Abar-Hitz: An Annotation Tool for the Basque Dependency Treebank

Arantza Díaz de Ilarraza, Aitzpea Garmendia, Maite Oronoz

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
{jipdisaa, jibgamia, jiporanm}@si.ehu.es

Abstract

This paper presents the process followed to design and build a graphical and language independent tool, Abar-Hitz, for the creation and management of the Basque Dependency Treebank. Abar-Hitz makes the annotation process faster and avoids possible mistakes linguists can make. It is composed of three areas: the corpus area, the tagging area and the tree visualizer area. Three linguists used Abar-Hitz to tag 25.000 word-forms from the Eus3LB corpus, making clear, as the evaluation results show, its utility.

1. Introduction

This paper presents the process followed to design and build a tool, Abar-Hitz, for the creation and management of the Basque Dependency Treebank (Aduriz *et al.*, 2003). We think that the BDT is a necessary resource for the linguistic research in general and for the development of real applications in the area of NLP. This work is part of a general project[1] which objective is to build annotated corpora with linguistic annotation at syntactic, semantic, and pragmatic levels in three languages (Catalan, Spanish and Basque). Using the Abar-Hitz computational tool, the manual tagging for annotating the corpus will be easier and faster, and it will ensure the syntactic correctness of the written tags.

The corpus we annotated, Eus3LB, is a corpus of standard written Basque that contains 25.000 word-forms from EPEC (Aduriz *et al.*, 2003) and 25.000 words coming from newspapers that can be considered equivalent to the corpora in the other languages in the project. The tool has been used in the final state of the syntactic tagging process, and at the same time for correcting the manually tagged corpus developed in the meanwhile. In a near future, the tool will be massively used for tagging 300.000 word-forms.

Four linguists, who have contributed in the design of the tool, are annotating syntactically the corpus following the dependency-based formalism as explained in Aduriz *et al.*, (2002). In order to define the syntactic tagging system, we adopted the framework presented in Carroll *et al.* (1998, 1999). However, there are certain differences: in our system, arguments that are not lexicalized may appear in grammatical relations (for example, the phonetically empty *pro* argument which appears in the so-called pro-drop languages). It follows the EAGLES standards and it is based on the idea of adding to each sentence a series of grammatical relations specifying dependencies between modifiers and their nucleus. The tagset we have defined describes the most important grammatical structures such as relative clauses, coordination, discontinuous elements, eliptic elements and so on, and it has been deeply described in (Aduriz *et al.*, 2002). Besides, semantic tags are considered as a preprocess of the semantic annotation task, the next step in our work. Mistakes can be made while tagging in: i) the number of slots (e.g., putting 4 slots in a *ncsubj* relation which needs 5), ii) the type of each slot (e.g., putting a word in the first slot of the *ncsubj* instead of a case-mark) or iii) simply to misspell the name of the tag. Our annotation tool will guide the annotator in the process avoiding this type of errors. Abar-Hitz provides facilities for establishing the dependencies and visualizing the resulting tree for each sentence. The dependency tags have been declaratively expressed so the tool can be adapted to other tagsets.

Software quality characteristics defined in the ISO/IEC 9126-1 (2001) as functionality, reliability, usability, efficiency, maintainability, and portability have been taken into consideration.

2. The Abar-Hitz tool

Before designing Abar-Hitz, we analyzed some other annotation tools. WordFreak (Morton & LaCivita, 2003), a natural language annotation tool, can be used in different annotation tasks in English, Chinese, and Arabic, including: constituent parse structure and dependent, ACE named-entity, coreference, POS, NPCoref, token, sentence, and paragraph annotations. Our research group has been working several years in some of these areas as tokenization, morphosyntactic analysis, POS tagging, named-entity recognition, etc. and we already have a manually annotated corpus (EPEC), as well as automatic tools for this kind of analysis. The next step in our linguistic analysis chain (Aduriz *et al.*, 2004) is the development of a treebank. The annotation formalism we selected is not supported by WordFreak, so we decided to design and develop our own annotation tool, Abar-Hitz.

In the annotation process, it is interesting to have the possibility of displaying the analysis tree. We took this fact into consideration and added this functionality to Abar-Hitz after analyzing different tree editors: i) Treetrans, a tool for creating and manipulating syntactic trees that is part of the Annotation Graph Toolkit (AGTK) (Bird *et al.*, 2002) is based on constituents so, it was not appropriate for our purposes. ii) The graphical tree editor TrEd[2] is

[1] http://www.dlsi.ua.es/projectes/3lb

[2] http://ckl.mff.cuni.cz/pajas/tred/

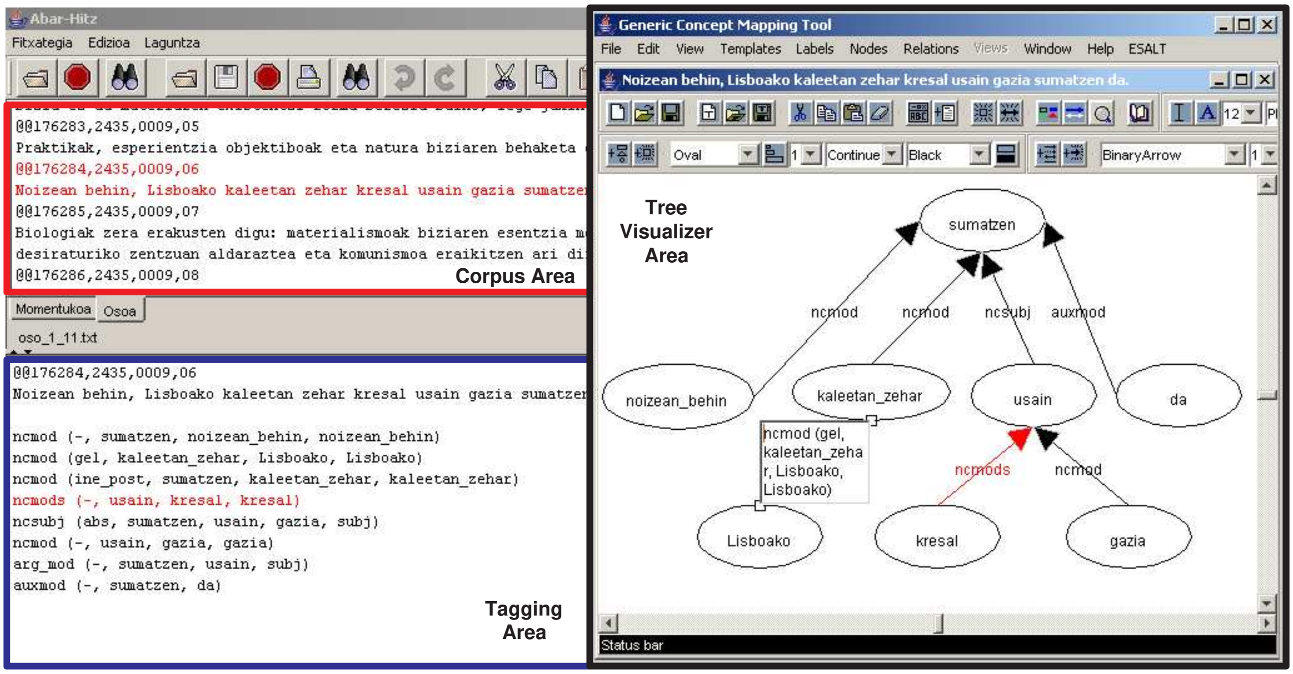

Figure 1: The three areas of the Abar-Hitz tool.

used in the Prague Dependency Treebank and it employs a dependency-based formalism. We were interested in establishing dependency tags as explicit relations between the head and the dependent nodes. The TrEd tool adds the dependency tag as an attribute of the dependent node. This idea does not fit to our needs. Moreover, TrEd does not provide us an appropriate way to represent both, syntactic and semantic dependency tags, in the way we proposed. iii) The Graph Tree Editor tool[3] similar to TrEd employs the same file format (FS) which is not appropriate for us due to the reasons previously explained. iv) There is another tool, TreeScape[4], that draws not editable trees. v) CM-ED (Arruarte *et al.*, 2001) is a concept map editor developed by the intelligent tutor group (GALAN) of our department that has been adapted for being a tree editor called ESALT. This tool follows a dependency-based formalism and it is the one that best fits our needs that is why we chose it as our tree visualizer.

2.1. Software quality characteristics

As we said before, some software quality characteristics defined in the ISO/IEC 9126-1 (2001) as functionality, reliability, usability, efficiency, maintainability, and portability have been taken into consideration.

When measuring the functionality of Abar-Hitz, the suitability that is the capability of the tool to provide an appropriate set of functions to the linguists, and the accuracy of the tool, have been proved in the evaluation process. In the near future we want to measure the interoperability of the tool with other systems.

Abar-hitz has maintained a good level of performance when it has been used for the development of the Basque treebank with a high reliability. Temporal files are created to recover the data in case of failure (recoverability).

As linguists have contributed in the development of Abar-Hitz, it has been designed and developed in a way that it is easy to understand, learn and use, becoming an attractive and usable tool.

Because some of the computers that the linguists use are quite old, the efficiency of Abar-Hitz had to be changed during its development process.

Abar-Hitz has been developed in Java and using modules with the idea of getting an easily maintainable and portable tool. It was tested under Microsoft Windows, Linux and Unix environments.

2.2. The interface

Abar-Hitz communicates with the user by means of a friendly interface. Figure 1 shows the three areas of Abar-Hitz, the corpus area, the tagging area, and the tree visualizer area, when tagging the sentence *Noizean behin, Lisboako kaleetan zehar kresal usain gazia sumatzen da* 'From time to time, the salty scent of seawater can be perceived in the streets of Lisbon'. The last two areas are edition areas with all the functionalities of text editors as cut, copy, paste, search, replace, undo, redo and print, while the text in the corpus area can not be changed.

2.2.1. The corpus area

In the corpus area, any text file can be opened. These corpus texts can be shown in two formats, i) the whole file, and ii) sets of three sentences, where the sentence to be tagged is marked in red. Abar-Hitz processes files in which sentences are tagged with a reference number identified by the '@@' symbol.

[3] http://quest.ms.mff.cuni.cz/pdt/Tools/Tree_Editors/Graph/
[4] http://www.cis.upenn.edu/ josephr/Trees/

2.2.2. The tagging area

There are two options in the tagging area: i) begin tagging a new sentence from raw text or, ii) revise an already annotated corpus.

In the first case, when the linguists start tagging a new dependency-relation in a sentence, an alphabetically ordered list of dependency-tags appear in a new window as can be shown in figure 2. The user can select the tag by typing the first letters of it. The values of the slots are chosen in a similar way. When all the slots are fulfilled, the system finishes the tagging. If the values of the slot are predefined (e.g. specific case-marks) the user can select the correct one from a list extracted from an XML document. When the value is a word, the dependency tag is completed extracting words from the sentence. This mechanism avoids mistakes and saves time to the user. As the description of each tag is stored in an XML file, the application is flexible enough to admit new tags or changes in the existing tagset, or even to define a new tagset in another language. In order to make the annotation process faster, extra effort has been put implementing the tagging area. We have given preference to the interaction using the keyboard instead of the mouse because it is faster.

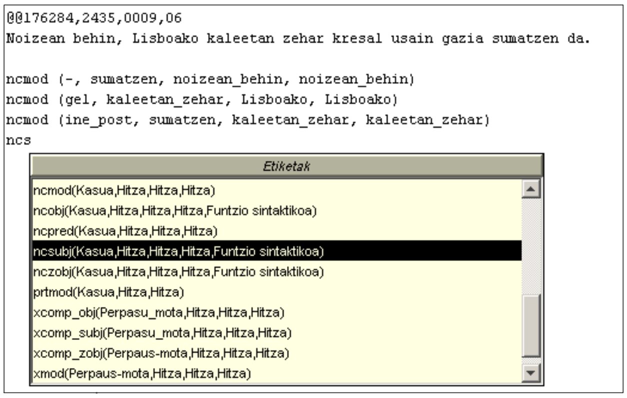

Figure 2: The tagging process.

In the second case, when an annotated sentence is opened, the correctness of the tags and their slots is automatically checked.

In both cases, the linguists have the option of verifying at any time the syntax of the tags by means of a button and the mistakes are marked in red. If the users later visualize the sentence-tree, they could check if all the nodes are attached to other, and if the tree has a unique root.

Remarkable characteristics of Abar-Hitz are that i) continuous and dispersed elements can be tagged, and ii) it is possible to have more than one analysis for the same sentence. In case of ambiguous sentences and when more than a sentence is loaded under the same reference number, an independent tree is visualized for each sentence analysis.

2.2.3. The tree visualizer area

ESALT, the tree visualizer, interprets the relation tags of the sentence and then, draws the tree on the corresponding window. Each node is tagged with a word or multiword, and each connector is tagged with the name of the relation. Before the tree is drawn, the correctness of each tag is verified using the XML file previously mentioned and if any tag is incorrect, an error warning appears in the screen and the connector of the relation is marked in red.

The drawn tree can be graphically manipulated so the user can change the tags and their fields, roll up subtrees, remove/add nodes, remove/add connectors (dependencies) and so on. The changes in the tree will be reflected in the tagging area, and the correctness of the new tagging will be automatically verified when its window is closed.

3. Evaluation

As we said before, some months ago three linguists of our research group started to analyze a corpus of 50.000 word-forms for the development of the Basque Dependency Treebank. As Abar-Hitz was not completely implemented, they tagged half of this corpus manually. As soon as the development was completed, they used massively the tool during two months to tag automatically the second half.

After finishing the analysis of the corpus, the manually tagged part was revised opening each sentence with Abar-Hitz and drawing its corresponding tree, with the idea of correcting mistakes. Table 1 shows the result of measuring the correctness of 181 already tagged sentences.

Sentences	Mistakes	Total	Percents
Wrong	Label	30	16,57 %
	Number of slots	12	6,63%
	Label + Number of slots	10	5,52%
	Total	52	28,73%
Correct		129	71,27%
Total		181	100,00%

Table 1: The evaluation results.

As can be seen in the table, three kinds of mistakes were taken into consideration, i) misspelling in the name of the dependency-tag, ii) erroneous number of slots in the tag, and iii) combination of the errors previously mentioned. When more than an error is detected in a sentence, a unique error is considered. These results give an idea of the utility of Abar-Hitz. %28,73 of the sentences have some kind of error and %22,09 of them are mistype errors that can be avoided when using the tool. Although mistakes in the word-forms of the corpus were not studied, linguists who check the correctness of the half manually tagged corpus, realize that there were quite frequent (e.g. mistakes in the words). The Abar-Hitz tool can prevent from these misspellings.

When tagging automatically the corpus, the linguists inform us about some needs. We take them into account and improve the tool in these aspects: i) new functionalities as search/replace, undo/redo and print were added, ii) speed when loading the drawn trees was increased so the efficiency of Abar-Hitz improved, iii) functionalities for tagging continuous and dispersed elements and comments were added.

4. Conclusions

The present article outlines the process of design and creation of a graphical and language independent annotation tool, that we have used for tagging the BDT. Abar-Hitz makes the annotation process faster and avoids possible mistakes linguists can make.

In the last years, our research group has been working on the integration of our NLP tools. EULIA (Artola *et al.*, 2004) is a tool which has been designed for dealing with the linguistic annotated corpora generated by the set of different linguistic processing tools. The objective of EULIA is to provide a flexible and extensible environment for creating, consulting, visualizing, and modifying documents generated by existing linguistic tools. The documents used as input and output of the different tools contain TEI-conformant (TEI-C, 1987) feature structures (FS) coded in XML. The tools integrated until now are a lexical database, a tokenizer, a wide-coverage morphosyntactic analyzer, a general purpose tagger/lemmatizer, and a shallow syntactic analyzer. We plan to use EULIA into Abar-Hitz, so the linguists could have all the information they want at any level when tagging the BDT.

Following the idea of integration of NLP tools, the format for the XML documents that will store the deep syntactic information has been defined. In the future, Abar-Hitz will produce these XML documents and they will be comparable to those that will produce the parser. In this way, the results of the automatic analysis at deep syntactic level will be evaluated.

5. Acknowledgments

This research is being supported by the University of the Basque Country (9/UPV00141.226-14601/2002), the Ministry of Industry of the Basque Government (XUXENG project, OD02UN52; CORPUSTR project S-PE03UN15), the Inter-ministerial Commission for Science and Technology of the Spanish Government (FIT-150500-2002-244), and the European Community (MEANING project, IST-2001-34460).

6. References

Aduriz I., Aldezabal I., Aranzabe M.J., Arrieta B., Arriola J.M., Atutxa A., Díaz de Ilarraza A., Gojenola K., Oronoz M., Sarasola K. (2002). Construcción de un corpus etiquetado sintácticamente para el euskera. Actas del XVIII Congreso de la SEPLN. Universidad de Valladolid, septiembre de 2002.

Aduriz I., Aranzabe M.J., Arriola J.M., Atutxa A., Díaz de Ilarraza A.,Ezeiza N., Gojenola K., Oronoz M., Soroa A., Urizar R. (2003). Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for the automatic processing. *Corpus Linguistics Around the World.* Book series: Language and Computers. Ed. Andrew Wilson, Paul Rayson, and Dawn Archer. Rodopi. Netherlands. Forthcoming.

Aduriz I., Aranzabe M.J., Arriola J.M., Atutxa A., Díaz de Ilarraza A.,Garmendia A., Oronoz M. (2003). Construction of a Basque Dependency Treebank. Second Workshop on Treebanks and Linguistic Theories (TLT2003). Vaxjo, Sweden. 14-15 November, 2003.

Aduriz I., Aranzabe M.J., Arriola J.M., Díaz de Ilarraza A., Gojenola K., Oronoz M., Uria L. (2004). A Cascaded Syntactic Analyser for Basque. Fifth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing2004). Seoul, Korea.

Arruarte A., Elorriaga J.A., Rueda U. (2001). A Templated Based Concept Mapping Tool for Computer-Aided Learning. Okamoto, R. Hartley, Kinshuk, J.P. Klus (Eds.), IEEE International Conference on Advanced Learning Technologies. IEEE Computer Society, pp. 309-312..

Artola X., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Sologaistoa A., Soroa A. (2004). EULIA: a graphical web interface for creating, browsing and editing linguistically annotated corpora. Workshop on XML-based richly annotated corpora in the fourth International Conference on Language Resources and Evaluation (LREC2004). Lisbon, Portugal. 29 May, 2004.(Submitted)

Bird S., Maeda K., Ma X., Lee H., Randall B., Zayat S. (2002) TreeTrans: Diverse Tools Built on The Annotation Graph Toolkit. Third International Conference on Language Resources and Evaluation (LREC2002). Las Palmas, Canary Islands, Spain. 29-31 May, 2002.

Carroll J., Minnen G., Briscoe T. (1999). Corpus Annotation for Parser Evaluation. Proceedings of Workshop on Linguistically Interpretated Corpora, EACL99. Bergen.

Morton T., LaCivita J. (2003) WordFreak: An Open Tool for Linguistic Annotation. Proceedings of HLT-NAACL 2003, Edmonton, May-June.

TEI-C (1987) Text Encoding Initiative Consortium. Text Encoding Initiative Website http://www.teic.org