MORPHOSYNTACTIC DISAMBIGUATION AND SHALLOW PARSING IN COMPUTATIONAL PROCESSING OF BASQUE

Itziar Aduriz and Arantza Díaz de Ilarraza

Abstract

Our goal in this article is to show the improvements in the computational treatment of Basque, and more specifically, in the areas of morphosyntactic disambiguation and shallow parsing.[1] *The improvements presented in this paper include the following: analyses of previously identified ambiguities in morphosyntax and in syntactic functions, their disambiguation, and finally, an outline of possible steps in terms of shallow parsing based on the results provided by the disambiguation process. The work is part of the current research within the field of Natural Language Processing (NLP) in Basque, and more specifically, part of the work that is being done within the IXA*[2] *group.*

1. Introduction

Morphosyntactic ambiguity is, as is well-known, one of the most difficult problems in NLP. Ambiguity arises from previously done morphological analyses, and hence, it is closely dependent on decisions made at the morphological level. This paper includes a presentation of the systematic analysis of ambiguity in Basque. Also, we briefly describe the morphological analyzer in order to understand how ambiguity arises in Basque, namely from the lexical Basque Database (i.e., Euskararen Datu-base lexikala (EDBL)), which is the basis for all information.

Analyses of morphosyntactic ambiguity are followed by the process of disambiguation of such analyses, which is a crucial step in NLP as mentioned above. For this task, we have created a grammar composed of over one thousand rules by following linguistic criteria. We also present the formalism chosen to carry out this step, namely, Constraint Grammar.

When treating language computationally, syntactic analyses necessarily follow morphological analyses and their disambiguation. Linguistic research has shown that any thorough syntactic analysis on languages encounters difficulties, even when adopting a specific theoretical framework. Matters get much worse in a computational

[1] Shallow Parsing is also known as Partial Syntax. The aim of this type of analysis is to analyze all the structures in a corpus, albeit not in depth.

[2] http://ixa.si.ehu.es

[ASJU Geh 46, 2003, 1-21]
http://www.ehu.es/ojs/index.php/asju

approach to language, and as a solution, several pre-syntactic analyses (or steps parallel to the syntactic analyses) have been proposed. This is the case of the so-called *partial analyses* and the *analysis of parts of sentences.*

2. Analyses of sintagmatic components: disambiguating-grammars and syntactic analyzers

The main difference between morphological and syntactic analyses relates to a change in viewpoint. Whereas a morphological analysis involves a paradigmatic viewpoint, those analyses that come after the morphological level (i.e., those directed to syntax and syntactic analyses themselves) involve a sintagmatic perspective. The former are limited to the word-level, and in contrast, the later target bigger units formed by words: chunks, phrases, clauses, etc. Thus, the onset of the sintagmatic analysis directly follows the morphological analysis. In fact, it is when we morphologically disambiguate a word that we start to consider its context, and even more clearly, when we reach the step of the syntactic analysis, which analyzes the relations between sintagmatic components.

There are various approaches to morphosyntactic disambiguation and to syntactic analysis, and variations depend on the criterion that is selected in each case. These criteria include the type of information we take as basis, or the result that we may want to obtain. In short, differences among various approaches depend on the viewpoint that is selected for approaching syntax. For our purposes, and in order to understand the formalism we have selected for the case of Basque, we will only mention the most prominent trends and their features.[3]

There are three prominent tendencies in disambiguating-grammars[4] and in syntactic analyzers: those based on linguistic descriptions, those based on statistical techniques, and finally, hybrid methods, which combine both.

2.1. Linguistic descriptions as basis

A. In disambiguating-grammars: these grammars are based on rules that are created by using linguistic knowledge, and are called knowledge-driven taggers. It is commonplace to create the rules manually, rendering them both very precise and costly. The predecessor for this tagging system is the tagger called TAGGIT created to tag the Brown Corpus in the 70's (Greene & Rubin 1971). The most widespread successor of this system, namely Karlsson's *Constraint Grammar* (henceforth CG), is the grammar we selected for morphosyntactic disambiguation in Basque. Based on this formalism, the group directed by Karlsson created the EngCG, namely the disambiguating-grammar for English. A more detailed presentation of this grammar is included later in this paper.

B. In syntactic analyzers: these are based on theories of grammar, such as *Lexical Functional Grammar* (LFG), *Generalized Structure Grammar* (GPSG), *Head Phrase Structure Grammar* (HPSG), *Government and Binding* (GB), etc. They mainly focus on

---

[3] See (Ezeiza 2002), a dissertation including further information on this topic.

[4] Disambiguating-grammars are also sometimes called *Taggers.*

sentences that are interesting from a linguistic viewpoint, rather than on real texts. However, parsers based on such descriptions typically fail when faced with texts in newspapers and technical texts. A further arising problem, due to the ambiguity, is that, for those sentences that they recognize, they provide several alternative interpretations and do not decide on the correct one. This is the reason why they are known to be of limited use in NLP. However, there are some applications that are based on theories of grammar that have targeted real texts. One example is the *Xerox Linguistic Environment* (XLE) (Kaplan & Newman 1997), which eases the creation of wide-coverage grammars based on LFG and obtains lexical and morphological information from external sources.

2.2. Probability-based techniques

During the last decade, approaches based on statistics have become increasingly common in taggers as well as in analyzers. They are based on empirical evidence that is retrieved by automatically analyzing large corpora.[5] However, it is not the case that they do not involve any basic linguistic knowledge. Rather, manual work on creating grammars is minimized to the limit, and linguistic knowledge from tagged corpora is retrieved by probabilistic means.

Most probability-based systems make shallow analyses by following this strategy. Taggers are one example, whose aim is to assign the syntactic category that fits to each word. Several statistical methods that involve various degrees of complexity have been employed to achieve this aim. The most simple ones (which use bigrams),[6] or those displaying greater difficulty, such as the decision-trees in (Màrquez 1999) (taggers employing machine learning systems), and the memory-based learning in (Daelemans et al. 1996).

Overall, the use of purely statistical methods has encountered difficulties in treating phenomena that appear outside the domain of limited texts. Thus, by looking at the results obtained, we conclude that such taggers display limited successful performance. For instance Voutilainen (1994); Brill & Wu (1998) report a %95-97 success for several languages. Such percentages are unacceptable when we consider syntax, since they would imply the existence of one error per sentence in a great number of sentences.

2.3. Approaches combining probability and linguistic knowledge

These approaches combine probabilistic methods and linguistic knowledge with the aim of gathering their advantages. In general, linguists write the rules of grammar, but the application of the rules is typically based on statistical knowledge. This statistical knowledge is extracted from large tagged corpora. The following employ this system of work: *IBM/Lancaster Approach* (Black et al. 1993) and Padró (1997).

Also, the *CG* formalism by Karlsson would fit in this group, since, although it is linguistic in nature, it employs some (although little) statistical information in the

---

[5] Also known as *treebank* or *parser bank*.

[6] The probability of a tag is calculated by considering the tag of the surrounding word.

English version. Apart for tagging purposes, CG is also used to improve shallow syntactic analyzers with great success. In fact, it has been one of the most successful system in the market for the last years.

The next section is about the CG analyzer, which is our choice for the morphosyntactic disambiguator as well as for the shallow syntactic analyzer.

3. Constraint Grammar (CG) parser

The CG parser was created in the 80's by Fred Karlsson and colleagues at the University of Helsinki. Here is the list of its most important characteristics, as stated in Karlsson et al. (1995):[7]

- Goal: the most important goal of this analyzer is to reduce ambiguity, i.e., to decide on correct/adequate analyses among the possible interpretations of a form. Another aim is to provide the shallow syntactic analysis of a text that has been previously morphologically analyzed.
- The goal of grammars aimed at *Parsers* (CG being one of them) is not to indicate the (non)grammaticality of sentences, but rather to provide a solution for every analysis by dispensing with the biggest possible number of erroneous/inadequate interpretations. A solution is sought for every element that needs analyzing in the text. In this sense, we may qualify this grammar as robust.
- It is independent from languages and from programming codes.
- Grammars and lexicons are adaptable to particular types of texts.
- The basis of the grammar is composed of constraint rules. Yet, when rules cannot provide a solution, there is room for the use of elements that contain probabilistic features; this contributes to robustness in the grammar.
- The core and basis of the grammar is the morphological analysis and the lexicon.
- The task of the grammar is threefold:
  - morphosyntactic disambiguation related to context;
  - assignment of boundaries between clauses;
  - assignment of surface syntactic functions and their disambiguation.
- CG is restrictive in the sense that its goal is to reduce morphological and syntactic ambiguity, in other words, to discard analyses that do not correspond to particular contexts.
- Since ambiguity arises at the word-level, the object-unit of analysis is the word.
- Syntactic analysis assigns a function to each word: first, it will complete words lacking a syntactic function in the database, and next, it will engage in the task of syntactic disambiguation. Thus, it will also provide information about the existing relations between words. However, the analysis is shallow and linear in the sense that it does not directly establish any tree or hierarchical relation.
- The basis of the rules is composed of the information provided by analyses of grammars and corpora.

---

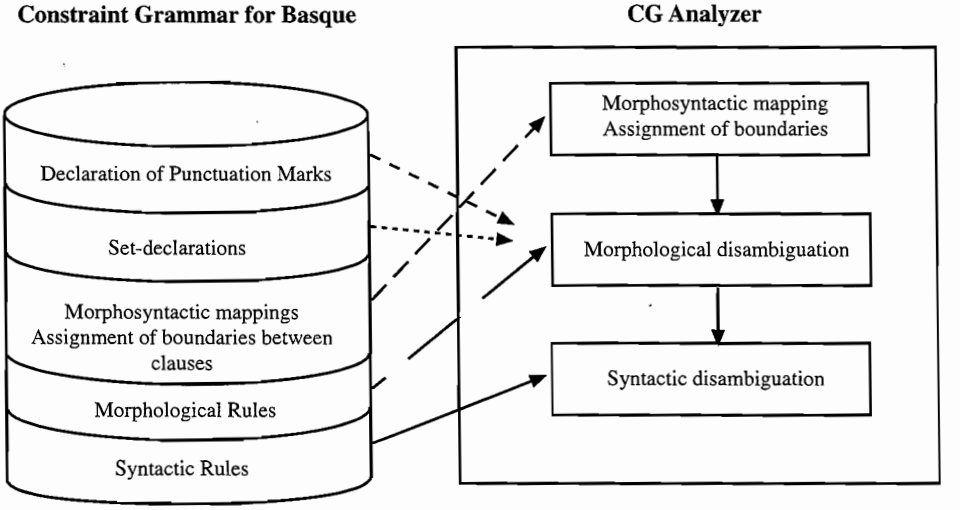
[7] The listed characteristics in the text strictly follow the scheme in the book.

— The amount of theoretical abstraction employed in the CGP is low compared to the rules in theories based on formal syntax, such as those that make use of the Government and Binding Theory or the Generalized Phrase Structural Grammar (GPSG).
— Rules are independent from each other.
— Pre-processing is important. The tasks of the pre-processor include tokenization, i.e., item-recognition,[8] assignment of boundaries to paragraphs, recognition of multi-word units, étc.
— There is room for using the standard SGML coding system in rules.

Constraint-grammars have been extensively employed in writing grammars for languages, although not all have been published. For English, we find EngCG (Karlsson et al. 1995), for Turkish Oflazer & Kuruöz (1994), for French (Chanod & Tapanainen 1995), for Finish, for Swedish, for Swahili, for Danish, for German, for Portuguese and for Spanish (Sánchez 1997), and for Basque (Aduriz 2000). The next section is a detailed presentation of the later.

4. Disambiguating-grammar for Basque: EUSMG

Considering the features described above, we chose the CG formalism for our purposes of starting to handle the syntax of Basque. The correctness of our choice has already been proven from a theoretical viewpoint. Let us next consider the application we have created for Basque. Although overall we have followed general principles, we have also been forced to make certain particular decisions.



IV.1. Architecture of the CG analyzer

[8] Tokenization involves token or item recognition. In other words, the isolation of the units that the morphological analyzer will use as input. These units may be elements such as lexical elements or marks of punctuation.

This grammar contains six parts, and the parser recognizes them all, even when they are empty of content (signaled with *NIL*). We will next present the details of the parts of the grammar, but first, consider the picture above in IV.1, which shows the general architecture of the grammar in a schematic guise.

4.1. Declaration of punctuation marks that delimit sentence boundaries

This will define periods, semicolons, question marks, etc.: DELIMITERS = "<$.>" "<$;>" "<$?>" "<$!>".

4.2. Set-declarations

Rules employ features of grammatical elements (including category, case, definiteness, etc.) in order to refer to the context of the word that is being treated. Often, it is possible to group elements that contain similar features. However, in order to use this information in rules, it is necessary to previously define the sets or groups. This section defines such sets. Consider the following examples:

[9] LIST ADLAG = "izan" "*edun" "*edin" "*ezan"

[10] LIST PERIFRASTIKOAK = ADOIN BURU EZBU GERO

[11] LIST DENB = DENB

4.3. Morphosyntactic mappings

The purpose of the mapping is to add information. The existing relations between morphological and syntactic features are typically expressed by mapping-rules. The mapping process assigns a syntactic function to each morphological interpretation. They are employed to assign syntactic functions that do not originate in the database. The grammar contains 83 mapping-rules.

Mapping-rules display the following shape:

<domain (operation), syntactic tag, the word *TARGET*, goal-interpretation, the word *IF*, contextual conditions >

— MAP (@-JADNAG) TARGET (ADI) IF (0 BURU) (1 ADL);

- Example: Basoan biziaz aberetu EGIN[12] dira

(lit. forest-loc. live-instr. beasten make have)
'By living in the forest, they have become beasts.'

---

[9] Set-declarations invariably start with the word LIST. ADLAG is the abbreviation for Auxiliary.

[10] The abbreviations for this set are the following: ADOIN "root of verb"; BURU "perfective"; EZBU "imperfective"; GERO "future tense".

[11] DENB "tensed".

[12] The word being treated will be written using capital letters.

Numbers signal the position of the element that we are working on (see more on this below, in the section that explains disambiguating rules). Numbers may either have a positive or negative value, which indicate the right or the left side respectively (for instance (1 ADL) refers to the right side). Position "0" refers to the element we are presently working on (e.g. (0 BURU)). Taking into account these clarifications, let us now consider how the mapping-rule above is paraphrased: *map the syntactic function @-JADNAG*[13] *to forms of the category verb (ADI), if the form itself is perfective* (0 BURU) *and if an Auxiliary* (1 *ADL*) *is placed to its right.*

4.4. Assignment of boundaries between clauses.

The assignment of clause boundaries is carried out by the mapping-rules mentioned above. When applicable, what is being assigned is the word MUGA. It recognizes both coordinate clauses that are attached by a conjunct as well as clauses attached by subordination. Let us consider an example:

— MAP (@MUGA) TARGET (KONP) IF (NOT 0 BAIT) (1C ADI-ADT OR ADPOSAG)[14]

- Example: Etorriko ZELA uste nuen
  Etorriko ZELA pentsatzen nuen

  (lit. 'Come-fut would-that thought I-did'),
  ('I thought that he/she would come.')

This rule states the following: *assign a boundary mark to completive subordinate clauses if they are immediately followed by a verb.* For more information on this, see Appendix C in Aduriz (2000).

4.5. Disambiguating Rules

Morphosyntactic and syntactic disambiguation is carried out by the same type of rules. These rules deal with both general and specific phenomena, and they contain the following domains:

<(domain) operation, goal-interpretation, the word *IF*, conditions of the word we are dealing with, contextual conditions >

The following example illustrates these domains:

— REMOVE (ADI) IF (0 ADJ) (NOT -2 DET)(-1 ZERO) (1 DET)

- Example: Bizitoki JAKIN bat ez zutela...

  (lit. 'home known one not had-they-that'),
  ('That they did not have a specific home.')

---

[13] -JADNAG "Non-finite main verb".

[14] Here is what the abbreviations in this rule stand for: KONP "completive"; ADI-ADT "verb and synthetic verb"; ADPOSAG "the component of periphrastic verbs built with the verb *izan*" (i.e., to be) (e.g.: *behar izan* "have to").

The rule can be paraphrased as follows:

> Delete the interpretation of the verb (ADI), if the form we are considering is also an adjective (0 ADJ), and if there is no Determiner in a two-word-distance to its left (NOT -2 DET); if, to a one-word-distance to its left there in an element with no morphemes (-1 ZERO), and to its right, to a one-word-distance, there is a determiner (1 DET).

Disambiguating rules can be classified into groups in terms for their degree of certainty in the correctness of the results that they provide. Particularly, in the grammar we have developed for Basque, we distinguish four groups of rules: the first group includes morphosyntactic rules that are most certain; the second contains morphosyntactic rules of less certainty; the third subsumes syntactic rules that are most certain, and finally, the fourth contains syntactic rules that are less certain, and more generally, idiosyncratic rules. One advantage of this classification is that it provides some order to the grammar, which would otherwise be hard to achieve in the presence of such amount of rules (Sánchez 1997).

The grammar for Basque contains 1.113 disambiguating rules: 672 in the first group; 45 in the second; 289 in the third, and finally, 107 in the fourth.

5. The bases and sources of the EUSMG

5.1. The Basque Lexical Database (i.e. *Euskararen Datu-Base Lexikala*: EDBL)

The lexicon, which is the core of the morphological processor, is organized in a database, namely in the EDBL (Alegria et al. 1997). When starting in an applied project of a real scale, it is necessary that linguistic data be structured and organized in a database. Although the EDBL was at first created to deal with morphology through a two-level formalism, it is currently the general Lexicon Database employed for treating Basque computationally. It is the basic necessary source of knowledge that is used in many aspects of NLP. This is the reason why it gathers various sorts of information: morphological and syntactic. Though it still does not include semantic information (namely, distinction among different meanings), the fact that it contains a homograph-identifier signals certain proximity to including semantic information (Agirre et al. 1994; Aduriz et al. 1998). Yet, the most important source of information is lexical information. The EDBL currently contains over ninety thousand entries, a meaningful number from our viewpoint, comparable to the amount employed for applications created for other languages.

As for the lexicographic-linguistic descriptions employed in building the EDBL, rather than following the demands of the computational applications, we have submitted to standard lexicography and to linguistic rules. The reason is that the Database is general, in the sense that it is designed for its use in several applications, and hence, we cannot restrict it according to the demands of specific applications.[15]

---

[15] In our case, ambiguity could have been reduced in the database itself. For instance, in the case of forms that are potentially ambiguous such as adjective/noun homographs, ambiguity can be reduced by assigning them a specific category in the database (as in the English version of CG, ENGCG: see Karlsson et al. (1995: 94-95).

The information corresponding to each entry is gathered in domains. Here are some: canonical form, two-level form (adapted to the model that will be employed in morphology), information regarding the morphemes that each category may take (continuation-classes), homograph-identifier, source, example(s), etc.

5.2. Corpora

The CG formalism belongs to approaches that are considered as empirical. Thus, one of its most prominent characteristics is that it is inclined to real corpora (Karlsson et al. 1995: 17) in two important respects. On the one hand, corpora, along with the grammar, are the source for linguistic information, especially in the process of creating disambiguating rules; on the other, they are the necessary locus for testing the degree of precision of applications and tools. Additionally, apart from the possible applications or uses mentioned above, corpora have become a necessary tool in statistical approaches. For instance, the tagging processes that employ the Markovian model and the model of Bayes employ corpora as recognition sources.

As for Basque, there are two corpora that are approved by the *Euskaltzaindia* or the Academy of Basque Language, and both are set within a project for lexical fixation:[16] the historical corpus called —*Orotariko Euskal Hiztegia* (OEH)— and the referential corpus that contains items of current use in Basque (named *"XX. mendeko euskararen corpus estatistikoa"*).[17] The later Urkia & Sagarna (1991; Urkia 2002) is a statistical corpus on written language, and it is balanced in terms of the types of texts.[18] This corpus is yearly updated, and currently contains 4.657.165 lemmatized words.[19]

We have employed parts of the XX century balanced corpus in our project. More specifically, we have made use of one part of the corpus in the process of rule-making and another when measuring the precision of the rules. There are few available Basque corpora (as is the case of all minority languages like Basque). In contrast, there are many resources for English, such as the Brown Corpus and the Penn Treebank. In fact, the size of the corpus is extremely relevant: in order to provide a thorough description of linguistic phenomena, it is necessary that the corpus be big. Thus, the more corpora available, the more thorough and precise are the researches based on them.

More information on corpora in Basque has been published by UZEI in the minutes from the meetings that have recently taken place, where they analyzed the current situation of corpora in the Basque Country (available from http://www.uzei.com).

---

[16] "Both corpora gather written documents, but they differ in one important respect: whereas the historical corpus collects complete works that are thoroughly stripped, the referential corpus is statistical in the sense that the focus of interest is lexical variety rather than the quality of works. It is called referential precisely because it reflects the current use of Basque" (Urkia & Sagarna 1991).

[17] The corpus of the XX century is being developed in UZEI. See *http://www.uzei.com* or *http://www.euskaracorpusa.net*

[18] "Corpora can be classified in a simple manner as being: balanced/non-balanced. In balanced corpora we find balance in the types of texts that they include, leaving aside particular characteristics pertaining to special texts. In order to achieve this, one must select many small, representative parts of texts from various sources by employing statistical techniques" (Alegria 1995).

[19] These are the available data as for the end of December 2002.

5.3. Morphological Analyzer

In section 3, where the basics of the CG formalism were discussed, we have mentioned that *the morphological analysis and the lexicon are its core and basis.* The morphological analysis of the EUSMG is provided by the analyzer, and the lexical is contained in the EDBL database. The previous section has presented the main characteristics of the EDBL database. Let us consider the features of the analyzer next.

In the beginnings of the computational treatment of Basque morphology, researchers searched for the model that would best describe the Basque features. There was more than one model available at the time (see Alegria 1995, Urkia 1997), and after several tests, we viewed the two-level morphology proposed by Koskenniemi (1983) as most suitable. Next, we will briefly mention the most relevant features of two-level models as stated in the above mentioned research by Alegria and Urkia: a) it is a general model, in the sense that it is applicable to any language, precisely because it distinguishes linguistic knowledge from algorithmic knowledge. b) it is applicable both to morphological analyses of words as well as for word generation; c) surface and deep lexical systems are clearly distinguished, which permits dispensing with the use of allomorphs; d) it employs parallel rules rather than the rewrite rules from generative phonology. This renders the system simpler both conceptually and computationally.

The core components of two-level formalisms are the lexical-system, morphotactics, and morphophonological rules. Other two characteristics of this analyzer are its robustness and its flexibility, which are the consequence of the following three improvements on the two-level morphology system by Koskenniemi. First, it left room for including the lexicon of the user; second, by employing the very same two-level model, non-standard forms corresponding to standard ones[20] were treated, with the aim of providing further robustness to the analyzer; finally, the analysis of unknown words was used for analyses of texts as well as for phonological analyses.

In fact, for each word that is recognized, the analyzer brings along all the information previously contained in the form of separate morphemes. In addition, often, it forms a set of ambiguous analyses, as in the following:

((**forma "bide"**)[21]
((**analisi 1**)
((lema "bide")((SAR bide)(KAT IZE)(AZP ARR))))
((**analisi 2**)
((lema "bide")((SAR bide)(KAT PRT)(MDL ZIU))))
((**analisi 3**)
((lema "bide")((SAR bide)(KAT IZE)(AZP ARR)))
((morf "0")((SAR 0)(KAT DEK)(KAS ABS)(MUG MG)(FS1 @OBJ)(FS2 @SUB)(FS3 @PRED))))

[20] The general description of Basque morphology constructed following the two-level model was done on standard Basque. The description of the non-standard Basque was done later.

[21] This is what the abbreviation in the analyses stands for: in the first analysis, SAR "entry"; KAT "category"; IZE "noun"; AZP "subcategory"; ARR "common". In the second analysis: PRT "particle"; MDL "mood"; ZIU "certain". In the third analysis: morf "morpheme"; DEK "declinative"; KAS "case"; ABS "absolutive"; MUG "definite"; MG "indefinite"; FS "syntactic function"; OBJ "object"; SUB "subject"; PRED "predicative".

Each possible analysis may include both morphological information (category, subcategory, case-definiteness, etc.), and syntactic information (syntactic functions). Some syntactic functions have already been defined in the database; however, others are assigned by mapping-rules, as we have just seen above when dealing with the grammar. Thus, the nature of the resulting ambiguity lies on the description of the language, in the sense that it arises as a consequence of the decisions made about this description and the criteria that have been followed. Hence, linguistic description necessarily conditions the result. This is the reason why it is worth being diligent when defining the analyzer and when building the lexicon for the database. Thus, the next step involves working on the results provided by the syntactic analyzer. This is the input-base for our work, namely the problem of ambiguity arising from analyzing texts.

6. Analyzing ambiguity

6.1. Delimiting the object of study

Ambiguity brings along disruption of communication in every aspect related to language. This is so, because more than one interpretation is available for each sentence, word, etc. Language is ambiguous in its nature.

> That ambiguity is natural to common language —to what we typically call bare language— in any of its various forms is such a well-known fact that there is no need to resort to refined dialectical and rhetorical techniques to convince the skeptical about it. (...). No doubt, ambiguity is one of the recurrent universals in natural language (...). (Michelena 1972).

The literature contains many references to this phenomenon, and the problem has been approached from various viewpoints. In fact, because of the extensive domain that it covers, it includes many types of alterations in language. Computationally, the problem of ambiguity is mostly related to *parsing* or to syntactic analysis. What must be first treated is the ambiguity arising from the information provided by morphological-morphosyntactic analyses.

Ambiguity is a persistent problem in linguistic analyses, but it becomes even a more serious and complicated trouble in computational analyses. In fact, these analyses process information (lexical and morphosyntactic) that has been gathered in the computer and provide results unlimitedly, often much unexpected ones. This is why ambiguity is one of the hardest problems in NLP, and especially in syntactic analyses.

Our object of study is, thus, the ambiguity arising in computational analyses. Before entering into details, let us mention that our study is set in shallow parsing. This conditions the domain of the object of study and the manner of dealing with ambiguity, which is far from any that has been proposed in theoretical approaches. Notice that we leave aside semantic, pragmatic and deep syntactic ambiguity (the later also called structural, as in *I saw the man with the telescope*). Rather, we study more local ambiguities, those concerning morphosyntax and syntactic functions. Recall that what are being considered are objects within blank spaces or words, and that disambiguation will be carried out by considering only their local context. Therefore, we deal with categorial disambiguation (e.g. the word *omen* may well be a particle, a common noun

or a verb in Basque), morphosyntactic ambiguity (the form *etxeak* 'the houses' may either be absolutive definite plural or ergative singular) and syntactic ambiguity or the ambiguity pertaining to syntactic functions (the word *etxeak* may either be subject, object or predicative).

It is obvious that such forms are not ambiguous in certain particular contexts. This is precisely the task of this grammar, i.e., to decide on the correct category, case or function of an object in a particular context. Keep in mind that forms are analyzed in isolation up to this point. However, in this step, where disambiguation is necessary, context becomes relevant.

6.2. Types of ambiguities

After presenting the types of ambiguities we are considering, we suggest distinguishing four types of morphosyntactic ambiguities: categorial ambiguity, ambiguity related to declension affixes, ambiguity in subordinating suffixes, and finally, ambiguity in aspect and mood-tense. We separate these types from the ambiguity arising in syntactic functions, which will be presented in a separate section below. With the purpose of getting information about the four types of ambiguities mentioned above, a first overall analysis of ambiguity was made by taking into account the most relevant linguistic phenomena. For this purpose, we used the morphological analyzer and the EDBL as sources, and based on these two we withdrew the first groups of ambiguity. Next, we made research on the corpus with the aim of accounting for the appearance of those groups of ambiguities. As a result, we evaluated the size of the problem and its actual relevance in each case. This contributed to robustness in the analysis.

Let us next consider in some detail the types of ambiguities, some illustrations of the problem, and the characteristics that they display:

1. *Categorial ambiguity:* categorial ambiguity is the most complicated and outstanding problem that we encounter after overcoming the level of the word. This has been widely attested in the literature that has been concerned with syntactic analyses and disambiguation (Karlsson et al. 1995, Padró 1997, Màrquez 1999). Within the typology of ambiguities in Basque, categorial ambiguity requires special consideration for the following reason: a word with a base category easily changes its function according to the context in which it appears. In these instances, it is necessary to resort to context in order to determine its category (or function)[22] (Euskaltzaindia 1993: 134). For present purposes, we will only mention the most clear cases, and for complete details see (Aduriz 2000). One clear case is the ambiguity between adjectives and adverbs. (e.g. *azkar* 'quicle, quiclely', *luze* 'long, lengthy'); others include the ambiguity arising in roots of verbs, adjectives and adverbs (e.g. *bizkor* 'hurry up, stimulate, vigorous, vigorously's'), or the one between auxiliaries and synthetic verbs (e.g. *da* 'be', *du* 'have').

[22] The fact that the terms category and function have recurrently been confused in Basque as well as in other languages has created many problems in creating a categorial system. On this problem, see further details in Huddleston & Pullum (2002), Zabala & Odriozola (1994), Aduriz (2000).

2. *Ambiguity in declension-affixes:* this group is also very productive in Basque. In fact, by virtue of being an agglutinative language, Basque employs morphemes on words where other languages employ syntactic structures. More specifically, Basque makes use of case-declension morphemes as well subordinating suffixes. Thus, ambiguity emerging from bound morphemes is extremely relevant in Basque. Among others, we find the following ambiguities in this group: absolutive definite plural and ergative definite singular markers are ambiguous; suffix *-ko* is ambiguous between locative-genitive, attributive and distributive.
3. *Ambiguity in subordinating suffix/prefixes:* this group includes ambiguities such as the one appearing in suffix *-(e)la* or in prefix *bait-*. More specifically, *-(e)la* may have the value of a complementizer or that of an adjunct (modal or temporal). As for prefix *bait-*, it can either express relative or causative meaning, it can form subordinate complements, or it can appear in consecutive clauses. It is important to note that the type of ambiguity in this section is closely related to verb subcategorization. In this sense, the more elaborate the subcategorization, the less ambiguity problems we will encounter.
4. *Ambiguity related to verbal aspect and mood-tense:* in this group, we find ambiguity arising in certain verbs between the verb-root and the participial form (e.g. *egon* 'be, stoy', *joan* 'go', etc.). Apart from the instances just illustrated, a big percentage of ambiguity in this group is due to ambiguity between synthetic verbs in the past (e.g. *nindoan* 'I was going', *zekarren* 'he was bringing') and the subordinating particle *-(e)n*.

6.3. Measuring Ambiguity

Next, we present the data about the percentages of appearance of the various types of morphosyntactic ambiguities that were described above. The results have been organized into two groups: first, we will present the data corresponding to categorial ambiguity, and next, we will include the rest of the information provided by the morphosyntactic analyzer in order to determine the results, i.e., we will include the three types of ambiguity that have been defined in the description (ambiguity related to free morphemes, subordinating elements and those related to verbs). In order to reach our goal, we have taken as basis a corpus that contains 14,000 text words. Because the corpus is real, apart from standard forms, we also find unknown words in the texts (those that are not included in the EDBL),[23] as well as variants to standard forms (dialectal forms, for instance) (Alegria 1995, Ezeiza 1997). The analyzer has provided a potential solution for every form through the analysis of standard forms, the variants and through the analysis of unknown words.

[23] "Half of the words that remain unanalyzed are not recognized because they are not identified in the corresponding lexicon (...). Although the reasons for the non-appearance of a word in the lexicon are diverse, (...) it is often impossible or extremely difficult to find all those lemmas in the general lexicon, for they are often the result of context or of the particular use of the author" (Alegria 1995).

	Categorial ambiguity	Complete morphosyntactic ambiguity
Standard forms	%46,34	%80,09
Variants	%32,89	%81,25
Unknown forms	%57,95	%95,88
Average	%37,80	%65,75

VI.1. Measurements of ambiguity

As the chart above shows, the number of possible interpretations of non-standard forms is larger than the one found in standard forms.

Categorial ambiguity includes 20 tags: 17 general tags and other three tags that serve to tag special cases such as ellipsis. This is the reason why the %37,80 of words in a text are ambiguous. In other words, rather than obtaining a single reading for each word, we find a bit more than one and a half per word. Moreover, ambiguity percentages double when we consider the overall morphosyntactic ambiguity, which reaches %65,75. In other terms, each form contains 2,81 interpretations. This is not surprising considering that all the information of the analyzer is located here: case, number, definiteness, mood and tense in the case of verbs, etc.

This ambiguity percentage differences between categorial ambiguity and the complete morphosyntactic ambiguity are not exclusively attested in Basque. Overall, for logical reasons, it is frequent in agglutinative languages: since these languages contain a rich morphology, this type of ambiguity that is not related to categories is much more persistent compared to languages such as English or Spanish. That morphosyntactic ambiguity is a serious problem is more obviously exemplified by Basque compared to the data in other languages (Karlsson et al. 1995: 23). For example, in Finnish, it reaches %11,2. In Swedish, as in Hebrew it is larger, %60 in both. Spanish displays around %43, and English %35.[24]

7. Morphosyntactic disambiguating rules

In everyday spoken language, the speaker (and the hearer) has available resources such as accent and context to solve ambiguity problems. Morphosyntactic disambiguating rules are to replace the resources that we mechanically employ in spoken language.[25] What do we understand by the term *disambiguation*? The answer is the following: to choose one out of various possible ways of understanding a form in a given context.

---

[24] However, it is difficult to make ambiguity comparisons between Basque and other languages for two reasons: first, because the basic tags employed vary, and second, because the base-texts are also different in nature. In order to obtain comparable results, both base-texts and the tagging-system should ideally be similar (Màrquez 1999).

[25] The previous section has restricted the domain of ambiguity we have treated and the types of ambiguities we have considered.

> In Constraint Grammar, disambiguation does not mean "bring out all alternatives" but rather "pick the appropriate alternative(s) by discarding one or more inappropriate ones". The Constraint Grammar notion of morphological disambiguation is functionally similar to the notion "homograph separation" (...) (Karlsson et al. 1995).

Disambiguating rules came into existence with the aim of fixing ambiguity problems that had previously been detected. A 14.000 word corpus was used for creating the rules. The rules are tested once and again, the errors are fixed, and information added; in other words, rules are refined until texts are correctly disambiguated. Since constraint-rules are a consequence of grammar-rules, in each case of ambiguity, we will derive principles similar to the grammar-rules that are derived from sets of rules.

The next section includes an illustration of this, namely, an example of rules and principles pertaining to morphosyntactic ambiguity. In fact, we will not thoroughly explain the grammar itself, since we have already explained in detail the design of the grammar where disambiguating rules are organized as well as the details of the syntax of the rules.[26]

7.1. Rules and Principles

The grammar contains 1.113 rules, and we have written the theoretical principles based on these rules. Mostly, rules are general, in the sense that they may refer to a whole group of ambiguities. However, in some cases, rules must be particular and they may only refer to specific words in a group of ambiguities. We will present disambiguating rules with an example: we will first explain a general principle, and next, we will show one of the rules that corresponds to this theoretical principle. We will provide the example[27] together with the rule.

Let us consider one of the examples of ambiguity that was mentioned earlier, in section 6.2: the form *bizkor* may be either the root of the verb-form *bizkortu*, an adjective, or an adverb. The following is the theoretical principle that deals with this issue:

> In periphrastic forms, roots of verbs take auxiliaries of the form **edin* or **ezan*[28] (ADL1) either to their left or right depending on the sentence-type. Participles, instead, are accompanied by either the auxiliary izan or **edun.*

One of the rules corresponding to this principle is the following:

— SELECT (ADOIN) IF (0 ADJ-ADB) (1C ADL1);[29]

---

[26] A detailed explanation of the grammar is available in the report (Aduriz et al. 2000, 2003).

[27] The examples of the rules that are provided in the text are mostly taken from real corpora.

[28] **edin* and **ezan* are roots of auxiliaries that express possibility, subjunctivity and imperative meanings. Instead, in the indicative, we use *izan* and **edun.*

[29] The abbreviations used in the rules: ADOIN "root of verb"; ADJ "adjective"; ADB "adverb"; ADL1 "set that includes **edin* and **ezan*".

- Example: azkar BIZKOR zaitezen

  Lit. 'Fast hurry-up you-subj.'
  'That you may hurry-up quickly.'

The application of this rule on ambiguous forms like *bizkor* gives as a result the choice corresponding to the root of the verb and discards the adjectival and adverbial interpretive options. However, a single rule does not always provide the correct conclusive analysis. Complete disambiguation is achieved through the intersection of the set of rules that consider context. There is the possibility that a certain context is not defined or that some casuistry has not been taken into account. If this is the case, the word will remain ambiguous. This may provoke an error in choosing the correct option among the possibilities by ruling out the correct reading. Let us consider how we have measured all this casuistry.

7.2. The results

Chart VII.1 shows the results of categorial disambiguation.[30] The results have been obtained by using a corpus of 10.000 words that has not been previously used in testing and rule-making:[31]

	Analyses per word	Ambiguity	Correctly interpreted words
Input	1,50	%37,80	%100,00
Output	1,18	%14,12	%99,12

VII.1. Results of categorial disambiguation

Compare this with Table VII.2 below, which displays the data resulting from the complete morphosyntactic disambiguation in the same corpus:

	Analyses per word	Ambiguity	Correctly interpreted words
Input	2,81	%65,75	%100,00
Output	1,76	%33,28	%97,51

VII.2. Results of the complete morphosyntactic disambiguation

[30] The data, wieh were used in Aduriz (2000), correspond to the year 2000.

[31] It has been necessary to manually disambiguate part of the corpus that has been used to calculate the results of the grammar and measure its efficiency. This task was also carried out within the IXA group while creating the rules.

The data show the robustness and power of the disambiguating-grammar when treating real texts. The amount of morphosyntactic ambiguity has dropped down to half: from %65,75 in the input to %33,28. In other words, in the input, from 2,81 possible analyses for each word, it has dropped to 1,7. In this disambiguating process, the correct interpretations have maintained at %97,51. Categorial disambiguation shows results that are even more successful. From 1,50 possible analyses per word, we get almost only one (1,18). In terms of percentages, ambiguity in the input has dropped from %37,80 to %14,12 in the output. Moreover, correct interpretations are maintained at %99,12.

In our view, the amount of errors found in categorial ambiguity is totally acceptable (0,8). It is more serious if we consider the whole morphosyntactic ambiguity. In fact, we know that the existence of non-standard forms considerably raises ambiguity percentages, and that the fact that the grammar is written for the standard mode often provokes errors. In addition, the percentage pertaining to non-disambiguated forms is mostly due to absence of information (the topic of subcategorization). To this, we need to add the cases that are impossible to disambiguate morphosyntactically due to their semantic or pragmatic nature. Also, remember that input ambiguity depends on the linguistic descriptions that have been made. Since the basis for this task, i.e., the database, is constantly being updated, the ambiguity that is added and the efforts of disambiguation do not go in parallel. Thus, the fact that we work on texts, descriptions and real data constantly requires updating. All this clearly shows the cyclicity in our work, as well as the distance from data that we sometimes need to set.

8. First steps in the syntactic analysis: shallow parsing (partial syntax)

Constraint Grammar works on a shallow or partial type of syntax. Thus, the main goal is to assign the corresponding syntactic function(s) to each word, and next, to disambiguate them (as we have just seen for the morphosyntactic level). This results in a shallow analysis of syntax.

As it was just mentioned, the first step is to assign all the possible syntactic function-tags[32] to every word-form. The procedure of assigning syntactic function-tags to words or morphemes is parallel to the assignment of morphological features to words. The next task is to reduce syntactic ambiguity. Syntactic ambiguity refers to situations where a word displays more than one syntactic function-tag (note that, for our purposes, ambiguity refers to syntactic functions rather than to structural ambiguity). Consider the following example in detail:

> Txakurrak (@OBJ @SUBJ)
>
> 'Dogs/the dog' (ambiguous between the object and the subject function respectively)

Again, as in the previous section, the main aim is to reduce ambiguity. For this purpose, we employ the syntactic constraints that are based in context. The goal of

[32] Syntactic function-tags are labeled with @.

applying syntactic constraint-rules is to reduce the number of potential syntactic tags into one, namely the correct one. However, constraint-rules do not attempt at cases where syntactic ambiguity is irreducible. For example, in the sentence "**Txakurrak** (@OBJ @SUBJ) **egunkariak** (@OBJ @SUBJ) ahoan zekartzan" (lit. '**dog-the papers** mouth-in-the was-bringing'), the words in boldface are left without disambiguating. Another option is that a decision would be made for isolating a single syntactic function. Recall that we do not consider structural or semantic-pragmatic ambiguities in this step.

8.1. Syntactic Functions

Syntactic functions-tags inform us about the function of words in sentences. They provide direct information about shallow parsing or shallow syntax, namely information about the surface structure of verb-chains. In fact, they provide information about the existing relations between words. Many of these functions do not correspond to the traditional functions; they are often tags that serve to form phrases or verb-chains. In fact, the grammar requires that every word must necessarily carry some tag. This is the reason why we sometimes find the traditional syntactic tags, and others, mere tags.

There are two main types of function-tags: heads and modifiers. *Main syntactic functions* correspond to the former, for instance, subject, object, indirect object, etc., and also to certain syntactic functions pertaining to verbs (+*JADNAG* e.g., matrix finite verb). In contrast, *modifiers* indicate the direction relative to their head: they express syntactic dependencies between elements within noun phrases or verbal periphrases. For example, adjectives that are followed by a noun that modify it are labeled with the tag @IZLG>, and the tag itself indicates that the head being modified is placed to its right (((*mendiko* (@IZLG>) *tontorretik* (@ADLG)),[33] literally "mountain-of top-from", meaning "from the top of the mountain").

Syntactic function-tags follow the philosophy of the CG formalism, in the sense that they are based on the *functionally labeled dependency syntax.*[34] By adopting the CG formalism, we express the syntactic functions of words and the interdependencies that exist among them.

8.2. Syntactic disambiguation

In the previous section, we have mentioned the importance of syntactic functions in shallow parsing in terms of the information that they provide about the existing syntactic relation among words. Thus, in cases where a reading contains more than one function, i.e., when it is ambiguous, we will need to disambiguate it.

The disambiguation of syntactic functions is carried out by syntactic rules, just as in morphosyntactic ambiguity. Thus, the aim of syntactic rules is to reduce the number of

---

[33] For further information on the later function-tags that we have employed, see the appendices in (Aduriz 2000).

[34] The concept of dependency-syntax has a long tradition in grammatical analyses since the Greco-Roman era. More recently, within the application of formalisms to syntactic theory, among others we find Tesnière (1959), Hays (1964) and Mel'čuk (1988), the ones who have recovered dependency-syntax in theoretical terms.

function-tags in each word-form into one. The set of rules, which includes both morphological and syntactic rules, are related to each other. In fact, syntactic constraint-rules are applied only after morphosyntactic disambiguation has taken place.

Let us consider one syntactic-ambiguity problem: absolutive forms in singular, plural and indefinites may either have subject, object or predicative functions. When faced with this ambiguity problem, several disambiguating rules have been created. One of them is the following:

— REMOVE (@OBJ) (0C ABS) (NOT *-1 NORK) (*-1 (NR_HU)) (1 (PUNT_PUNT));

- Example: Eta bertan agortu zen haren ODOL-JARIOA.

  Lit.: And there dry-up did his/her blood-flow
  "And there ended his/her bleeding."

The above rule can be paraphrased as follows: *delete the object function from the reading, if the word only contains absolutive case (0C ABS); if the sentence contains a verb of the* NOR[35] *paradigm (NR_HU), if it contains no auxiliary of the* NORK[36] *paradigm (NORK) and if there is a full stop to the right of the ambiguous form (PUNT_PUNT), i.e., if the sentence ends.* The example above suggests that agreement is of key importance to disambiguate syntactic functions, because it includes information on verbal subcategorization. This device will be particularly useful in disambiguating verbs including subordinating affixes.

9. Conclusions

We have developed a constraint-grammar for Basque in terms of shallow parsing with two aims: first, to obtain disambiguation of words that appear in real texts, and second, to develop the first steps in syntax by defining the existing surface relations between words. The main contributions of this grammar include a systematic analysis of ambiguities related to morphosyntax and to shallow parsing, as well as the specification of the disambiguating rules. However, the results of the syntactic analysis include no explicit phrasal-structure, since it does specify any hierarchies of components of phrasal nature.

Along these lines, several successful grammars that are capable of recognizing phases and verb-chains have been developed with great success (Arriola 2000, Aduriz et al. 2001). These grammars are based on the results of the analysis of CG, and mostly on syntactic functions, and they recognize the basic syntactic functions. The applications of these grammars would follow the grammar that we have presented in this paper.[37] Moreover, the results provided by CG have been the basis for other solutions in syntax, for example in the development of a PATR grammar, as presented in Gojenola (2000).

---

[35] Auxiliaries of the NOR paradigm arise when the verb subcategorizes for a single argument. This argument is marked Absolutive.

[36] Auxiliaries of the NORK paradigm arise in transitive environments, where the verb subcategorizes for a subject and an object, which will be marked Ergative and Absolutive respectively.

[37] These grammars have been created by using the mapping-rules of Constraint Grammar.

Also, the analysis (of grammar and phrasal disambiguation) will invariably serve as a basis and starting point for a deeper analysis. Thus, with the goal of achieving deeper analyses, the latest research on syntax in the IXA group is aimed at creating a corpus that is analyzed both syntactically and semantically. In order to do the syntactic annotation, we are currently working on a dependency-based grammar for Basque (Aduriz et al. 2002).

References

Aduriz, I., 2000, *EUSMG: Morfologiatik sintaxira Murriztapen Gramatika erabiliz.* Doctoral Dissertation, Basque Philology Section, University of the Basque Country.

—, 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.

—, —, Aranzabe, M., Arrieta, B., Arriola, J.M., Atutxa, A., Díaz de Ilarraza, A., Gojenola, K., Oronoz, M., Sarasola, K., 2002, "Construcción de un hábeas etiquetado sintácticamente para el euskara". *XVII Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural* (SEPLN). Revista n.º 29. Valladolid, España.

—, Arriola, J.M., Artola, X., Diaz de Ilarraza, A., Gojenola, K., Maritxalar, M., 2000, *Euskararako Murriztapen Gramatika: mapaketak, erregela morfosintaktikoak eta sintaktikoak.* UPV/ EHU/LSI/TR 12-2000.

—, —, Díaz de Ilarraza, A., 2003, "Desanbiguazio morfologikoa, azterketa sintaktikoaren lehen urratsak eta aplikazioak Murriztapen Gramatikaren eredu konputazionala jarraituz". In J.M. Makatzaga & B. Oyharçabal (eds.), *Euskal gramatikari eta literaturari buruzko ikerketak XXI. mendearen atarian. Gramatika gaiak,* Iker-14 (I), Euskaltzaindia, Bilbao: 3-35.

Agirre, E., Arregi, X., Arriola, J.M., Artola, X., Insausti, J.M., 1994, "Euskararen Datu-Base Lexikala (EDBL)". Report UPV/ EHU / LSI / TR8-94.

Alegria, I., 1995, *Euskal morfologiaren tratamendu automatikorako tresnak.* Doctoral Dissertation, Language and Computational Systems Section, University of the Basque Country.

—, Artola, X., Sarasola, K., 1997, "Hizkuntzaren tratamendu automatikoa", *Jakin* 102, 61-82.

Arriola, J.M., 2000, Euskal hiztegia-*ren azterketa eta egituratzea ezagutza lexikalaren eskuratze automatikoari begira.* Doctoral Dissertation, Basque Philology Section, University of the Basque Country.

Black, E., Garside, R., Leech, G., 1993, *Statistically-Driven Computer Grammars of English: The IBM / Lancaster Approach.* Black, Garside & Leech (eds.), Rodopi, Amsterdam.

Brill, E. & Wu, J., 1998, "Clasifier Combination for Improved Lexical Disambiguation". *COLING-ACL'98,* Montreal.

Chanod, J.P. & Tapanainen, P., 1995, "Tagging French – comparing a statistical and a constraint-based method". *Proceedings of EACL'95,* 149-156.

Daelemans, W., Zavrel, J., Berck, P., Gillis, S., 1996, "MBT: A Memory-Based Part-of-speech Tagger Generator". *Proceedings of the 4th Workshop on Very Large Corpora,* 14-27. Copenhagen.

Euskaltzaindia, 1993, *Euskal Gramatika Laburra: Perpaus Bakuna.* Euskaltzaindia, Bilbo.

Ezeiza, N., 1997, *EUSLEM, euskararako lematizatzaile/etiketatzaile baten diseinua eta inplementazioa.* Small Thesis Report, Language and Computational Systems Section, University of the Basque Country.

—, 2002, *Corpusak ustiatzeko tresna linguistikoak. Euskararen etiketatzaile morfosintaktiko sendo eta malgua.* Thesis report, Language and Computational Systems Section, University of the Basque Country.

Gojenola, K., 2000, *Euskararen sintaxi konputazionalerantz.* Doctoral Dissertation, Language and Computational Systems Section, University of the Basque Country.

Green, B. & Rubin, G., 1971, *Automatic Grammatical Tagging of English.* Providence: Brown University.

Hays, D.C., 1964, "Dependency theory: a formalism and some observations", *Language* 40, 511-25.

Huddleston, R. & Pullum, G., 2002, *The Cambridge Grammar of the English Language.* Cambridge University Press.

Kaplan, R.M. & Newman, P.S., 1997, "Lexical Resource Reconciliation in the Xerox Linguistic Enviroment". *Proc. of a Workshop on Computational Enviroments for Grammar Development and Linguistic Engineering,* Madrid.

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

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

Màrquez, L., 1999, *Part-of-Speech Tagging: A Machine Learning Approach based on Decision Trees.* Doctoral Dissertation, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

Mel'čuk, I.A., 1988, *Dependency Syntax: Theory and Practice.* State University of New York Press.

Michelena, L., 1972, "De la ambigüedad sintáctica", *Revista Española de Lingüística* 2, 237-247.

Oflazer, K. & Kuruöz, I., 1994, "Tagging and morphological disambiguation of Turkish Text". *Proceedings of ANLP-94,* 144-149.

Padró, L., 1997, *A hybrid environment for syntax-semantic tagging.* Doctoral Dissertation. Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

Sánchez, F., 1997, *Análisis morfosintáctico y desambiguación en castellano.* Doctoral Dissertation, Department of Linguistics, Modern Languages, Logic and Philosophy of Science. Universidad Autónoma de Madrid.

Tesnière, L., 1959/66, *Eléments de Syntaxe Structurale,* (2nd revised edition). Paris, Klincksieck.

Urkia, M., 1997, *Euskal morfologiaren tratamendu automatikorantz.* Doctoral Dissertation, Basque Philology Section, University of the Basque Country.

—, 2002, (forthcoming), "XX. mendeko euskararen corpus estatistikoa", in *Hizkuntza-corpusak. Oraina eta geroa.* Donostia.

— & Sagarna, A., 1991, "Terminología y lexicografía asistida por ordenador. La experiencia de UZEI". *Actas del VII congreso de la SPLN.* Donostia.

Voutilainen, A., 1994, *Designing a Parsing Grammar.* Publications of the Department of General Linguistics, 22. University of Helsinki.

Zabala, I. & Odriozola, J.C., 1994, "'Adjektiboen' eta 'adberbioen' arteko muga zehatzik eza", *ASJU* 28.2, 525-541.