped subordination suffixes related to time: *-nean* (when), *-t(z)ean* (when), *-rako* (for when), *-terakoan* (while), *-takoan* (after), *-ino* (until), *-netik* (since), *-neko* (of when). We must also say that the grouping could be done in a different way depending on the definition of "similarity". After the grouping, we had 28 groups of elements.

2. Using the auxiliary verb. The auxiliary verb in Basque gives information about the "grammatical cases" (absolutive, ergative and dative). So, even when a sentence does not contain an NP corresponding to one of these cases, the auxiliary verb reflects their occurrence and, therefore, we can assume that the elements exist. This feature is characteristic of *pro-drop* languages. Nevertheless, in unergative verbs the object NP (marked with the absolutive case) does not exist, even when the auxiliary verb marks it. Taking these verbs into account, we have decided not to recover NPs in the absolutive case, because doing it the system would get incorrect information about all of the unergative verbs.

   Summarizing, the recovering of cases has been applied in the following syntactic environments:

   - If the auxiliary is of the type absolutive-ergative (this type of verb is usually represented by the form corresponding to the present indicative in third person singular: DU), the NP in the ergative case will be recovered. This assumption will be wrong for all the verbs associated to weather (to rain, to snow, …), because they will never have a subject in the ergative case. However, as these verbs form a reduced set that could be treated separately, we estimated that the application of this heuristic will be useful.
   - If the auxiliary verb is of the type absolutive-ergative-dative (DIO), the ergative and the dative NPs will be recovered.
   - If the auxiliary verb is of the type absolutive-dative (ZAIO), the dative NP will be recovered.

3. Elimination of ill-formed syntactic combinations. Several combinations of cases with the auxiliary verb can never appear in a sentence and, consequently, we eliminated them, because they will always correspond to an error of the syntactic extraction system. Most of the times the errors appear because the main sentence and the subordinated ones are incorrectly delimited:

- An ergative NP can never appear with an auxiliary verb of the absolutive (DA) or absolutive-dative type (ZAIO).
- A verb cannot contain two ergative NPs.
- Syntactic structures with more than five elements (arguments or adjuncts) are not common, and most of the times are a result of errors of our analyzer. For that reason, we did not take them into account.

	Input sentence	Output
1	*Bideoa bezalako euskarri berrien abantailak azpimarratu zituen Villotak, eta* ***ildo horretan dokumentala bideo-sorkuntzara hurbildu dela*** *deritzo.* Villota stressed the advantages of new media such as video, and **in a similar way** he thinks that **documental has neared towards video creation.**	**verb**: *hurbildu* **auxiliary**: *dela* (DA) **absolutive**: *dokumental* **head**: documental (sing.) **inessive(in)**: *ildo horretan* **head**: way (sing.) **adlative(to)**: *sorkuntzara* **head**: creation (sing.)
2	*Unionista amorratuenek eta, gezurtia deitu zioten Trimbleri ,* ***UUPko burua sarrerara hurbildu zenean*** *.* And the most stubborn unionists called Trimble liar, when **the head of UUP neared the entry.**	**verb**: *hurbildu* **auxiliary**: zenean (DA) **absolutive**: *UUPko burua* **head**: head (sing.) **adlative(to)**: *sarrerara* **head**: entry (sing.)
3	*Baina* ***jendea frontoira hurbiltzen ari da*** *, erantzuten ari da .* But **people is nearing the fronton,** they are responding.	**verb**: *hurbildu* **auxiliary**: *da* (DA) **absolutive**: *jendea* **head**: people **adlative(to)**: *frontoira* **head**: fronton (sing.)
4	*Garaipena eskuan,* ***Pascual Jover minutu batzuk beranduago hurbildu zen Vital kutxaren aretora*** *.* With the victory in his hands, **Pascual Jover neared the Vital hall several minutes later.**	**verb**: *hurbildu* **auxiliary**: zen (DA) **absolutive**: *minutu batzuk* **head**: minute (pl.)
5	Manifestazioa Hernani kaletik zihoala , **pertsona bat ondoko kale batetik hurbildu zen presoen aldeko oihalarekin** eta eskuak goraturik , bake seinalean . When the demonstration crossed Hernani street, **one person neared with a sheet in favour of prisoners from a street nearby** and his hands up, in sign of peace.	**verb**: *hurbildu* **auxiliary**: *zen* (DA) **absolutive**: *pertsona bat* **head**: person (sing.) **ablative(from)**: *ondoko kale batetik* **head**: street (sing.)

Table 6. Examples of input sentences and their corresponding output.

2.3 RESULTS

Table 6 presents an example of the results obtained by the system when applied to the verb *hurbildu* (to near). The second column contains the input sentence, where the subsentence corresponding to the target verb has been marked in bold type. The third column presents the result obtained by our system. For each instance of the target verb the system gets its auxiliary verb and, for each dependent, its case, head and number. For example, in sentence 1 the system finds NPs in the absolutive, inessive and adlative cases. The result will be the set of candidate dependents, where some of them will be arguments and the rest will correspond to adjuncts. For example, in sentence 1 the inessive NP *ildo horretan* (in the same way) is an adjunct, while the other NPs correspond to arguments.

Sentence 4 is an example where the system gets an incorrect result, because the syntactic analyzer does not recognize the temporal modifier *minutu batzuk beranduago* (several minutes later) as a single unit, due to a gap in the partial grammar. As a result, it incorrectly proposes *minutu batzuk* (several minutes, absolutive) as the subject of the target verb.

Finally, sentence 5 shows how sometimes the system does not obtain the complete list of dependents of a verb. In this example, the analyzer correctly identifies two dependents, but misses a third one: *presoen aldeko ohialarekin* (with a sheet in support of prisoners). This is due to unresolved ambiguity of the auxiliary verb *zen*, which can be both sentence final and a verb in the past tense. In this example, the correct reading corresponds to the past tense, which would imply that this element is a dependent. However, as the morphosyntactic disambiguation process is not able to decide about which one is the correct interpretation, the system, in case of doubt, does not take any risk, and discards the element, taking into account the sentence final interpretation. This strategy tries to maximize precision (that is, to minimize the number of incorrect dependents) at the cost of lowering recall (some correct elements will also be discarded).

In order to estimate the results obtained by our system, we tested three different approximations:

1. General frequency of dependents. With the aim of obtaining a general view of the corpus, we measured the relative frequency of each type of dependent, including all the cases for NPs (PPs) and each type of subordinated sentence. Figure 8 shows the ten kinds of dependents appearing most in the corpus (those that appear in more than 1% of the sentences). Table 7 shows the correspondence of the abbreviations in the table with their associated syntactic element.

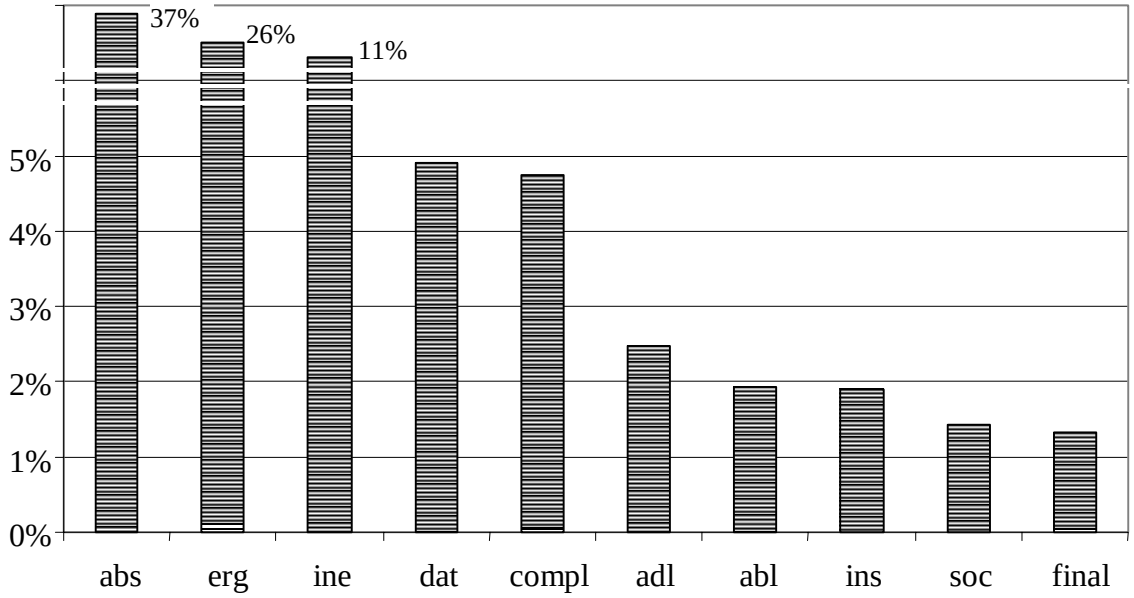

Figure 8. Most frequent cases and subordination suffixes appearing in the corpus.

Case	Abbreviation	Example
absolutive	abs	THE HOUSE (object)
ergative	erg	THE MAN (subject)
inessive	ine	IN THE HOUSE
dative	dat	TO THE MAN
completive subordinated sentence	compl	(I know) THAT SHE WOULD COME
adlative	adl	TO THE HOUSE
ablative	abl	FROM THE HOUSE
instrumental	ins	WITH THE HAMMER
sociative	soc	WITH THE MAN
final subordinated sentence	final	(I did it) FOR YOU TO COME

Table 7. Different types of dependents.

Figure 8 shows that three types of dependents appear most frequently: NPs in the absolutive, ergative and inessive case. The high frequency of the absolutive case can be considered normal, as this is the case used to represent the subject of intransitive verbs as well as the object of transitive ones, that is, this case will appear with most of the verbs. Similarly, the ergative case is used as the subject of transitive and unergative verbs. The high frequency of the inessive case can be explained if we take into account that the corpus is formed by newspaper texts, which must be situated both in time and location.

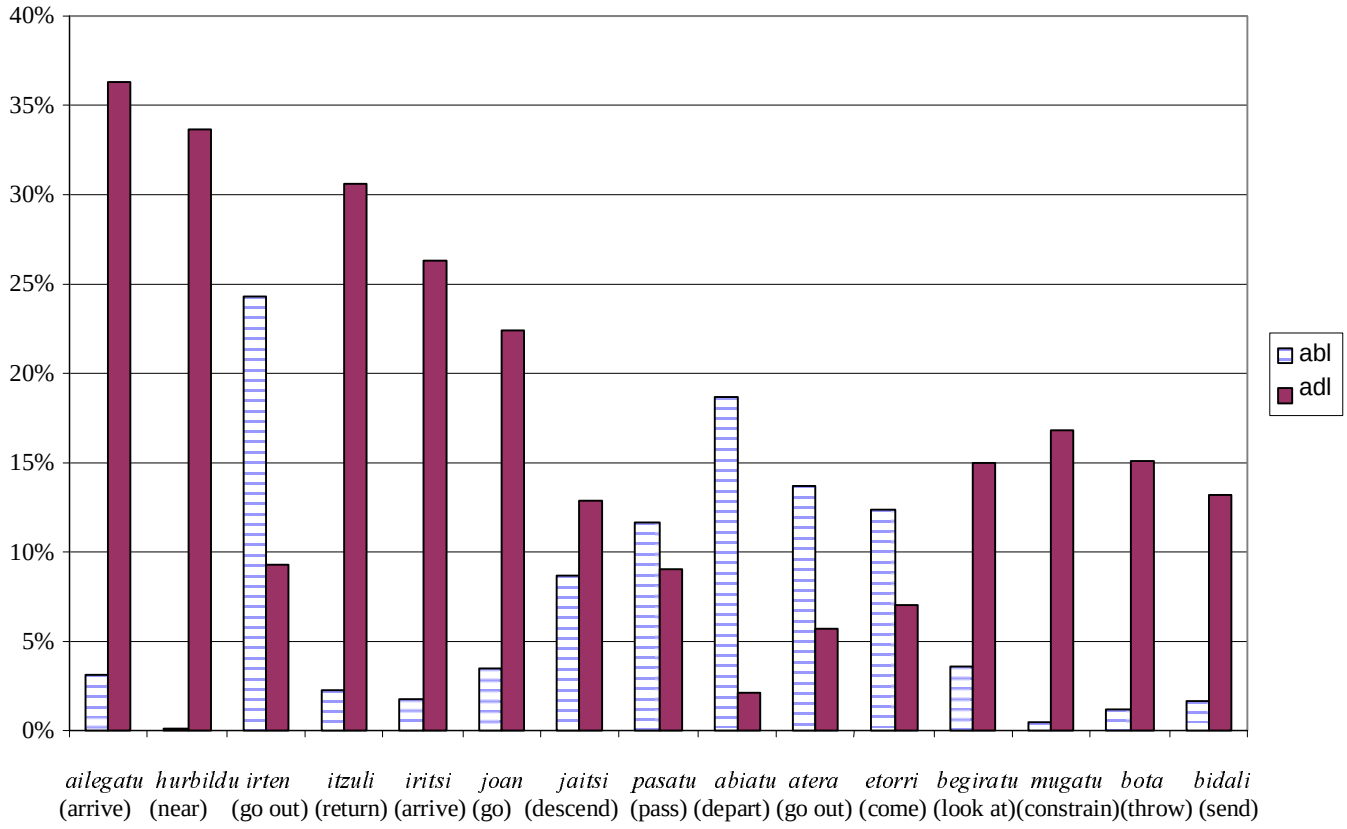

Figure 9. Verbs with high frequency of ablative and adlative cases.

If we look at the next most frequent types of dependents, we find the dative case, typically representative of goal, and completive sentences. These can also be derived from the type of corpus, because many communicative verbs are used, containing a message that has to be transmitted (and sometimes has an associated goal). This is the case with verbs expressing volition, desire or preference.

Next to these elements we find the locative cases: ablative, adlative and instrumental (by, by means of), followed by the sociative and the subordination suffix *-t(z)eko*, which can be both final and completive.

2. In a second approximation we wanted to investigate the validity of the results regarding the ability of the system to detect certain types of verbs from their associated dependents. In our experiment we tried to select verbs corresponding to motion taking those verbs with the highest frequencies of the ablative (from) and adlative (to) cases. Figure 9 shows the 15 verbs with a highest frequency of these two cases in the corpus.

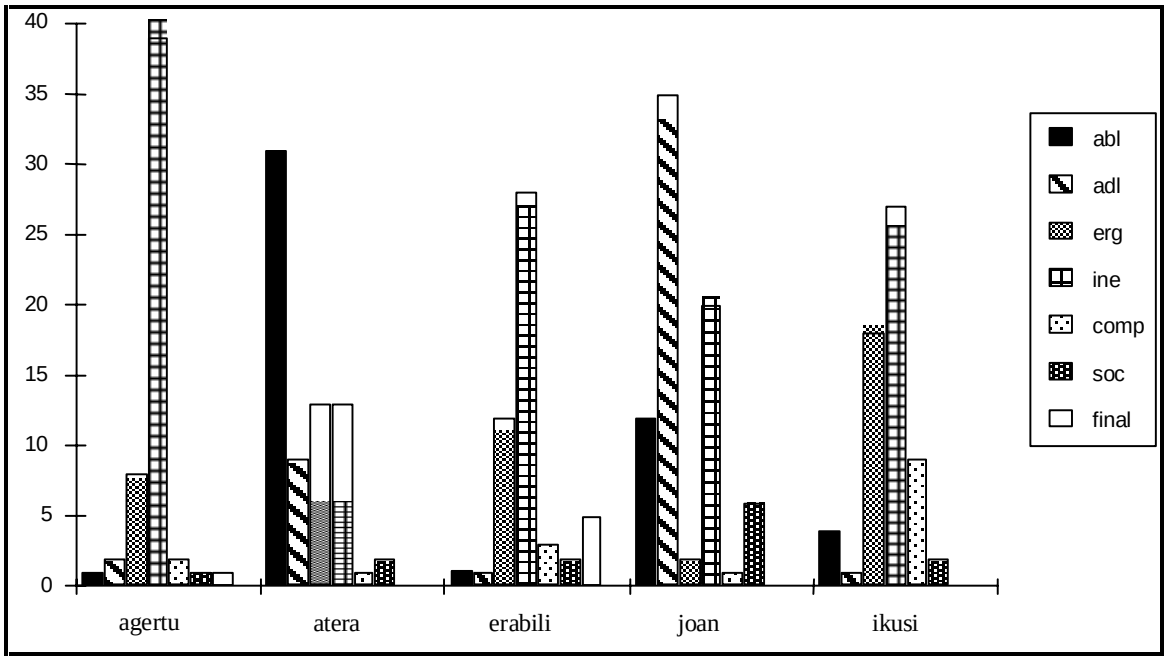

Figure 10. Frequency of elements appearing with each of five verbs.

The results show the usefulness of the system to find verbs with similar characteristics. From the 15 verbs with highest proportion of the cases ablative and adlative, 13 correspond to typical motion verbs. The two exceptions are *mugatu* (to constrain) and *begiratu* (to look at).

Even when all the verbs admit both cases, many times the verb shows preference for one of them. For example, the verb *hurbildu* (to near), in one extreme, is rarely accompanied by the ablative case. This asymmetry could be explained defining two subclasses of motion verbs:

- Verbs expressing source or beginning by means of the ablative case (from). This set would contain the following verbs, sorted by descending order of frequency: *irten* (to go out), *abiatu* (to depart), *atera* (to go out), *etorri* (to come) and *pasatu* (to pass).
- Verbs expressing destiny, which express a goal or arrival by means of the adlative case: *ailegatu* (to arrive), *hurbildu* (to near), *itzuli* (to return), *iritsi* (to arrive) and *joan* (to go).

3. Finally, we studied the frequencies of dependents for five common verbs: *agertu* (to appear), *atera* (to go out), *erabili* (to use), *joan* (to go) and *ikusi* (to see).

   Figure 10 shows the frequencies of elements appearing with each verb. The absolutive case has been omitted, because it is the most frequent one in all the verbs, due to the reasons explained before. The inessive is predominant, as it situates the sentences in temporal and spatial coordinates. The ergative case gives the subject of actions. After

these elements, we can see how each verb shows preference for different kinds of subcategorized elements. For example, the verb *erabili* (to use) contains a high proportion of subordinated sentences with the *–t(z)eko* suffix, expressing finality.

These results show that the tool is useful for the automatic selection of possible subcategorized elements. The information obtained can then be used by a linguist or processed by statistical methods to select subcategorization frames for verbs.

3 CONCLUSION

In this work we have presented PATRIXA, a syntactic analyzer for Basque based on a unification-based formalism (PATR), and its application to the automatic analysis of texts, in order to extract information on verbal subcategorization.

These are the main features of the syntactic analyzer:

- Lexical coverage. As the system is based on a wide-coverage lexical database of Basque (EDBL) with more than 70,000 entries, the system is very robust, capable of analyzing almost any word occurring in texts.
- Grammatical coverage. The system correctly analyzes NPs, PPs, simple sentences and subordinated sentences. However, the grammar does not address several linguistic phenomena such as coordination or complex sentences.
- Ambiguity. Many times the syntactic analyzer obtains more than one analysis for a piece of text. For example, *gizonak* can be both “the man”(subject) and “the men”(object). This has been dealt with by means of special disambiguation rules (Aldezabal et al. 2003).

In the second part of the work (section 2), we have presented the application of the grammar to the automatic analysis of texts, with the objective of obtaining information on verbal subcategorization. These are the main characteristics of the experiment:

- The corpus contains more than a million words of newspaper texts, with the objective of obtaining information about 1,400 verbs.
- The system obtained, for each verb and sentence, a list of its corresponding dependents (arguments and adjuncts). For evaluation we measured precision (the number of correctly selected elements / all the elements returned by the parser) and recall (the number of correctly selected elements / all the elements present in the sentence). The results are reliable, with 87% precision (this corresponds to the proportion of correctly selected dependents) and 66% recall (that is, the system

obtained an analysis for 66% of the sentences). Although there is always a balance between recall and precision, we tried to maximize the latter, sometimes at the cost of lowering recall.

The following are the lines of work to continue in the future:

- Extension of the grammar. We plan to extend the grammar in two ways. First, including syntactic constructions not treated at the moment, such as coordination or complex sentences. Second, including subcategorization information, not present at the moment in the lexical database.
- Regarding the results of the analyzer, the information gathered will be used to manually and automatically extract subcategorization information about verbs.
- We also plan to compare the results with other works on extraction of subcategorization information. For example, (Arriola 2000) has studied the extraction of this kind of information from a dictionary (Sarasola 1997).

ACKNOWLEDGEMENTS

This research has been supported by the European Commission (MEANING IST-2001-34460), the Spanish Ministry of Science and Technology (Hermes TIC2000-0335-C03-03), the University of the Basque Country (9/UPV00141.226-14601/2002) and the Basque Government (ETORTEK2002/HIZKING21).

BIBLIOGRAPHY

Abaitua J., Aduriz I., Agirre E., Alegria I., Arregi X., Artola X., Arriola J.M., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Maritxalar M., Sarasola K., Urkia M., Zubizarreta J.R. 1992. *Estudio comparativo de diferentes formalismos sintacticos para su aplicacion al euskara*. Technical report, UPV/EHU/LSI.

Abaitua, J. 1988. *Complex predicates in Basque: from lexical forms to functional structures*. PhD, University of Manchester.

Abney S. P. 1997. Part-of-Speech Tagging and Partial Parsing. S. Young eta G. Bloothooft, editors, *Corpus-Based Methods in Language and Speech Processing*, Kluwer, Dordrecht.

Aduriz I., Arriola J.M., Artola X., Díaz de Ilarraza A., Gojenola K., Maritxalar M. 1997. *Morphosyntactic disambiguation for Basque based on the Constraint Grammar Formalism*. Conference on Recent Advances in Natural Language Processing, RANLP'97, Bulgaria.

Aduriz I., Aldezabal I., Ansa O., Artola X., Díaz de Ilarraza A., Insausti J. M. 1998. *EDBL: a Multi-Purposed Lexical Support for the Treatment of Basque*. Proceedings of the First International Conference on Language Resources and Evaluation, LREC'98, Granada.

Aduriz I. 2000. *Morfologiatik Sintaxira Murriztapen Gramatika baliatuz*. PhD Thesis, Department of Basque Philology, University of the Basque Country.

Aduriz I., Arriola J. M. 2001. *Euskararen murriztapen gramatika. Desanbiguazio morfologikoaren tratamendua, azterketa sintaktikoaren lehen urratsak eta aplikazioak*. P. Lafitteren sortzearen mendemuga, Euskaltzaindia (Gramatika batzordea), Baiona.

Aduriz I., Aldezabal I., Alegria I., Arriola J., Díaz de Ilarraza A., Ezeiza N., Gojenola K. 2003 *Finite State Applications for Basque*. EACL'2003 Workshop on Finite-State Methods in Natural Language Processing, Hungary.

Agirre E., Arregi X., Arriola J.M., Artola X., Díaz De Ilarraza A., Insausti J.M., Sarasola K. 1995. *Different issues in the design of a general-purpose Lexical Database for Basque*. First Workshop on Applications of Natural Language to Databases, France.

Aldezabal I., Goenaga P., Gojenola K., Sarasola K. 1998. *Subcategorización verbal vasca: propuesta inicial y herramienta de validación*. Proceedings of SEPLN'98, Alicante.

Aldezabal I., Gojenola K., Sarasola K. 2000. *A Bootstrapping Approach to Parser Developm*ent. Proceedings of the International Workshop on Parsing Technologies, IWPT'2000, Trento.

Aldezabal I., Aranzabe M., Atutxa A., Gojenola K., Sarasola K., Goenaga P. 2001. *Extracción masiva de información sobre subcategorización verbal vasca a partir de corpus*. Proceedings of SEPLN'2001, Jaén.

Aldezabal I., Atutxa A., Aranzabe M., Gojenola K., Oronoz M., Sarasola K. 2003. *Application of finite-state transducers to the acquisition of verb subcategorization information*. To appear in the Special Issue on Finite State Methods in NLP, Journal of Natural Language Engineering, Cambridge University Press.

Alegria I., Artola X., Sarasola K., Urkia M. 1996. *Automatic morphological analysis of Basque*. Literary & Linguistic Computing, Vol. 11, Nº 4, 193-203. Oxford University Press. Oxford.

Arriola J.M. 2000. *Hauta-Lanerako Euskal Hiztegi-ko informazio lexikalaren erauzketa erdi-automatikoa eta bere integrazioa sistema konputazional batean*. PhD Thesis, Department of Basque Philology, University of the Basque Country.

Bresnan J. 1982. *The Mental Representaion of Grammatical Relations*. The MIT Press, Cambridge, Massachusetts.

Briscoe T., Carroll J. 1997. *Automatic Extraction of Subcategorization from Corpora*. Proceedings of the Conference on Applied Natural Language Processing, ANLP-97, EEUU.

Carroll G., Rooth M. 1998. *Valence Induction with a Head-Lexicalized PCFG*. Proceedings of the Conference on Empirical Methods in Natural Language Processing, Granada.

Carroll J., Minen G., Briscoe T. 1998. *Can Subcategorisation Probabilities Help a Statistical Parser?* Proceedings of the 6th ACL/SIGDAT Workshop on Very Large Corpora, Montreal.

Chomsky, N. 1957. *Syntactic structures*. The Hague: Mouton.

Collins M. 1997. *Three New Probabilistic Models for Statistical Parsing*. Proceedings of the Conference of the Association for Computational Linguistics, ACL'97, Madrid.

Ezeiza N., Alegria I., Arriola J.M., Urizar R., Aduriz I. 1998. *Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages*. COLING-ACL'98, Montreal.

Gazdar G., Klein E., Pullum G., Sag I. 1985. *Generalized Phrase Structure Grammar*. Harvard University Press.

Goenaga P. 1980. *Gramatika bideetan*. Erein

Grishman R., Macleod C., Meyers A. 1994. *Comlex syntax: building a computational lexicon*. Proceedings of the Conference on Computational Linguistics, COLING-94, Japan.

Karlsson F., Voutilainen A., Heikkila J., Anttila A. 1995. *Constraint Grammar: A Language-independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Karttunen L., Chanod J-P., Grefenstette G., Schiller A. 1997. *Regular Expressions For Language Engineering*. Journal of Natural Language Engineering, Cambridge University Press

Koskenniemi K. 1983. *Two-level Morphology: A general Computational Model for Word-Form Recognition and Production*. PhD thesis, University of Helsinki.

Kuhn J., Eckle-Kohler J., Rohrer. C. 1998. *Lexicon Acquisition with and for Symbolic NLP-Systems -- a Bootstrapping Approach*. Proceedings of the International Conference on Language Resources and Evaluation, LREC'98, Granada.

Sammuelson C., Voutilainen A. 1997. *Comparing a Linguistic and a Stochastic Tagger*. Proceedings of ACL-EACL'97, Madrid.

Sarasola I. 1997 *Euskal hiztegia*. Kutxa Fundazioa. Donostia.

Pollard C., Sag I. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press.

Shieber S.M. 1986. *An Introduction to Unification-Based Approaches to Grammar*. CSLI Lecture Notes, 4 zenbakia, Stanford.

UZEI Institute and the Academy of the Basque Language. 2003. *Statistical corpus of 20th Century Euskera.* http://www.euskaracorpusa.net/XXmendea/index.html

Zubizarreta J.R. 1992. Un modelo funcional de diálogo para diálogos orientados por la tarea. PhD Thesis, Department of Computer Languages and Systems, University of the Basque Country.