A BOOTSTRAPPING APPROACH TO PARSER DEVELOPMENT

Izaskun Aldezabal, Koldo Gojenola, Kepa Sarasola

Department of Computer Languages and Systems
Informatika Fakultatea, 649 P. K., Euskal Herriko Unibertsitatea,
20080 Donostia (Euskal Herria)

{jibalroi, jipgogak, jipsagak}@si.ehu.es

Abstract

This paper presents a robust parsing system for unrestricted Basque texts. It analyzes a sentence in two stages: a unification-based parser builds basic syntactic units such as NPs, PPs, and sentential complements, while a finite-state parser performs syntactic disambiguation and filtering of the results. The system has been applied to the acquisition of verbal subcategorization information, obtaining 66% recall and 87% precision in the determination of verb subcategorization instances. This information will be later incorporated to the parser, in order to improve its performance.

1 Introduction

As NLP-based applications are growing, there is a stronger need for wide-coverage parsing systems. At the moment, comprehensive grammars are available for some languages, like English [Briscoe and Carroll 1993], or parallel LFG German, French and English grammars [Butt et al., 1999], developed after a considerable effort. Moreover, the huge size of the now available corpora demands successive extensions of the grammars, to include corpus-specific information or to augment the basic syntactic grammars with lexical information, like subcategorization frames or selectional restrictions [Briscoe and Carroll 1997, Carroll and Rooth 1998]. However, the situation is different for most other languages, due to several reasons:

- 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.

Although there are current efforts for the development of parsing systems for other languages [Oflazer 1999, Hajic and Hladká 1998], there will always be the problem of reaching the complexity and performance of the parsers for the most studied languages. This is in spite of the efforts to make publicly available language resources (ELRA) that could at most alleviate the problem. Therefore, methods must be devised which obtain results automatically, minimizing development costs.

This work presents both the development of a parsing system for unrestricted Basque texts and the first results obtained using it in the process of acquiring subcategorization information. The system is applied to Basque, which has as its main characteristics being agglutinative and having basically constituent-free order. These characteristics involve some complexities for syntactic analysis.

As a first step, a basic parsing system has been developed. It consists of two modules, applied sequentially: an unification based chart-parser and a finite-state parser. This combined system covers the syntactic core of the language; however, although it is useful for several non-trivial applications like the detection of syntactic errors, is still incomplete, lacking important aspects like subcategorization information. For this reason, we have applied this basic parser to text corpora, with the aim of obtaining subcategorization information that will be used to enrich the lexical database. This way, we plan to develop a parsing system in a bootstrapping fashion, with incremental improvements.

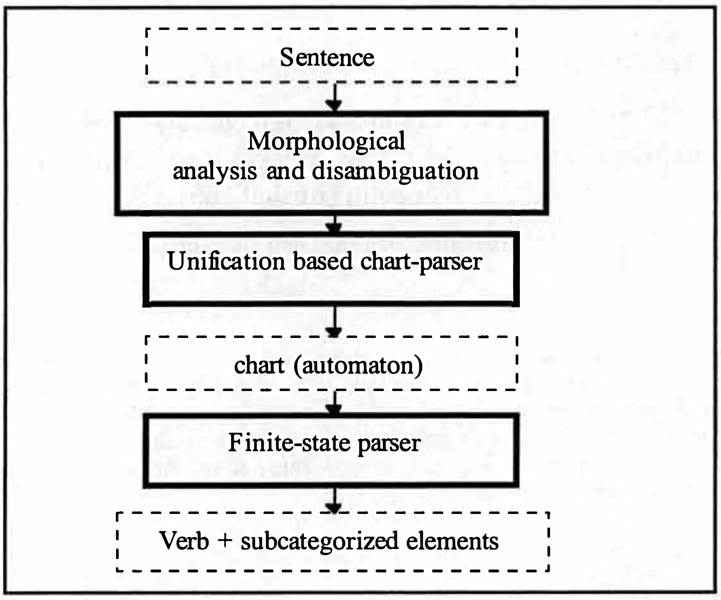
Sentence
Morphological analysis and disambiguation
Unification based chart-parser
chart (automaton)
Finite-state parser
Verb + subcategorized elements

Figure 1. Overview of the system.

The rest of the paper is organized as follows. Section 2 presents the basic parsing system we have implemented, detailing its main components and justifying its sequential architecture. It also examines the application of the system to the extraction of subcategorization information. Section 3 gives the results of its evaluation against a set of manually tagged 500 sentences, while section 4 reviews the literature on parsing systems and automatic acquisition of subcategorization information.

2 The Parser

We have developed a parsing system divided in two main modules: a unification based parser and a finite-state parser (see figure 1). Prior to parsing, there is another step concerned with morphological analysis and disambiguation, using the basic tools for Basque (http://ixa.si.ehu.es) that have been developed in previous projects:

- The lexical database. 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 segmentation. Inflectional morphology of Basque was completely described in [Alegria *et al.* 1996]. This system applies Two-Level Morphology 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, capable of treating unknown words and non-standard forms (dialectal variants and typical errors).

- 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 (Constraint Grammar (CG) formalism [Samuelsson and Voutilainen 1997]). This tool reduces the high word-level ambiguity from 2.65 to 1.19 interpretations, still leaving a number of interpretations per word.

2.1 The Unification Based Parser

After morphological analysis and disambiguation, each word is assigned one or more readings, each of them as a list of its components (lemma and morphemes) with their associated morphosyntactic information. Some facts concerning syntactic analysis must be taken into account:

- The morpheme is the basic unit of syntactic analysis, following the most extended syntactic descriptions for Basque [Abaitua 1988]. In figure 2, dashed lines represent lemmas and morphemes, that is, units smaller than the word that will form the input to the syntactic analyzer. This kind of analysis has been adopted by other systems for agglutinative languages like Hungarian [Prószéky 1996] and Turkish [Oflazer 1999].
- Regarding the syntactic structure of Basque, it has been considered as a language with free order of constituents. However, this is only true for main sentence constituents with respect to the verb (such as noun phrases, prepositional phrases and sentential complements), because inside those constituents the order of elements is fixed or quite limited. Moreover, the syntactic relationships inside these fixed order components require the testing of complex agreement and the building of non-trivial syntactic structures, not definable by finite-state techniques [Beesley 1998]. These facts led us to describe the syntax of these components by means of feature structures, using a unification based formalism.
- There are more problems if we want to go beyond the level of the main sentential constituents (verbs, NPs, PPs and sentential complements). As their relative order is almost free, their analysis would suppose the proliferation of a high number of unsolvable attachment ambiguities. Example 1 shows the presence of two elements (PP and NP) that could be attached to either of the two surrounding verbs (giving three different interpretations). Although this kind of ambiguity can be resolved in some cases (the auxiliary verb, when present, can indicate information about the case, number and person of subject, object and indirect object), a general solution will need at least the use of subcategorization information, unavailable at the moment. As a result, the effort devoted to the design of such a grammar would be of little final value at the moment, due to unsolved ambiguity. Furthermore, its development would also be a costly enterprise. For that reason, we decided to postpone the development of that part of the grammar until the relevant information is available.

... it was seen necessary to create an institution at this side of the Pyrenees ...

... beharrezko	*ikusi zen*	*Pirineotako bazter honetan*	*erakunde bat*	*sortzea ...*
Adjective	Verb	PP	NP	Verb
(necessary)	(was seen)	(at this side of the Pyrennees)	(an institution)	(to create)

Example 1. The attachment of elements between the two verbs needs (at least) subcategorization information.

Hence, a partial unification grammar has been developed that gives a complete coverage of the main elements of the sentence (NPs, PPs and sentential complements). At the moment the grammar contains 120 rules written in the PATR-II formalism. We chose this formalism because there has not been a broad description for Basque using more elaborated theories like LFG [Abaitua 1988] and HPSG, and also because the theories are based on information not available at the moment, such as verb subcategorization. PATR-II is more flexible at the cost of extra writing, as it is defined at a lower level. There is an average number of 15 equations per rule, some of them for testing conditions like agreement, and others for structure building. The main phenomena covered are:

- Noun phrases and prepositional phrases. Agreement among the component elements is verified, added to the proper use of determiners.
- Subordinate sentences, such as sentential complements (completive clauses, indirect questions, ...) and relative clauses.
- 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.

Example 2 shows the (simplified) rule that combines a noun-group (noun + adjectives + determiners + noun modifiers) with a case mark (simple or composed), forming an indefinite NP or PP. Although we will not explain the rule in detail, the example shows the relative complexity of the rules as they must test for several kinds of agreement on number, definiteness and case. As a consequence, the linguistically relevant morphosyntactic information is very rich compared to most chunking systems. This will have the effect of increased flexibility, as different applications will typically use only a subset of the information.

```
rule NP_1_def
X0 ---> X1 X2
        X1/category                       <=>   noun-group
        X2/category                       <=>   case-morpheme
        X0/category                       <=>   NP
        X1/sint/agr/def                   <=>   X2/sint/agr/def
        X1/sint/agr/num                   <=>   X2/sint/agr/num
        X1/head/plu                       <=>   minus
        X2/sint/agr/case                  not   [genitive]
        X2/sint/agr/def                   <=>   indefinite
        or[X1/sint/det/head/subcategory   in    [definite, indefinite, interrogative]
           X1/head/subcategory            in    [cpronoun, interrogative-pronoun]
           X2/sint/agr/case               in    [partitive, prolative, inessive]
           X1/head/subcategory            in    [place-name, proper-name]
                                    ...
```

Example 2. Rule that combines a noun-group with a case-morpheme.

This system can be seen as a shallow parser [Abney 1997, Giguet and Vergne 1997] that can be used for subsequent processing, following "... *the basic assumption that it is possible to define an interesting intermediate level between words and sentences*", as [Basili *et al.* 1998] point out. The parser is applied bottom-up to each sentence, giving a chart as a result. The output for each sentence still contains both morphological and syntactic ambiguities, giving a huge number of different potential readings per sentence. Figure 2 shows an example where dashed lines are used to indicate lexical elements (lemmas or morphemes), while plain lines define syntactic ones. The bold circles represent word-boundaries, and the plain ones delimit morpheme-boundaries. Although the figure has been simplified, each arc is actually represented by its morphological and syntactic information, in the form of a sequence of feature-value pairs.

2.2 The Finite-State Parser

As we showed in the previous section, the unification based parser obtains the decomposition of a sentence into its main syntactic components. However, this result is not directly useful due to several reasons:

- Ambiguity. There are multiple readings for each sentence, as a result of both morphological ambiguity (1.19 interpretations per word-form) and syntactic ambiguities introduced by the unification based parser.
- Different output profiles. Linguistic information is defined at different levels, each of which will be useful depending on a particular application. For example, in the acquisition of subcategorization information all NPs, PPs and sentential complements will be necessary, but for term identification only NPs and PPs are needed (this means that sentential complements, which may include NPs and PPs, must be eliminated from the output).

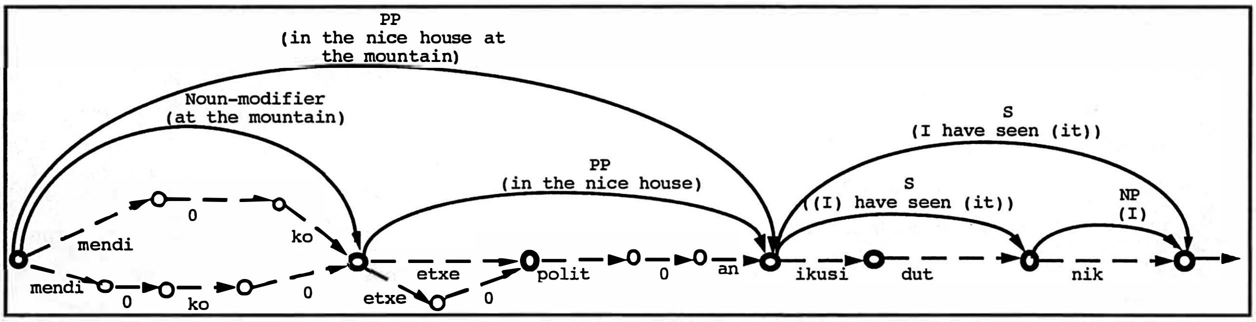

Figure 2. State of the chart after the analysis of *Mendiko etxe politean ikusi dut nik* ('I have seen (it) in the nice house at the mountain').

As a consequence, a tool is needed that will allow the definition of complex linguistic patterns for disambiguation and filtering. In recent years, several parsing systems based on finite-state technology have been developed, based on automata and transducers [Roche and Schabes 1997]. We decided to treat the resulting chart (see figure 2) as an automaton to which finite-state disambiguation constraints and filters can be applied, encoded in the form of regular expressions and relations. This way, finite-state rules provide a modular, declarative and flexible workbench to deal with the resulting chart. Currently we use the Xerox Finite State Tool (XFST, http://www.rxrc.xerox.com/research/mltt/fst/home.html), which has as its main characteristic a rich set of operations, like the replacement operator [Karttunen *et al.* 1997], defined in terms of simpler regular expressions (or relations) so that the combined expressions always belong to the finite-state calculus and can, therefore, be implemented using a finite-state automaton (transducer).

Among the finite-state operators used we apply composition, intersection and union of automata and transducers. We use both ordinary composition and the recently defined *lenient composition* [Karttunen 1998]. This operator allows the application of different eliminating constraints to a sentence, always with the certainty that when some constraint eliminates all the interpretations, then the constraint is not applied at all, that is, the interpretations are 'rescued'. The operator was first proposed to formalize Optimality Theory constraints in phonology. As Karttunen points out, it also provides a flexible way to enforce linguistic or empirical constraints in syntactic disambiguation.

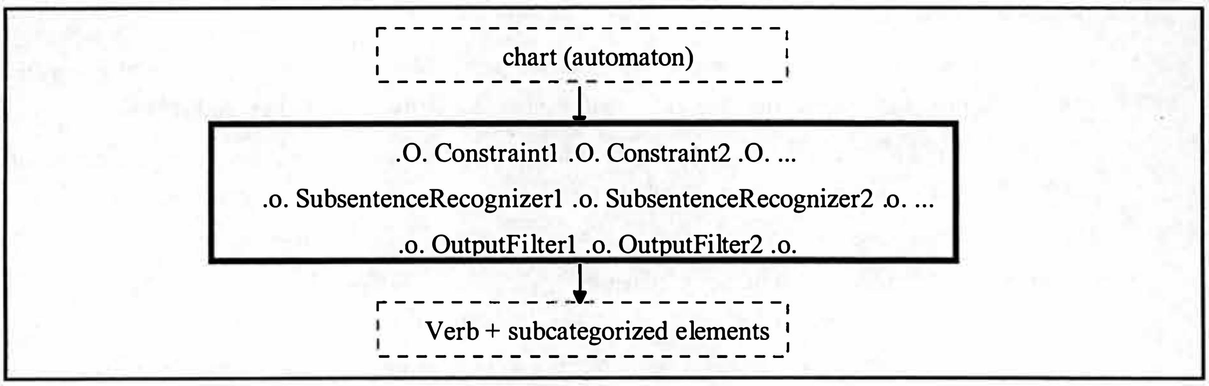
chart (automaton)
.O. Constraint1 .O. Constraint2 .O. ...
.o. SubsentenceRecognizer1 .o. SubsentenceRecognizer2 .o. ...
.o. OutputFilter1 .o. OutputFilter2 .o.
Verb + subcategorized elements

Figure 3. The finite-state parser.

The design of the finite-state rules is a non-trivial task when dealing with real texts, including proper names, syntactic/spelling errors, unknown/foreign words and a wide variety of syntactic constructions. So we had to define 388 finite-state definitions and constraints for the acquisition of verb subcategorization information (actually most of them reflect linguistic facts that can be directly used in other applications). They range from simple local constraints (305 automata with less than 100 states) to most complex patterns (there are a few automata with more than 15,000 states and 300,000 arcs).

As constraints and filters are defined by means of automata and transducers, they could theoretically be merged into a single final automaton, hence improving performance. However, as patterns are more complex, the size of the resulting automaton grows prohibitively large, so we had to arrange it by sequencing the automata. Although this slows down parsing time, it makes the compilation viable [Tapanainen 1997]. The interaction of different automata is a matter that requires further investigation.

As a first evaluation of the system, we chose the problem of acquiring verbal subcategorization information, that is, given a sentence and a verb, extracting the verb's corresponding subcategorized elements. This application has the advantage of a well defined environment to test the performance of the parser, and also that the resulting subcategorization frames may be fed back to the parser, to improve its coverage and precision [Briscoe and Carroll 1997].

These are the main operations performed by the finite-state parser (see figure 3):

- Disambiguation. As whole syntactic units can be used, this process is similar to that of Constraint Grammar disambiguation, with the advantage of being able to reference syntactic units wider than the word, which must be defined in a roundabout manner in the word-based CG formalism. As figure 3 shows, the disambiguation constraints are applied using the lenient composition operator (.O.), so that no constraint will discard all the readings of a sentence, making the system robust.
- Extracting parts of a sentence. The global ambiguity of a sentence is considerably reduced if only part of it is considered (see example 3). For instance, in the case of extracting verb subcategorization information, some rules examine the context of the target verb and define the scope of the subsentence to which the disambiguation operations will be applied (these filters use the ordinary composition operator .o.).
- Filtering. Sometimes not all the available information is relevant. For example, the *noun/adjective* ambiguity present in *zuriekin* ('with the whites' (*zuri* as a noun) / 'with the white ones' (adjective)) can be ignored when acquiring verb subcategorization information, as we are interested in the syntactic category and the grammatical case (prepositional phrase and commitative, respectively), the same in both alternatives.

Example 3 shows the application to a sentence containing the wordform *doa* (goes), in the context of analyzing the verb *joan* (to go). The result is simplified, as both the input and output are presented as text, rather than as an automaton containing feature-value pairs that represent syntactic components (the translation of the output subsentence is given later in example 4).

Eta lepo honetatik Narbajara bidegarbia doa Negurako pagaditik
Larraingoitiko hareharrizko harrobitik zehar igaroaz

↓

(lepo honetatik) (Narbajara) (bide garbia) (doa)

Example 3. Simplified input and output of the parser.

3 Evaluation

We took a corpus consisting of 500 sentences corresponding to 5 verbs, that is, 100 sentences per verb. In order to test different corpora, half of the sentences were taken from a general corpus of Basque, while the other half came from newspaper texts. We manually marked for each sentence the occurrence of each target verb and its associated subcategorized elements, and then compared it with the output of the parser. 350 sentences were used for the refinement of both the unification based parser and the finite-state parser, while the remaining set of 150 sentences (30 for each verb) was only examined for the final test.

Regarding the tested sentences, we did not select them by lexical or syntactic coverage of the parser, i. e., we took the first set of 500 sentences containing the target verbs from the two corpora, so that we could measure the actual performance of the parser with unrestricted texts. There are several extra difficulties added to the problem of ambiguity:

- Sentence length. Each sentence contains an instance of the target verb together with other main or subordinate subsentences (the average sentence length is 22 words). Delimiting the exact boundaries of the subsentence corresponding to the target verb is a difficult task.
- Multiword lexical units. Although we plan to include their treatment in the morphological analysis phase, it is not implemented yet. Its main consequence will be an increase in the number of errors (false positives), as some non-compositional elements will be interpreted compositionally by the general unification based grammar.
- Unknown words, proper names and spelling errors. Although the morphological analyzer recognizes a subset of them, the rest is problematic because each of them will give a number of hypothetical interpretations, therefore increasing ambiguity and consequently the error rate. Increasing lexical coverage will have a positive impact in future developments.

To evaluate the correct analysis of a sentence, we have developed a simple coding scheme inspired on [Carroll *et al.* 1999], who define a hierarchy of grammatical relations. Instead of marking syntactic functions, we annotate the declension case, lemma and number of NPs and PPs [Oesterle and Maier-Meyer 1998], and the subordination type for sentential complements (see example 4). We have postponed the assignment of syntactic functions until relevant data is available.

Input: ... *lepo honetatik Narbajara bide garbia doa* ... (... a clear path goes from this hill to Narbaja ...)

Output:	*lepo honetatik*	*Narbajara*	*bide garbia*	*doa*
	ablative(lepo, doa)	*alative(Narbaja, doa)*	*absolutive(bide, doa)*	*doa*
	from this hill	to Narbaja	a clear path	goes
	ablative(hill, goes)	alative(Narbaja, goes)	absolutive(path, goes)	goes

Example 4. Coding scheme based on the grammatical case of subcategorized elements.

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). Table 1 shows the results as the mean over all sentences. Although there is always a balance between recall and precision, we tried to maximize the latter, sometimes at the cost of lowering recall. As we could inspect the development corpus during the refinement of the parser, the results in the second and third columns can be understood as an upper limit of the parser in its current state, approximately 92% precision and 71% recall. As we will explain next, these results can be improved refining the lexicon and the grammars.

	Development corpus (350 sentences)		**Test corpus (150 sentences)**	
	Precision	**Recall**	**Precision**	**Recall**
agertu (to appear)	95%	69%	87%	62%
atera (to go)	91%	65%	92%	64%
erabili (to use)	92%	70%	86%	55%
ikusi (to see)	91%	76%	87%	78%
joan (to go)	93%	74%	83%	70%
Total	92%	71%	87%	66%

Table 1. Evaluation results.

We examined manually the causes of the errors (68 errors were identified in the test corpus causing problems in precision or recall), which can be classified into several types[1]:

- Errors due to multiword units (5), unknown words, proper names (9) and spelling errors (8). Their treatment corresponds naturally to morphological analysis and is mainly linked to future extensions of the lexicon.
- Errors due to incorrect disambiguation. They can be subdivided into two main types. When the morphological disambiguator chooses an incorrect reading, it has the effect of causing a false positive, i.e., decreasing precision (9 errors). On the other hand, sometimes more than one alternative is left (including the correct one). Its main effect will be increased ambiguity that will show as lower recall (5 errors).
- Errors due to the lack of syntactic coverage of the grammars (32 errors). This kind of errors define the limits of the partial parsing approach. Although more than half of these errors are due to the incompleteness of the grammar, and they can be solved simply by extending it to cover the

[1] We did not examine the relationships among different errors, as many times one kind of error has the effect of causing other error types.

corresponding phenomena, there are other errors that would need qualitative changes, like the inclusion of subcategorization information. Finally, a third set of errors are due to the characteristics of unrestricted corpora, such as syntactically odd constructions, that we doubt a parser could analyze even after solving the two other problems.

As the results show, more than half of the errors could be solved by improvements on the lexicon, the morphological analyzer and morphological disambiguation (totaling 36 errors). Although morphological disambiguation is relatively difficult to extend and modify, further careful work extending the treatment of proper names, spelling errors and multiword units would imply a noticeable increase in both recall and precision. In a similar way, work must be done extending the basic syntactic grammar, which we estimate could reduce the syntactic errors to about a half of the present ones. Consequently, we consider the results satisfactory, with 87% precision and 66% recall, as the results for new sentences are near the expected best results (those obtained for the development corpus, with 92% precision and 71% recall), showing that the system behaves correctly with unseen sentences.

After obtaining instances of putative subcategorization frames, there is still work to be done. In configurational languages like English, subcategorized elements appear at fixed places around the verb, while in nonconfigurational ones they can appear at several different positions (hypothetically all the permutations are possible). Basque being mainly a nonconfigurational and pro-drop language with respect to phrases in ergative, absolutive and dative cases, there is not a direct correspondence between subcategorization instances and frames, as one subcategorization frame may correspond to several kinds of instances. As an experiment, we applied the parser to 1,000 sentences (22,000 words) corresponding to the verb *ikusi* ('to see'), and classified the results according to the different sets of subcategorized elements, without taking their relative order in consideration. The results are presented in Table 2. Results were obtained for 826 sentences, after discarding those having more than one interpretation. For example, the patterns 'instrumental' and 'absolutive instrumental' correspond to the same subcategorization frame, due to pro-drop phenomena with the phrase in the absolutive case. The possibility of automatically classifying subcategorization patterns into frames deserves further work.

Subcategorization pattern	**# of occurrences**
absolutive	206
inessive	59
inessive absolutive	42
ergative	36
instrumental	14
ergative absolutive	11
absolutive instrumental	6
absolutive inessive ergative	4

Table 2. Different patterns found in the corpus.

4 Related Work

[Abney 1997, Giguet and Vergne 1997, Basili *et al.* 1998] show the benefits of a stratified approach to parsing, where one or more intermediate levels can be defined between the basic level of words and the analysis of a full sentence. Our work differs from theirs in that we apply two different kinds of analyzers (unification based and finite-state), rather than defining the different levels using the same formalism.

[Ritchie *et al.* 1992] present a system that performs morphological analysis, divided in a segmentation phase (using finite state networks) and the application of a unification grammar for the combination of morphemes. The results of the segmentation are interpreted as a chart that serves as input to a unification based chart parser. Our system shares the idea of dividing work between different kinds of formalism. However, our approach differs in that we first apply a unification grammar, indispensable for the treatment of complex syntactic phenomena, and then a finite state grammar is used for disambiguation and filtering.

Regarding the problem of syntactic disambiguation, most grammar based systems [Briscoe and Carroll 1997] have adopted a statistical approach. For morphological disambiguation, however, there are both statistical and rule based systems, with better results for the second approach [Samuelsson and Voutilainen 1997]. Our system adopts a rule formalism based on regular expressions, using syntactic elements instead of words as the basic disambiguation unit. We justify this election on both the unavailability of syntactically annotated treebanks and the better performance of systems based on hand-coded rules.

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] describe a grammar based experiment for the extraction of subcategorization frames with their associated relative frequencies, obtaining 76.6% precision and 43.4% recall. Our results are not directly comparable, as we only estimate precision and recall on subcategorization instances, not frames.

[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 wordforms) 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 use patterns on partially parsed sentences. 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.

The above mentioned studies depend on a set of manually annotated analyses. [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 (50M).

5 Conclusion

This work presents the development of a robust parser for unrestricted Basque texts. As the linguistic resources are limited, the lexicon lacks important aspects such as verbal subcategorization information. We have implemented a basic syntactic parser using the information now available in the lexicon. The system has been divided in two sequential modules:

- A unification based grammar that covers the main sentence components of the sentence. It gives a description of well-formed linguistic phenomena. Due to the agglutinative nature of the language, feature structures are necessary to treat the wealth of information contained in words/morphemes.
- Finite-state rules that provide a modular, declarative and flexible workbench to deal with the resulting chart of syntactic elements. It establishes the application of empirical, corpus-oriented facts, versus the more general facts on linguistic well-formedness encoded in the unification grammar.

The unification based grammar and the finite-state one are complementary. The unification grammar is necessary to treat aspects like complex agreement and constituent order variations, currently unsolvable using finite-state networks, due to the exponential growth in size of the resulting automata [Beesley 1998]. The limits of this grammar are mainly defined by the unavailability of important information, like subcategorization frames. On the other hand, regular expressions and relations, in the form of automata and transducers, are indispensable to cope with morphological/syntactic ambiguity (by means of hand-coded rules or constraints) and filtering of the information relevant to each application, thus adding to the flexibility of the resulting tool.

The parser is being used in the process of acquiring verb subcategorization instances, obtaining 87% precision and 66% recall over a corpus of previously unseen 150 sentences. In future work, we plan to integrate the resulting subcategorization information into the grammar, so that it will be extended by successive bootstrapping cycles.

Acknowledgements

This research is supported by the Basque Goverment, the University of the Basque Country and the Interministerial Commision for Science and Technology (CICYT). Thanks to Gorka Elordieta for his help writing the final version of the paper.

Bibliography

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

[Abney 1997] Abney S. P. 1997. Part-of-Speech Tagging and Partial Parsing. in Corpus-Based Methods in Language and Speech Processing, Kluwer, Dordrecht.

[Aldezabal *et al.* 1999] Aldezabal I., Gojenola K., Oronoz M. 1999. Combining Chart-Parsing and Finite State Parsing. Proceedings of the ESSLLI'99 Student Session, Utrecht.

[Alegria *et al.* 1996] Alegria I., Artola X., Sarasola K., Urkia M. 1996. Automatic morphological analysis of Basque. Literary and Linguistic Computing 11 (4).

[Basili *et al.* 1998] Basili, R., Pazienza M.T., Zanzotto F.M. 1998. Efficient Parsing for Information Extraction. Proceedings of the 13th European Conference on Artificial Intelligence, John Wiley & Sons Ltd.

[Beesley 1998] Beesley K. 1998. Constraining Separate Morphotactic Dependencies in Finite-State Grammars. Proceedings of the International Workshop on Finite State Methods in Natural Language Processing, Ankara.

[Briscoe and Carroll 1993] Briscoe T., Carroll J. 1993. Generalized Probabilistic LR Parsing of Natural Language (Corpora) with Unification-Based Grammars. Computational Linguistics, vol. 19(1).

[Briscoe and Carroll 1997] Briscoe T., Carroll J. 1997. Automatic Extraction of Subcategorization from Corpora. ANLP'97, Washington.

[Butt *et al.* 1999] Butt M., King T.H., Nino M.E., Segond F. 1999 A Grammar Writer's Cookbook. Stanford, CA: CSLI Lecture Notes, CSLI Publications, 1999.

[Carroll and Rooth 1998] 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 *et al.* 1999] Carroll J, Minnen G., Briscoe T. 1999. Corpus Annotation for Parser Evaluation. Proceedings of Workshop on Linguistically Interpreted Corpora, EACL'99, Bergen.

[Ezeiza *et al.* 1998] 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.

[Giguet and Vergne 1997] Giguet E., Vergne J. 1997. From Part of Speech Tagging to Memory-based Deep Syntactic Analysis. Fifth International Workshop on Parsing Technologies, Boston.

[Grishman *et al.* 1994] Grishman R., Macleod C., Meyers A. 1994. Comlex Syntax: Building a Computational Lexicon. COLING'94, Japan.

[Hajic and Hladká 1998] Hajic J., Hladká B. 1998. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. COLING-ACL'98, Montreal.

[Karttunen *et al.* 1997] Karttunen L., Chanod J-P., Grefenstette G., Schiller A. 1997. Regular Expressions For Language Engineering. Natural Language Engineering.

[Karttunen 1998] Karttunen L. 1998. The Proper Treatment of Optimality in Computational Phonology. Proceedings of the International Workshop on Finite State Methods in Natural Language Processing, Ankara.

[Kuhn *et al.* 1998] Kuhn J., Eckle-Kohler J., Rohrer. C. 1998. Lexicon Acquisition with and for Symbolic NLP-Systems -- a Bootstrapping Approach. Int. Conference on Language Resources and Evaluation (LREC98), Granada.

[Oesterle and Maier-Meyer 1998] Oesterle J., Maier-Meyer P. 1998. The GNoP (German Noun Phrase) Treebank. LREC98, Granada.

[Oflazer 1999] Oflazer K. 1999. Dependency Parsing with an Extended Finite State Approach. ACL'99, Maryland.

[Prószéky 1996] Prószéky G. 1996. Morphological Analyzer as Syntactic Parser. COLING'96, Copenhagen.

[Ritchie *et al.* 1992] Ritchie G., Russel G. J., Black A., W., Pullman S. G. 1992. Computational Morphology: Practical Mechanisms for the English Lexicon. The MIT Press.

[Roche and Schabes 1997] Roche R., Schabes Y. 1997. Finite-State Language Processing. MIT Press.

[Samuelsson and Voutilainen 1997] Samuelsson C., Voutilainen A. 1997. Comparing a Linguistic and a Stochastic Tagger. ACL-EACL'97, Madrid.

[Tapanainen 1997] Tapanainen P. 1997. Applying a Finite-State Intersection Grammar. Finite-State Language Processing, MIT Press.