# Finite State Applications for Basque

I. Aduriz*, I. Aldezabal, I. Alegria, J. M. Arriola, A. Diaz de Ilarraza, N. Ezeiza, K. Gojenola
Ixa Taldea. University of the Basque Country
649 Postakutxa. E-20080 Donostia. Basque Country.
i.alegria@si.ehu.es / http://ixa.si.ehu.es
*Dept. of General Linguistics. University of Barcelona
E-08007 Barcelona

**Abstract**

We present the application of finite state technology (FST) to several kinds of linguistic processing of Basque, which can serve as a representative of agglutinative languages and languages with free order of constituents. Three main tools will be described in this context: a morphological analyzer, a morphosyntactic disambiguation tool and a surface syntax processor.

## 1 Introduction

Finite state technology has been applied to build three basic tools for Basque: morphological analysis/generation (section 2), morphosyntactic disambiguation (section 3) and surface syntax (section 4). We have developed several applications based on these tools (section 5).

All these works have been developed in the IXA group from the University of the Basque Country. The goal of this group is to do research and development of tools in the area of Natural Language Processing for Basque.

Figure 1 shows an overview of the tools developed, where the finite-state components are highlighted.

In the rest of this section we will give a brief description of Basque language with the aim of showing the main characteristics of the language related to the computational application.

### 1.1 A brief description of Basque

Basque is an agglutinative language, that is, for the formation of words the dictionary entry independently takes each of the elements necessary for the different functions (syntactic case included). More specifically, the affixes corresponding to the determinant, number and declension case are taken in this order and independently of each other. These elements appear only after the last element in the noun phrase. One of the main characteristics of Basque is its declension system with numerous cases, which differentiates it from languages spoken in the surrounding countries.

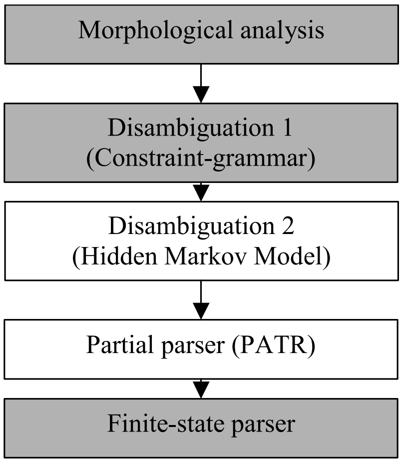


Figure 1. Scheme of all the components.

At sentence level, too, the verb appears as the last element in a neutral order. That is, given the language typology proposed by Greenberg, Basque is a Subject-Object-Verb (SOV) type language (Laka, 1998) or a final head type language. However, this corresponds to the neutral order, but in real sentences any order of the sentence elements (NPs, PPs) around the verb is possible, that is, Basque can also be considered a language with free order of sentence constituents.

These are the principal features that characterize the Basque language and, obviously, they will strongly determine the kind of design decisions to be taken in order to develop tools and applications for NLP of Basque.

## 2 Morphological analysis

In order to build a general-purpose morphological analyzer/generator for Basque we have used a multilevel design, which combines robustness and avoids overgeneration.

Important features of this design are homogeneity, modularity and reusability because the different steps are based on lexical transducers (Karttunen, 1994), far from ad-hoc solutions.

The analyzer consists of three main modules:

1. The standard analyzer (section 2.1), which uses general and user's lexicons. This module is able to analyze/generate standard language word-forms. To deal with the user's lexicon the general transducer described below (see section 2.3) is used.
2. The analysis and normalization of linguistic variants (dialectal uses and competence errors). This problem becomes critical in languages like Basque in which standardization is in process and dialectal forms are of widespread use. For this process the standard transducer is extended producing the *enhanced transducer*.
3. The guesser. In this case the standard transducer is simplified removing the lexical entries in open categories (names, adjectives, verbs, …), which constitute the vast majority of the entries, and is substituted by a general automata to describe any combination of characters, producing the *general transducer*.

In the next three subsections we will present the corresponding transducers.

### 2.1 The standard transducer

Three transducers are combined using composition in order to build the standard analyzer, which attaches to each input word-form all possible interpretations and its associated information.

1. **FST1 or Lexicon**. Over 70,000 entries have been defined corresponding to lemmas and affixes, grouped into 170 sublexicons. Each entry of the lexicon has, in addition to the morphological information, its continuation class. Lexical entries, sublexicons and continuation classes all together define the morphotactics graph. The upper side of the transducer is the whole morphological information, and the lower side is composed of the morphemes and the minimal morphological information to control the application of the other transducers in cascade (FST2 and FST3).
2. **FST2**: constraints on long-distance dependencies. Some dependencies among morphemes cannot be expressed with continuation classes because co-occurrence restrictions exist between morphemes that are physically separated in a word (Beesley, 1998). Three rules have been written to solve long-distance dependencies of morphemes: one in order to control hyphenated compounds, and two to avoid both prefixed and suffixed causal conjunctions
3. **FST3**: set of morphophonological rules. 24 two-level rules have been defined to express the morphological, phonological and orthographic changes between the lexical and the surface levels that happen when the morphemes are combined. Details can be consulted in (Alegria et al., 1996).

### 2.2 The enhanced transducer

A second morphological subsystem, which analyzes, normalizes, and generates linguistic variants, is added in order to increase the robustness of the morphological processor. This subsystem is the result of the composition of three transducers:

1. **FST1***: New morphemes linked to their corresponding standard ones in order to normalize or correct the non-standard morphemes are added to the standard lexicon. Thus, using the new entry *tikan*, dialectal form of the ablative singular

morpheme, linked to its corresponding right entry *tik* the system will be able to analyze and correct word-forms such as *etxetikan*, *kaletikan*,... (variants of *etxetik* ‘from the house’, *kaletik* ‘from the street’, ...). More than 1500 additional morphemes have been included.

2 **FST3***: The standard morphophonological rule-system with a small change: the morpheme boundary (+ symbol) is not eliminated in the lower level in order to use it to control changes in FST4. So, the language at this level corresponds to the surface level enriched with the + symbol.

3 **FST4**: New rules describing the most likely regular changes that are produced in the linguistic variants. These rules have the same structure and management as the standard ones, but all of them are optional. For instance, the rule `h:0 => V:V_V:V` describes that, between vowels, the h of the lexical level may disappear in the surface level. In this way the word-form *bear*, misspelling of *behar* (to need), can be analyzed. It is possible and clearer to put these non-standard rules in another level close to the surface, because most of the additional rules are due to phonetic changes and do not require morphological information. The additional rules do not need to be integrated with the standard ones, and consequently, it is not necessary to solve inconsistencies.

## 2.3 The general transducer

The problem of unknown words does not disappear with the previous transducer. In order to deal with it, a general transducer has been designed to relax the need of lemmas in the lexicon. Daciuk (Daciuk, 2000) proposes a similar way when he describes the *guessing automaton*, but the construction of the automaton is more complex than in our case.

The new transducer is the standard one modified in this way: the lexicon is reduced to affixes corresponding to open categories[1] and generic lemmas for each open category, while standard rules remain. So, the standard rule-system (FST3) is composed of a mini-lexicon (FST0) where the generic lemmas are obtained as a result of combining alphabetical characters and can be expressed in the lexicon as a cyclic sublexicon with the set of letters (some constraints are used with capital/non-capital letters according to the part of speech). Figure 2 shows the graph corresponding to the mini-lexicon (FST0).

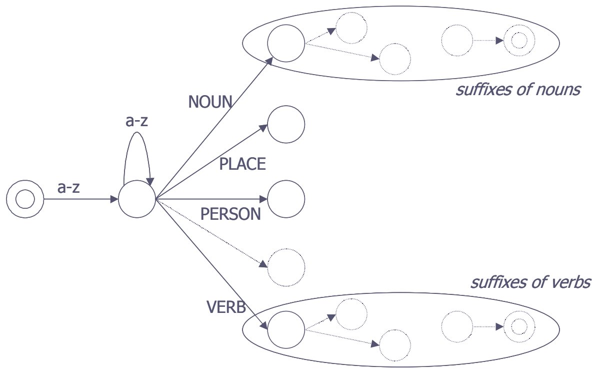


Figure 2- Simplified graph of the mini-lexicon.

This transducer is used in two steps of the analysis: the standard analysis for the user’s lexicon and the guesser.

In order to avoid the need of compiling the user’s lexicon with the standard description, the general transducer is used. If any of the hypothetical lemmas is found in the user’s lexicon then the analysis is added to the results obtained in the standard transducer.

## 2.4 Local disambiguation

As the morphological analyzer may sometimes overgenerate analysis of linguistic variants and unknown lemmas some heuristics have been introduced (Alegria et al., 2001).

The treatment of non-standard words has been added to the previously developed analyzer for two main reasons:

- The average number of interpretations in non-standard words is significantly higher than in standard words.
- There could be multiple lemmas for the same or similar morphological analysis.

[1] There are seven open categories and the most important ones are: common nouns, personal names, place nouns, adjectives and verbs.

This is a problem when we want to build a lemmatizer. For example, if *bitaminiko* (vitaminic) is not in the lexicon the results of the analysis of *bitaminikoaren* (from the vitaminic) as adjective can be multiple: *bitamini+ko+aren*, *bitaminiko+aren* and *bitaminikoaren*, but the only right analysis is the second one.

We think that it is important to reduce the ambiguity at this stage, so that the input of subsequent processes is more precise. However, we do not use information about surrounding words because a contextual tagger will be applied later.

This module consists of different methods for linguistic variants and unknown words, because overgeneration is produced by different facts in each case.

In the case of linguistic variants a heuristic tries to select the lemma that is "nearest" to the standard one according to the number of non-standard morphemes and rules applied. It chooses the interpretation that has less non-standard uses for each POS tag. Oflazer (2003) proposes finite-state based *lenient morphology* to face this problem, and we are doing some experiments in this way.

In order to detect and treat overgeneration in unknown words a combination of different strategies is used:

- Typographical disambiguation. Some analyses are discarded based on capital letters.
- Disambiguation of derivational words to counterbalance overgeneration.
- Identification and disambiguation of proper names not included in the lexicon.
- Disambiguation based on both statistical and linguistic information. Correlations among final character trigrams and POS tags are used.

## 3 Morphosyntactic disambiguation

In agglutinative languages such as Basque morphology is a previous step for disambiguation and tagging.

There are two basic groups of methods that are used in morphosyntactic disambiguation: statistical and rule-based methods. On one hand, statistical methods need little effort and obtain very good results (Church, 1988), at least when applied to English. But when we try to apply them to Basque we find additional problems. On the other hand, some rule-based systems such as the Constraint Grammar parser (Karlsson *et al.*, 1995; Samuelsson and Voutilainen, 1997), are, at least, as good as statistical systems and they are better adapted to free-order languages and agglutinative languages. Therefore, we have selected the rule-based Constraint Grammar (CG) formalism (Karlsson *et al.*, 1995) and the HMM based TATOO tagger (Armstrong *et al.*, 1995).

The CG approach is one of the best examples of tackling natural language (morpho)syntax by means of finite-state devices. We have chosen CG because it was designed with the aim of being a language-independent and robust tool to disambiguate and analyze unrestricted texts. The CG rule statements are close to real text sentences and directly address some crucial parsing problems, especially ambiguity.

The main reasons to select TATOO are its language and lexicon independence, modularity, and flexibility.

### 3.1 Types and rate of ambiguity

Three types of ambiguity may be distinguished:

Firstly, we consider categorial ambiguity, concerning part of speech, like Noun/Verb, Verb/Adjective/Adverb, etc. When the annotation is reduced to the basic 20 categories in the lexicon, we estimated from a corpus that over 36% of the words are ambiguous, with an average ambiguity rate of 1.48 analysis per token.

Secondly, there are several possible morphosyntactic interpretations attached to each input word-form. This is due to declension and other morphosyntactic and lexically non-independent features, including suffixes such as subordinate ones. Taking into account this ambiguity, over 66% of word-forms are ambiguous, with an average ambiguity rate of 2,89 analysis per token. This poses a really hard disambiguation problem. As Karlsson et al. (1995) point out, categorial ambiguity is one of the most pervasive problems facing anybody trying to construct a realistic and successful parser. But in the case of Basque, the morphosyntactic ambiguity of suffixes is added to POS ambiguity.

Finally, we must also consider that there are some cases in which the ambiguity belongs only to syntax, that is, the case in which the ambiguity concerns to syntactic functions. The ambiguity in this case is over 22%.[2]

Despite the classification into three types of ambiguity, it has to be noted that in Basque morphology and syntax are tightly related to each other. The resolution of one type of ambiguity often implies the (partial) resolution of other levels of ambiguity.

### 3.2 The design of a disambiguation grammar

In this section we describe the steps followed in the design of the rules and the results obtained. The main sources for the design of the grammar are the Lexical Database for Basque (Agirre *et al.*, 1995) and the morphological analyzer.

The process to formulate the rules has been carried out in different steps:

- Study of the phenomena of morphosyntactic ambiguity.
- Manual disambiguation of a corpus. Part of a corpus (about 24,000 words) has been manually disambiguated by two different linguists and the results were compared, applying the "double blind" method described in (Voutilainen & Järvinen, 1995).
- Design of disambiguation rules for the cases previously established. These rules were formulated, implemented, and tested using a part of the manually disambiguated corpus (14,000 words). The rest of the corpus (10,000 words) was used for testing.

At the moment, there are 1,116 morphosyntactic disambiguation rules grouped in four sections, depending on the goal and the confidence level. All these sections constitute a unique grammar and are applied one after the other. The user may select the sections depending on the needs of particular applications.

The results of the disambiguation are calculated on a 27,000-word test corpus that was neither previously examined nor used for the development of the rules.

Regarding categorial ambiguity, the number of interpretations is reduced from 1.48 to 1.14 analysis per token, maintaining a 99.41% of the correct interpretations. The percentage of ambiguous words is reduced from 36.03% to 13.09%.

Regarding the ambiguity of the full output of the morphological analyzer, the ambiguity is reduced from 66% to 39% ambiguous words and from 2.89 analysis per token in the input to 1.81 in the output, maintaining 98.44% of the correct interpretations.

Finally, the ambiguity related to the syntactic functions is reduced from 22% to 4% and from 1.71 to 1.13 average syntactic functions.

These results give us an idea of the robustness of the tool on real texts, achieving satisfactory accuracy. This reduction of the ambiguity is fairly good if we take into account that the original ambiguity rate is very high, compared to other works (Voutilainen, 1995). The remaining ambiguity is quite important, but using a stochastic post-process module will further reduce it (Ezeiza et al., 1998).

## 4 Syntactic analysis

In this section we will present the application of a finite state syntactic grammar to the acquisition of verb subcategorization information from a corpus of newspaper texts (Aldezabal et al., 2003).

### 4.1 Description

This work corresponds to the specific task of detecting verbs and their associated sentence components. The application works at a level higher than morphology, as previously built syntactic units must be dealt with, but only with a restricted subset of syntax: the relation of verbs with their adjuncts and complements. The task is complex because many problems must be taken

[2] It has been manually estimated taking into account the syntactic functions of 200 common words.

into account, such as ambiguity and determination of clause boundaries, among others. This way, without developing a full syntactic parser, we have been able to obtain useful results that are accurate enough. The final result will be a set of verb instances together with their syntactic dependents (arguments or adjuncts) for each sentence in the corpus. The information thus obtained will be the input to statistical filters in order to test their adequacy to distinguish arguments from adjuncts, with the aim of automatically obtaining verb subcategorization frames (Briscoe and Carroll 1997; Korhonen 2001).

As a starting point we recurred to shallow parsing (Abney, 1997). This allowed us to create a parser with a limited effort at the cost of losing coverage. After morphological analysis and disambiguation are performed (see Figure 1 in the introduction), words that form adjacent syntactic units can be grouped, such as NPs, PPs, verb chains and sentential complements, using morphological and syntactic tags. A unification grammar (Aldezabal et al., 2000) based on the PATR formalism containing 96 rules has been defined for that purpose.

However, the output of this phase is not directly usable, due mainly to two difficulties:

- Ambiguity. For example, "*ama polita*" can be interpreted as a single NP and also as two consecutive NPs.
- Difficulty of linking dependents with verbs. Due to the free constituent-order of Basque, syntactic elements appearing between two verbs are problematic, because they could be assigned to both verbs.

For that reason, the output of this stage is converted to an automaton. This is the input to a finite state parser that performs syntactic disambiguation, determination of clause boundaries and filtering of the results.

To implement the finite-state grammar we have applied operations on regular expressions and relations, among them composition and replacement, using the Xerox Finite State Tool (Karttunen et al. 1997). These are the main operations described by the finite state grammar:

- Clause recognition. In order to extract verb subcategorization information, a group of rules will examine the context of the target verb and define the scope of its associated clause. The global ambiguity is considerably reduced if only the clause corresponding to the target verb is considered. In the following steps the disambiguation operations will be applied to this clause.
- Disambiguation: longest match selection. A usual kind of ambiguity is produced when there are alternative readings of a sentence where some of the syntactic units can be either independent or can also be included inside bigger units.
- General disambiguation constraints. We have also developed a number of constraints that try to reduce several types of ambiguity.

The resulting finite-state grammar contains more than 300 rules (for more details, see Aldezabal et al., 2003). It has been applied to a corpus of 111,000 sentences from newspaper texts, totaling 1,337,445 words. When dealing with unrestricted texts there are several extra difficulties added to the problem of ambiguity, such as multiword lexical units, unknown words, proper names, spelling errors and long sentences (as these sentences contain multiple verbs, delimiting the exact boundaries of the clause corresponding to the target verb is a difficult task, due to free constituent order).

## 4.2 Evaluation

For evaluation we took a set of previously unseen 150 sentences (Aldezabal *et al.* 2000). We measured precision (# of correctly selected dependents / all the elements returned) and recall (# of correctly selected dependents / actual dependents in the sentence), applied to each instance of the target verb and its corresponding complements/adjuncts. Although there is always a balance between precision and recall, we tried to maximize precision, sometimes at the cost of lowering recall. We consider the results satisfactory, with 87% precision and 66% recall.

We examined the causes of the errors manually, concluding that about half of them can be improved by simple refinements of the lexicon and the grammars (both the partial unification-based grammar and the finite-state grammar), while the rest would require qualitative changes on the grammars (such as the incorporation of subcategorization information). The results have been used for the automatic selection of arguments and adjuncts by means of statistical filters (Aldezabal *et al*. 2002).

## 5 Other Applications

Morphological analysis is a basic tool that can be used in a wide range of applications. In the case of Basque, we have developed the following:

- A lemmatizer based on morphology has been developed and used in several applications: an intranet search engine (Aizpurua et al., 2000, www.egunkaria.com/hemeroteka), an assistant for verse making (Arrieta et al., 2001) and a bilingual on-line dictionary.
- Xuxen (Aldezabal et al., 1999) is a spelling checker that accepts as correct any word that allows a correct standard morphological analysis. If one of the possible lemmas returned by the general transducer is in the user's lexicon, the word is accepted. Otherwise, a misspelling is assumed, and using the enhanced transducer accurate proposals are generated. It is possible to include lemmas in the user's lexicon in order to recognize any inflected and derived form without recompiling the transducer.
- The morphological analyzer has also been used in a speech recognition system (López de Ipiña et al., 2002), in order to optimize the size of the dictionary.

The lemmatizer/tagger EUSLEM (Ezeiza et al. 1998) has been used to lemmatize/tag several millions of words of the Basque National Corpus (www.euskaracorpusa.net). It has also been used in experiments for a text-to-speech engine for Basque (Navas *et al*. 2002), The results are useful assigning phrase breaks, but deeper syntactic information is necessary for good intonation.

## 6 Conclusions

We have presented the application of finite state technology (FST) to several kinds of linguistic processing of Basque, which can serve as a representative of agglutinative languages and languages with free order of constituents.

We have shown how FST provides a full solution for morphological analysis, being at this moment the best alternative for morphologically complex languages. The proposals for user-lexicon, enhanced transducer and guesser are applicable to many lesser-studied languages.

In the case of morphosyntactic disambiguation, the Constraint Grammar of Basque outperforms other methods, such as statistical disambiguation (Ezeiza et al. 1998).

We have also investigated the application of FST to syntax, to the specific problem of disambiguation and linking a verb with its dependents, with satisfactory results. However, the development of a full syntactic analyzer based on FST is still a matter of research (Aït-Mokhtar et al. 2002).

Finally, we have presented several applications based on the previously presented basic tools. These applications demonstrate the benefits of linguistically sound methods and formalisms as the core of the linguistic processors.

## Acknowledgements

This research has been partially supported by the European Commission (MEANING IST-2001-34460), the Spanish Ministry of Science and Technology (Hermes TIC2000-0335-C03-03) and the University of the Basque Country (9/UPV00141.226-14601/2002). We would like to thank Xerox (XRCE) for letting us using their tools, and also to Ken Beesley and Lauri Karttunen for their help.

## References

Abney S. P. (1997) *Part-of-speech tagging and partial parsing*. S. Young and G. Bloothooft, editors, Corpus-Based Methods in Language and Speech Processing, Kluwer, Dordrecht.

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 Application of Natural Language to Databases.

Aït-Mokhtar S., Chanod J.-P., Roux C. (2002) *Robustness beyond shallowness: incremental deep parsing*. Journal of Natural Language Engineering, Special Issue on Robust Methods in Analysis of Natural Language Data.

Aldezabal I., Alegria I., Ansa O., Arriola J., Ezeiza N. (1999) *Designing spelling correctors for inflected languages using lexical transducers*. EACL'99.

Aldezabal I., Gojenola K., Sarasola K. (2000) *A Bootstrapping Approach to Parser Development*. International Workshop on Parsing Technologies.

Aldezabal I., Aranzabe M., Atutxa A., Gojenola K., Sarasola K. (2002) *Learnig Argument/Adjunct Distiction for Basque.* ACL'2002 SigLex Workshop on Unsupervised Lexical Acquisition

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.

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.

Alegria I., Aranzabe M., Ezeiza A., Ezeiza N., Urizar R. (2001) *Using Finite State Technology in Natural Language Processing of Basque* Conference on Implementations and Applications of Automata, pp. 2-12. 2001. Pretoria.

Alegria I., Aranzabe M., Ezeiza A., Ezeiza N., Urizar R. (2002) *Robustness and customisation in an analyzer/lemmatizer for Basque*. LREC-2002 Customizing knowledge in NLP applications.

Armstrong S., Russel G., Petitpierre D., Robert G. (1995) *An open architecture for Multilingual Text Processing.* EACL'95. vol 1, 101-106.

Arrieta B., Alegria I., Arregi X. (2001) *An assistant tool for Verse-Making in Basque based on Two-Level Morphology*. Literary and Linguistic Computing, Vol. 16, No. 1

Beesley K. (1998) *Constraining Separated Morphotactic Dependencies in Finite State Grammars*. Proc. of the International Workshop on Finite State Methods in NLP. Ankara.

Briscoe T., Carroll J. (1997) *Automatic extraction of subcategorization from corpora*. Proceedings of the Conference on Applied NLP, Washington.

Church K. W. (1988) *A stochastic parts program and phrase parser for unrestricted text*. ANLP.

Daciuk J. (2000) *Finite State Tools for Natural Language Processing*. Proceedings of the COLING 2000 workshop Using Toolsets and Architectures to Build NLP Systems, Luxembourg.

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, Montreal.

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*. Natural Language Engineering, Cambridge University Press.

Karttunen L. (1994) *Constructing Lexical Transducers*. Proc. of COLING´94, 406-411. 1994.

Korhonen A. (2001) *Subcategorization acquisition*. PhD Thesis, University of Cambridge.

Laka, I. (1998) *A Brief Grammar of Euskara, the Basque Language.* HTML document. http://www.ehu.es/grammar. Office of the Vice-Dean for the Basque Language. University of the Basque Country.

Lopez de Ipiña K., Ezeiza N., Bordel G. (2002) *Morphological segmentation for speech processing in Basque*. *Proceedings of TTS-2002*.

Navas E., Hernaez I., Ezeiza N. (2002) Assigning Phrase Breaks Using CARTs for Basque TTS. *Proceeding of Speech Prosody.*

Oflazer K. (2003). *Lenient Morphological Analysis*. To appear in the Special Issue on Finite State Methods in NLP, Journal of Natural Language Engineering.

Samuelsson C., Voutilainen A. (1997). *Comparing a Linguistic and a Stochastic Tagger*. ACL-EACL'97, Madrid.

Voutilainen A. (1995). *A Syntax-Based Part-of-speech Analyzer*. Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, EACL'95. Dublin.

Voutilainen A., Järvinen T. (1995). *Specifying a Shallow Grammatical Representation for Grammatical* EACL'95. Dublin, Ireland.