Towards a Dependency Parser for Basque

M.J. Aranzabe, J.M. Arriola and A. Diaz de Ilarraza,
Ixa Group. (http://ixa.si.ehu.es)
Department of Computer Languages and Systems
University of the Basque Country
P.O. box 649, E-20080 Donostia
jibarurm@si.ehu.es

Abstract

We present the Dependency Parser, called Maxuxta, for the linguistic processing of Basque, which can serve as a representative of agglutinative languages that are also characterized by the free order of its constituents. The Dependency syntactic model is applied to establish the dependency-based grammatical relations between the components within the clause. Such a deep analysis is used to improve the output of the shallow parsing where syntactic structure ambiguity is not fully and explicitly resolved. Previous to the completion of the grammar for the dependency parsing, the design of the Dependency Structure-based Scheme had to be accomplished; we concentrated on issues that must be resolved by any practical system that uses such models. This scheme was used both to the manual tagging of the corpus and to develop the parser. The manually tagged corpus has been used to evaluate the accuracy of the parser. We have evaluated the application of the grammar to corpus, measuring the linking of the verb with its dependents, with satisfactory results.

1 Introduction

This article describes the steps given for the construction of a dependency syntactic parser for Basque (Maxuxta). Our dependency analyser follows the constraint-based approach advocated by Karlsson (Karlsson, 1995). It takes as input the information obtained in the shallow parsing process (Abney, 1997). The shallow syntax refers to POS tagging and the chunking rules which group sequences of categories into structures (chunks) to facilitate the dependency analysis. The dependency parser is considered as the module involved in deep parsing (see Fig. 1). In this approach, incomplete syntactic structures are produced and, thus, the process goes beyond shallow parsing to a deeper language analysis in an incremental fashion (Aduriz et al., 2004). This allows us to tackle unrestricted text parsing through descriptions that are organized in ordered modules, depending on the depth level of the analysis (see Fig. 1).

In agglutinative languages like Basque, it is difficult to separate morphology from syntax. That is why we consider morphosyntactic parsing for the first phase of the shallow syntactic analyser.

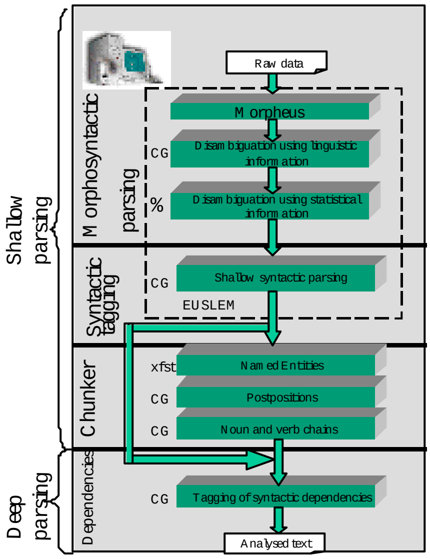

Fig. 1. Syntactic processing for Basque.

The dependency parser has been performed in order to improve the syntactic analysis achieved so far, in the sense that, apart from the surface structural properties, we have added information about deeper structures by expressing the relation between the head and the dependent in an explicit manner. Additionally, we have adopted solutions to overcome problems that have emerged in doing this analysis (such as discontinuous constituents, subordinate clauses, etc. This approach has been used in several projects (Järvinen & Tapanainen, 1998; Oflazer, 2003).

Before carrying out the definition of the grammar for the parser, we established the syntactic tagging system in linguistic terms. We simultaneously have applied it to build the treebank for Basque (Eus3LB[1]) (Aduriz et al., 2003) as well as to define the Dependency Grammar. The treebank would serve to evaluate and improve the dependency parser. This will enable us to check how robust our grammar is.

The dependency syntactic tagging system is based on the framework presented in Carroll et al., (1998, 1999): each sentence in the corpus is marked up with a set of grammatical relations (GRs), specifying the syntactic dependency which holds between each head and its dependent(s). However, there are certain differences: in our system, arguments that are not lexicalised may appear in grammatical relations (for example, the phonetically empty pro argument, which appears in the so-called pro-drop languages). The scheme is superficially similar to a syntactic dependency analysis in the style of Lin (1998). We annotate syntactically the Eus3LB corpus following the dependency-based formalism. The dependencies we have defined constitute a hierarchy (see Fig. 2) that describes the theoretically and empirically relevant dependency tags employed in the analysis of the basic syntactic structures of Basque.

This formalism is also used in the Prague Dependency Treebank for Czech (Hajic, 1998) and in NEGRA corpora for German (Brants et al., 2003) among others.

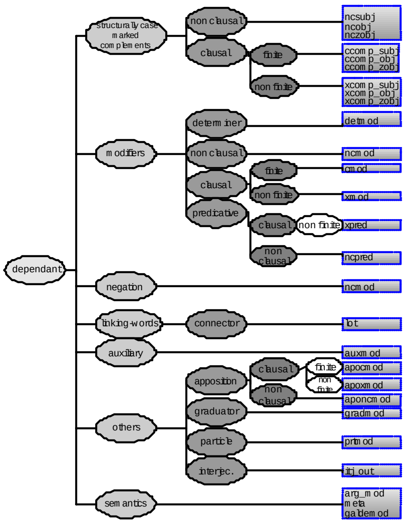

Fig. 2. Dependency relations hierarchy.

Section 2 examines the main features of the language involved in the analysis in terms of dependency relations. Taking into account these features, we will explain the reasons for choosing the dependency-based formalism. In section 3 we briefly describe the general parsing system. Section 4 explains the dependency relations, the implementation of the dependency rules and a preliminary evaluation. Finally, some conclusions and objectives for future work are presented.

2 A brief description of Basque in order to illustrate the adequacy of the adopted formalism

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

[1] This work is part of a general project (http://www.dlsi.ua.es/projectes/3lb) which objective is to build three linguistically annotated corpora with linguistic annotation at syntactic, semantic and pragmatic levels: Cat3LB (for Catalan), Cast3LB (for Spanish) (Civit & Martí, 2002) and Eus3LB (for Basque). The Catalan and the Spanish corpora include 100.000 words each, and the Basque Corpus 50.000 words.

Basque is its declension system with numerous cases, which differentiates it from languages spoken in the surrounding countries.

At sentence level, 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 have influenced us critically in our decision:

1. The dependency-based formalism is the one that could best deal with the free word order displayed by Basque syntax (Skut et al., 1997).
2. We consider that the computational tools developed so far in our group facilitate either achieving dependency relations or transforming from dependency-trees to other modes of representation.
3. From our viewpoint, it is less messy to evaluate the relation between the elements that compose a sentence rather than the relation of elements included in parenthesis.
4. Dependency-based formalism provides a way of expressing semantic relations.

3 Overview of the Syntactic Processing of Basque: from shallow parsing to deep parsing

We face the creation of a robust syntactic analyser by implementing it in sequential rule layers. In most of the cases, these layers are realized in grammars defined by the Constraint Grammar formalism (Karlsson et al., 1995; Tapanainen & Voutilainen, 1994). Each analysis layer uses the output of the previous layer as its input and enriches it with further information. Rule layers are grouped into modules depending on the level of depth of their analysis. Modularity helps to maintain linguistic data and makes the system easily customisable or reusable.

Figure 1 shows the architecture of the system, for more details, see Aduriz et al., 2004. The shallow parsing of the text begins with the morphosyntactic analysis and ends delimiting noun and verb chains. Finally, the deep analysis phase establishes the dependency-based grammatical relations between the components within the clause.

The parsing system is based on finite state grammars. The Constraint Grammar (CG) formalism has been chosen in most cases because, on the one hand, it is suitable for treating unrestricted texts and, on the other hand, it provides a useful methodology and the tools to tackle morphosyntax as well as free order phrase components in a direct way.

A series of grammars are implemented within the module of the shallow parsing which aim:

1. To be useful for the disambiguation of grammatical categories, removing incorrect tags based on the context.
2. To assign and disambiguate partial syntactic functions.
3. To assign the corresponding tags to delimit verb and noun chains.

3.1 Shallow Syntactic Analyser

The shallow or partial parsing analyser produces minimal and incomplete syntactic structures. The output of the shallow parser, as stated earlier, is the main base for the dependency parser. The shallow syntactic analyser includes the following modules:

1. The morphosyntactic analyser MORFEUS. The parsing process starts with the outcome of the morphosyntactic analyser MORFEUS (Alegria et al., 1996), which was created following a two-level morphology (Koskenniemi, 1983). It deals with the parsing of all the lexical units of a text, both simple words and multiword units as a Complex Lexical Unit (CLU).
2. The morphosyntactic disambiguation module EUSLEM. From the obtained results, grammatical categories and lemmas are disambiguated. Once morphosyntactic disambiguation has been performed, this module assigns a single syntactic function to each word.

3. The ckunk analysis module ZATIAK. This module identifies verb and noun chains based on the information about syntactic functions provided by each word-form. Entity names and postpositional phrases are also determined.

We will focus on the last step of the shallow analysis because it contains the more appropriate information to make explicit the dependency relations. Basically, we use the syntactic functions and the chunks that are determined in the partial analysis.

Shallow syntactic functions

The syntactic functions that are determined in the partial analysis are based on those given in Aduriz et al, 2000. The syntactic functions employed basically follow the same approach to syntactic tags found in ENGCG (Voutilainen et al., 1992), although some decisions and a few changes were necessary. There are three types of syntactic functions:

1. Those that represent the dependencies within noun chains (@CM>, @NC> etc.).
2. Non-dependent or main syntactic functions (@SUBJ, @OBJ, etc.).
3. Syntactic functions of the components of verb chains (@-FMAINV, @+FMAINV, etc.).

The distinction of these three groups is essential when designing the rules that assign the function tags for verb and noun chains detection.

Chunker: verb chain and noun chains

After the morphological analysis and the disambiguation are performed (see Figure 1), we have the corpus syntactically analysed following the CG syntax. In this syntactic representation there are not phrase units. But on the basis of this representation, the identification of various kinds of phrase units such as verb chains and noun chains is reasonably straightforward.

Verb chains

The identification of verb chains is based on both the verb function tags (@+FAUXV, @-FAUXV, @-FMAINV, @+FMAINV, etc.) and some particles (the negative particle, modal particles, etc.).

There are two types of verb chains: continuous and dispersed verb chains (the latter consisting of three components at most). The following function tags have been defined:

- %VCH: this tag is attached to a verb chain consisting of a single element.
- %INIT_VCH: this tag is attached to the initial element of a complex verb chain.
- %FIN_VCH: this tag is attached to the final element of a complex verb chain.

The tags used to mark-up dispersed verb chains are:

- %INIT_NCVCH: this tag is attached to the initial element of a non-continuous verb chain.
- %SEC_NCVCH: this tag is attached to the second element of a non-continuous verb chain.
- %FIN_NCVCH: this tag is attached to the final element of a non-continuous verb chain.

Noun chains

This module is based on the following assumption: any word having a modifier function tag has to be linked to some word or words with a main syntactic function tag. Moreover, a word with a main syntactic function tag can, by itself, constitute a phrase unit (for instance, noun phrases, adverbials and prepositional phrases). Taking into account this assumption, we recognise simple and coordinated noun chains, for which these three function tags have been established:

- %NCH: this tag is attached to words with main syntactic function tags that constitute a phrase unit by themselves
- %INIT_NCH: this tag is attached to the initial element of a phrase unit.
- %FIN_NCH: this tag is attached to the final element of a phrase unit.

Figure 3 shows part of the information obtained in the process of parsing the sentence Defentsako abokatuak desobedientzia zibilerako eskubidea aldarrikatu du epaiketan (The defense lawyer has claimed the right to civil disobedience in the trial) with its corresponding chains tags.

Let us know the some syntactic tags used in fig. 3: @NC>: noun complement; @CM>: modifier of the word carrying case in the noun chain; @-FMAINV: non finite main verb; @+FAUXV: finite auxiliary verb and @ADVL:adverbial.

```
"<Defentsako>"<INIT_CAP>"          defense
     "defentsa"N @NC> % INIT_NCH
"<abokatuak>"                      the lawyer
      "abokatu"N @SUBJ% FIN_NCH
"<desobedientzia>"                 disobedience
    "desobedientzia"N @CM> % INIT_NCH
"<zibilerako>"                     to civil
       "zibil"ADJ@<NC
"<eskubidea>"                      the right
       "eskubide"N @OBJ% FIN_NCH
"<aldarrikatu>"                    claimed
  "aldarrikatu"V @-FMAINV % INIT_VCH
"<du>"                             has
    "*edun"AUXV @+FAUXV% FIN_VCH
"<epaiketan>"                      in the trial
       "epaiketa"N @ADVL % NCH
  "<$.>"<PUNCT_PUNCT>"
```

Fig.3.Analysis of chains.English translation on the right

3.3 Deep Syntactic Analysis

The aim of the deep syntactic analysis is to make explicit the dependency relations between words or chunks. For this reason, we have designed a Dependency Grammar based on the Constraint Grammar Formalism.

4 The Dependency Grammar for the Parser

In this section we describe in more detail the dependency relations defined (see fig.2), the design of the rules and the results obtained. The results obtained in the deep parsing of sample sentence will help in providing a better understanding of the mentioned parsing process. This parsing process takes as basis the output of the shallow parser (see fig.3). The rules are implemented by means of the CG-2 parser (www.conexor.com).

4.1 The dependency relations

As Lin (2003) says a dependency relationship (Hays, 1964; Hudson, 1984; Mel'cuk, 1987; Bömová et al., 2003) is an asymmetric binary relationship between a word called head (or governor, parent), and another word called modifier (or dependent, daughter). Dependency grammars represent sentence structures as a set of dependency relationships. Normally the dependency relationships form a tree that connects all the words in a sentence. A word in the sentence may have several modifiers, but each word may modify at most one word. The root of the dependency tree does not modify any word. It is also called the head of the sentence.

For example, figure 4 describes the dependency structure of the example sentence.

We use a list of tuples to represent a dependency tree. Each tuple represents one relation in the dependency tree. For example, a structurally case-marked complement when complements are nc (non-clausal, Noun Phrases, henceforth NP) has the following format:

case: the case-mark by means of what the relation is established among the head and the modifier.

head: the modified word head of NP/dependent: the modifier. In this case, the head of the NP.

case-marked element within NP/dependent: the component of the dependent NP that carries the case.

subj relationship: the label assigned to the dependency relationship.

The syntactic dependencies between the components within the sentence are represented by tags starting with "&". The symbols ">" and "<" attached to each dependency-tag represent the direction in which we find the sentence component whose dependant is the target word.

In the example we can see that the noun phrase defentsako abokatuak 'the defense lawyer' depends on the verb aldarrikatu 'to claim', which is on its right side. A post-process will make this link explicit.

The dependency tree in fig 4 is represented by the following tuples:

Modifier	Cat	Head	Type
Defentsako	N	abokatuak	&NCMOD>
abokatuak	N	aldarrikatu	&NCSUBJ>
desobedientzia	N	eskubidea	&NCMOD>
zibilerako	ADJ	desobedientzia	&<NCMOD
eskubidea	N	aldarrikatu	&NCOBJ>
aldarrikatu	V		
du	Aux	aldarrikatu	&<AUXMOD
epaiketan	N	aldarrikatu	&<NCMOD

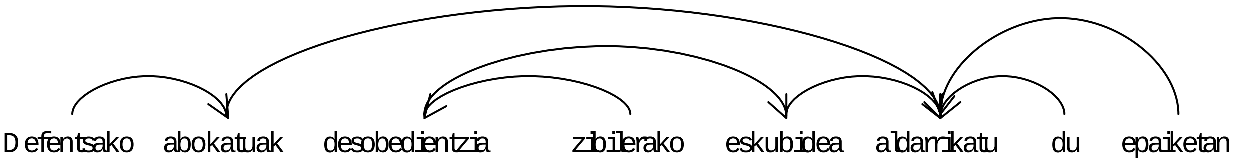

Fig.4. Dependency tree

42 The dependency grammar rules

The grammar consists of 255 rules that have been defined and distributed in the following way:

complements		modifiers			others
nc[2]	cc[3]	det	nc	cm[4]	
62	11	19	124	20	19

These rules were formulated, implemented, and tested using a part of the manually disambiguated corpus (24.000 words). For the moment, part of the rest of the corpus was used for testing.

For more details of the rules, we describe some examples that illustrate how dependency rules can be written to define different types of linguistic relations.

1. Verb-subject dependency

The following rule defines a verb-subject dependency relation between 2 words aldarrikatu (claimed) and abokatuak (lawyer) of the sentence in the previous example:

```
MAP (&NCSUBJ>) TARGET (NOUN)
  IF (0 (ERG) + (@SUBJ) + (%FIN_NCH))
    (*1 (@-FMAINV) + (%INIT_VCH)
     BARRIER (PUNCT_PUNCT));
```

The rule assigned the ncsubj tag to the noun abokatuak (lawyer) if the following conditions are satisfied: a) the noun is declined in ergative case; besides, it has assigned the @SUBJ syntactic function and, it is the last word of a noun chain; b) it has a non-finite main verb everywhere on its right before the punctuation mark.

2. Subordinate clause dependency

The following rule defines a complement subordinate clause dependency relation between a subordinate verb and a main verb. We illustrate this rule by means of an example in which the word egoten (usually stayed) is the verb of the complement subordinate clause linked to esan (told):

Example: Lehenago aitona egoten zela ni EGOTEN naizen tokian esan dit amonak[5].

```
MAP (&CCOMP>>) TARGET (V)
IF (0 (@-FMAINV) + (%INIT_VCH))
(1 (@+FAUXV_SUB) + (%FIN_VCH));
```

The rule assigned the CCOMP tag to the verb egoten (usually stayed) if the following conditions are satisfied: a) the verb is a non-finite main verb and, it's the first word-form of a verb chain; b) it has an auxiliary verb on its immediate right-side which has assigned the complement tag and appears as the last part of the verb chain.

3. Infinitive control

The following rule defines that in the sentence Jonek Miren etortzea nahi du. (John wants to come Mary), etortzea (infinitive subordinate clause with object function, "to come") is controled by the main verb nahi ("to want"). Taking into account, that etortzea is the controlled object of nahi, if there is another non-infinitive object Miren; then we will assign to it the subject dependency relation to the infinitive verb ("to come").

[2] nc: non-clausal complement or modifier

[3] cc: clausal complement

[4] cm: clausal modifier

[5] My grandmother told me my grandfather usually stayed where I am now

```
MAP (&NCSUBJ>) TARGET (NOUN)
IF (0 (ABS) + (@SUBJ) OR (@OBJ) + (%NCH))
  (1 (@-FMAINV_SUB_@OBJ)) (2 VTRANS_-FV));
```

4.3 Evaluation

The system has been manually tested on a corpus of newspaper articles (included in Eus3LB), containing 302 sentences (3266 words).

We have evaluated the precision (correctly selected dependent / number of dependant returned) and the recall (correctly selected dependent / actual dependent in the sentence) of the subject (including coordinated subjects), and modifier dependency of verbs. For subject, precision and recall were respectively 67% and 69%, while the figures for verb modifiers were 73% and 95%.

We have detected two main reasons for explaining these figures: 1) the analysis strategy is limited because we cannot make use of semantic or contextual information for resolving uncertainties at an early level; 2) errors in previous steps. These errors can be a) due either to an incorrect assignment of POS to word-forms or to the syncretism of case marks (@SUBJ, @OBJ); b) the presence of non-known word-forms that increases the number of possible analysis. At this moment, the head and dependent slot fillers are, in all cases, the base forms of single head words, so for example, 'multicomponent' heads, such as names, are reduced to a single word; thus the slot filler corresponding to Xabier Arzallus would be Arzallus.

5 Conclusions

We have presented the application of the dependency grammar parser for the processing of Basque, which can serve as a representative of agglutinative languages with free order of constituents.

We have shown how dependency grammar approach provides a good solution for deeper syntactic analysis, being at this moment the best alternative for morphologically complex languages.

We have also evaluated the application of the grammar to corpus, measuring the linking of the verb with its dependents, with satisfactory results. However, the development of a full dependency syntactic analyser is still a matter of research. For instance, all kinds of constructions without a clear syntactic head are difficult to analyse: ellipses, sentences without a verb (e.g., copula-less predicative), and coordination. All these aspects have been treated in our manually annotated Corpus; our efforts now are oriented to deal with them automatically.

6 Acnowledgments

This research is supported by the University of the Basque Country (9/UPV00141.226-14601/2002), the Ministry of Industry of the Basque Government (project XUXENG, OD02UN52).

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.

Aduriz I., Aranzabe M.J., Arriola J.M., Díaz de Ilarraza A., Gojenola K., Oronoz M., Uría L. 2004. A Cascaded Syntactic Analyser for Basque. In Gelbukh, A (ed.) Computational Linguistics and Intelligent Text Processing. Springer LNCS 2945.

Aduriz I., Aranzabe M.J., Arriola JM., Atutxa A., Díaz de Ilarraza A., Garmendia A., Oronoz M. 2003. Construction of a Basque Dependency Treebank. Proceedings of the Second Workshop on Treebanks and Linguistic Theories "TLT 2003", (J. Nivre and E. Hinrichs eds.), Växjö University Press. Växjö, Suecia

Aduriz I., Arriola JM., Artola X., Diaz de Illarraza A., Gojenola K., Maritxalar M. 2000. Euskararako Murriztapen Gramatika: mapaketak, erregela morfosintaktikoak eta sintaktikoak. UPV/EHU/LSI/TR 12-2000.

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.

Bömová, A., Haji,c, J., Hajicov´a, E., Hladk´a, B. 2003. The Prague DependencyTreebank: A Three level Annotation Scenario. In Abeillé (ed.) Treebanks Building and Using Parsed Corpora, Book Series: TEXT, SPEECH AND LANGUAGE TECHNOLOGY : Volume 20 Kluwer Academic Publisher, Dordrecht.

Brants T., Skut W. & Uszkoreit H. 2003 "Syntactic Annotation of a German Newspaper Corpus". In Abeillé (ed.) Treebanks Building and Using Parsed Corpora, Book Series: TEXT, SPEECH AND LANGUAGE TECHNOLOGY : Volume 20 Kluwer Academic Publisher, Dordrecht.

Carroll J., Briscoe E., Sanfilippo A. 1998. Parser evaluation: a survey and a new proposal. Proceedings of the 1st International Conference on Language Resources and Evaluation, 447-454. Granada, Spain.

Carroll J., Minnen G., Briscoe T. 1999. Corpus Annotation for Parser Evaluation. Proceedings of Workshop on Linguistically Interpretated Corpora, EACL´99. Bergen.

Civit M. & Martí M. 2002. Design Principles for a Spanish Treebank. Proceedings of The Treebank and Linguistic Theories (TLT2002). Sozopol, Bulgaria.

Hays, D. 1964. Dependency theory: a formalism and some observations. Language40, p. 511–525.

Hajic J. 1998. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In Issues of Valency and Meaning, 106-132. Karolinum, Praha.

Hudson, R. 1984. Word Grammar. Oxford, England: Basil Blackwell Publishers Limited.

Järvinen T. and Tapanainen P, 1998. Towards an implementable dependency grammar. In Proceedings of the Workshop "Processing of Dependency-Based Grammars", (eds.) Sylvain Kahane and Alain Polguère, Université de Montréal, Quebec, Canada, 15th August 1998, pp. 1-10.

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

Koskenniemi K 1983. Two-level Morphology: A general Computational Model for Word-Form Recognition and Production. University of Helsinki, Department of General Linguistics. Publications 11.

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.

Lin D. 1998. A Dependency-based Method for Evaluating Broad-Coverage Parsers. Natural Language Engineering.

Lin D. 2003. "Dependency-based evaluation of MINIPAR" in Building and Using syntactically annotated corpora, Abeillé, A. Ed. Kluwer, Dordrecht

Mel'cuk, I. A. 1987. Dependency syntax: theory and practice. Albany: State University of New York Press.

Oflazer K. 2003. Dependency Parsing with an Extended Finite-State Approach. ACL Journal of Computational Linguistics, Vol. 29, nº4.

Skut W., Krenn B., Brants T., Uszkoreit H. 1997. An Annotation Scheme for Free Word Order Languages. In Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97). Washington, DC, USA.

Tapanainen P. and Voutilainen A. 1994 Tagging Accurately-Don´t guess if you know. In Proceedings of the 4th Conference on Applied Natural Language Processing, Washington.

Voutilainen A., Heikkilä J. and Anttila A. 1992. Constraint Grammar of English. A Performance-Oriented Introduction. Publications of Department of General Linguistics, University of Helsinki, No. 21, Helsinki.