INTEGRATING NLP TOOLS FOR BASQUE IN TEXT EDITORS

Ansa O., Arregi X., Arrieta B., Díaz de Ilarraza A., Ezeiza N., Fernandez I., Garmendia A., Gojenola K., *Laskurain B., Maritxalar M., *Martínez E., Oronoz M. (1), Otegi A., Sarasola K., Uria L.

Affiliation: IXA Group (http://ixa.si.ehu.es)
University of the Basque Country (UPV/EHU)
Postal address: Faculty of Computer Science
649 p.k., 20080 Donostia (The Basque Country)
Tel.: +34 943 01 5298
Fax: +34 943 219306
Email: jiporanm@si.ehu.es (1)

*Affiliation: Eleka S.L. Ingenieritza Linguistikoa
Postal address: Asteasuain 14
20.170 Usurbil (The Basque Country)
Tel.: +34 943 363040

ABSTRACT

In this paper we present the integration of several NLP tools in text editors. These tools have been developed following a strategy of five phases that we have designed for the processing of Basque. We are nowadays involved in the fourth phase of the mentioned strategy and have already developed and integrated three significant NLP tools —the spelling checker/corrector Xuxen, the Spanish/Basque Elhuyar Dictionary and the Synonym Dictionary—. Our current goal is the grammar checker/corrector, called Xuxeng, and we hope its first version will be integrated in text editors in a short time. From our experience, we know all this technology is relevant to make easier the use of written Basque as well as to help in the standardisation process of our language.

1.- Introduction

A language that seeks to survive in the modern information society requires language technology products. Human Language Technologies are making an essential contribution to the success of the information society, but most of the working applications are available only in English. Minority languages have to make a great effort to face up to this challenge (Petek, 2000) (13) (Williams *et al*., 2001) (14).

Language foundations and research are essential for the creation of tools or applications but, in the same way, tools and applications will be very helpful in the research and improvement of language foundations. Therefore, these three levels (language foundations, tools and applications) have to be incrementally developed in a parallel and coordinated way in order to get the best benefit from them.

Some years ago, we proposed a five phases strategy to follow in the processing of a language (Díaz de Ilarraza *et al*., 2003) (9). Although the strategy was designed for our language —Basque— it can be used to prove the adequacy of products to suit other languages as well, especially minority languages that suffer from the same kind of scarcity in the development of language technologies. These are the five phases of the mentioned strategy:

- *Initial phase:* Laying foundations (collection of raw texts with no tagging marks, lexical database as a simple list of lemmas and affixes, morphological descriptions…).
- *Second phase:* Basic tools (morphological analyser/generator, lemmatiser/tagger, corpus tagged with parts-of-speech and lemmas, lexical database with parts-of-speech and morphological information…).
- *Third phase:* Tools of medium complexity (environment for tool integration using XML, spelling checker and corrector, surface syntax, structured version of dictionaries, bilingual dictionary integrated

with a common text processor to be consulted on-line, lexical database enriched with multiword lexical units…).

- *Fourth phase:* Advanced tools (syntactically tagged corpus, grammar and style checkers, lexical semantic knowledge base —creation of a taxonomy of concepts such as WordNet—, language learning systems…).
- *Fifth phase:* Multilingualism and general applications (semantically tagged corpus after word senses have been disambiguated, information retrieval and extraction, translation aids, dialogue systems…).

As far as the automatic processing of Basque is concerned, some features of our language have to be known in order to evaluate the applicability of our strategy for other minority languages. Basque is an agglutinative language with a very rich morphology, and it has basically constituent-free order at sentence level. There are nowadays around 700,000 Basque speakers and they comprise about 25% of the total population of the Basque Country —although they are not evenly distributed. Despite there are six dialects, since 1968 the Academy of the Language (Euskaltzaindia) is involved in a process of standardisation of the language. At present, the morphology is completely standardised but the syntactic standardisation is still in progress. The spelling checker has proved to be a very useful tool to help the standardisation process of Basque. And we hope that the future construction of a grammar and style corrector will also contribute positively to this process.

According to this general strategy, this paper describes our work on the integration of NLP tools in text editors. Section 2 presents three significant tools that the IXA group has already developed and integrated in text editors —the spelling checker/corrector, the Spanish/Basque Dictionary and the Synonym Dictionary. The third section is focused on the grammar checker/corrector, called Xuxeng, we are working on at the moment. And finally, some conclusions and future work are outlined.

2.- Already developed and integrated tools

In the IXA Group we started working on the initial phase fifteen years ago and we are now working on the fourth phase. The spelling checker/corrector —called Xuxen—, a Basque-Spanish bilingual dictionary, and a synonym dictionary have already been integrated in Microsoft Office®.

2.1.- The spelling checker/corrector Xuxen

Xuxen —the spelling checker/corrector for Basque— is based on the formalism of two-level morphology (Aldezabal *et al.*, 1999) (4). As Basque is a highly inflected language, spelling checking and correction have been devised as a by-product for the morphological analyser/generator.

The lexical information needed by the morphological analyser/generator is stored in a general-purpose lexical database, EDBL —Euskararen Datu-Base Lexikala— (Aldezabal *et al.*, 2001) (5). Some lexicographers enrich the database (correcting, updating and adding new entries) with a browser-based user interface. Of course, not all the technical entries, person names, place names, etc. are stored in the database, but Xuxen provides the users the possibility to create their own user-lexicon with those words they commonly use. This is, in fact, one of the distinguishing features of this spelling checker/corrector. Unlike checkers of other main languages, Xuxen offers the users the possibility to enter new word-forms in their user-lexicon and it is able to recognise all the inflected forms of the entered words. Therefore, when a word is not known by the checker, it is assumed to be a misspelling, the user is given a warning and he/she has two different options:

a) to select among one of the possible correct proposals (if any).

b) to enter the new entry in the user-lexicon. As figure 1 shows, in the openned window the user enters the lemma and its parts-of-speech (noun, verb, person name…). Once the lemma is stored in the user-lexicon, the spelling checker Xuxen recognises it as well as all its possible inflected forms.

This utility we have implemented is of capital importance since Basque is very rich in morphology.

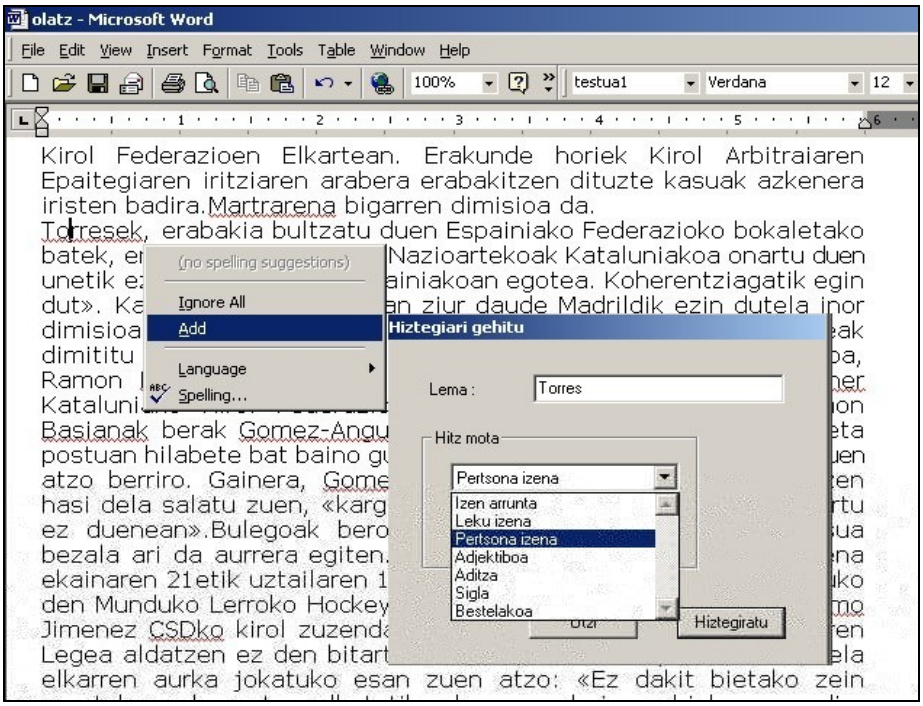

Figure 1: For the unrecognised word *Torresek*, the user clicks on the *Add* option to introduce the new entry's lemma (Torres) and its word class (person name) in the user-lexicon. Afterwards, the system will be able to recognize all its inflected forms.

From the user's point of view, Xuxen is a valid system to correct documents elaborated by some text processors. As it operates at a usual speed and takes up reasonable amount of space, it works well with any microcomputer. So far, Xuxen has been integrated in personal computers in Microsoft Office'97®, Microsoft Office'2000® and Microsoft Office XP® for Windows®, and in Office'2001® for Macintosh®. Apart from this, Xuxen can be used in Quark Express®, in Open Office1.0, and it is also possible to integrate it in intranets as well as to use it through the web. This spelling checker/corrector is widely used in the mentioned applications.

2.2.- The Spanish-Basque Elhuyar Dictionary

We have also implemented an on-line bilingual dictionary based on lemmatisation (Agirre *et al.*, 1999) (3). Due to the fact that Microsoft Word® did not incorporate bilingual dictionaries in the version we were working on, we decided to develop our own plug-in according to the style of the Word® text processor. This plug-in allows the user to consult the bilingual Spanish-Basque Elhuyar[1] dictionary —with 40,000 entries—while working with the Microsoft Word® 2000 text-processor. It contains 3 main modules: a lemmatiser for Spanish[2], a lemmatiser for Basque (Alegria *et al.*, 2003) (7), and a bilingual dictionary.

Part-of-speech (POS) labels as well as lexical units of both the dictionary and the two lemmatisers have been mapped in order to preserve a unified and correct treatment. Derived forms and multiword units have been also consistently treated. When the user selects a word-form in the text, all its possible lemmas and parts-of-speech are shown, as well as their corresponding equivalent in the other language.

The Graphical User Interface (GUI) we developed consists of a pop-up menu and a specialised window (see figure 2). The specialised window includes different options such as: i) the looking up of the text words, ii) the interactive sequential navigation in a sentence, iii) the direct insertion of translations in the text, and iv) the direct access to the web version of the bilingual dictionary. The resulting tool is therefore more than a pure consulting dictionary because it lemmatises the word-form we are looking for. For example, if we would like to know the meaning of the Basque word-form *orbainetan (in the scars)*, we would have to look up its entry *orbain (scar)* in the printed dictionary, and as the entry *orbain* is far from the context *orbainetan* we are interested in, the user could give up without finding the intended equivalent. Our dictionary system automatically offers the possible lemmas and all their corresponding equivalents. This plug-in was developed using Visual Basic.

[1] http://www.elhuyar.com/hiztegia/

[2] We have used the MACO system, developed by the TALP group from the UPC (Universitat Politècnica de Catalunya) and the CLIC (Centre de Llenguatge i Computació of Barcelona).

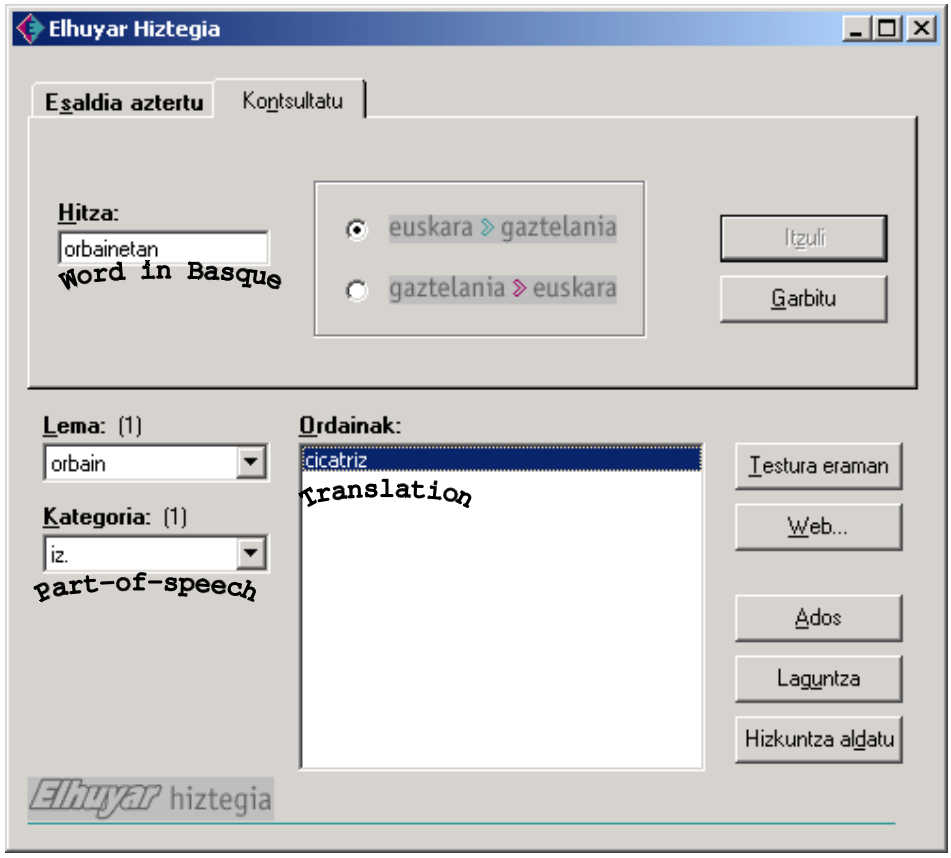

Figure 2: Specialised window of the bilingual dictionary integrated in Word®.

2.3.- The Synonym Dictionary

A synonym dictionary with 23,150 entries, created by UZEI[3], has been also integrated in Microsoft Word®. In this case, in order to integrate the dictionary we have not implemented an autonomous plug-in, but have used the Thesaurus API (*ctapi*) from Microsoft®. This decision carries important consequences. On the one hand, due to the fact that this API is used by Microsoft® to integrate synonym dictionaries in different languages, the Thesaurus API facilitates a global frame for the application and a normalised interface for the user. But on the other hand, it has limitations as far as design possibilities are concerned.

In the format of the Thesaurus API for Basque, there are some specific features we would like to mention. Firstly, in the printed dictionary there is a vast group of tags (20 tags for grammatical categories, 20 dialectal tags and 7 tags for usage features). This is an important information to the user and therefore, it is necessary to integrate it in the interface. Moreover, a word-form can have more than an entry (depending on its grammatical category), and at the same time, an entry can have more than one meaning. All this information has been adapted, as far as possible, to the output format used by the Thesaurus API.

Secondly, we have integrated a morphological analyser/generator, which is necessary for lemmatisation/generation, in the Thesaurus API. This way, in order to know the synonym of an inflected word-form, the API first lemmatises the given word-form and obtains its lemma, parts-of-speech and morphological information. It saves this information and looks up the lemma's synonyms in the dictionary. On the last step, the API uses the morphological generation process taking into account the word-form's morphological information previously saved. Thus, we obtain the synonym or synonyms of the consulted word-form already inflected. In case the given word-form is not in the dictionary, the most approximate forms will appear in the dialog box. Figure 3 shows the synonyms for the word *iritziaren* (*of the opinion*).

[3] An enterprise dedicated to linguistic processes and their difussion (http://www.uzei.com).

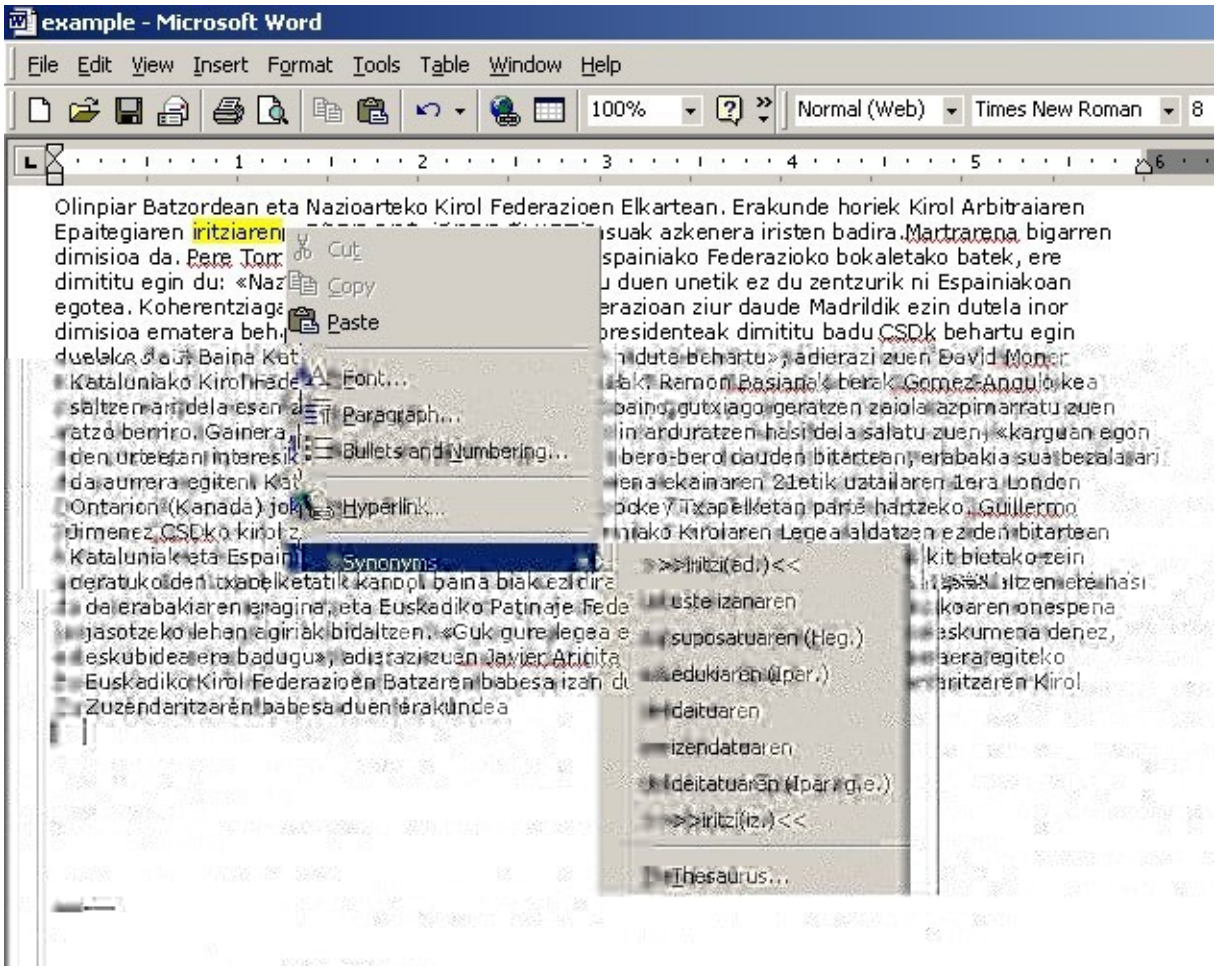

Figure 3: Synonyms of the inflected word-form *iritziaren* (*of the opinion*) in the Thesaurus Word for Basque. Being Basque an agglutinative language, it is important to have an application which shows the synonyms already inflected.

In addition, it is possible to look up the synonyms for all the words we have in the text and we can have access to the dictionary entries either directly from the text or from the dialog box. This application has been implemented in C language, and many potential users have already tested it satisfactorily.

3.- Towards the development of a grammar checker: Xuxeng

3.1.- Syntactic analysis

When starting the fourth phase of our strategy, and after having developed and commercialised the tools mentioned in section 2, the next step to face up is the creation of a grammar checker/corrector for Basque, which should be also integrated in different text-processors. In order to make possible this achievement, we have been working on syntax analysis for the last ten years. There are several formalisms to carry out the syntactic processing of a language and in the IXA group we have chosen two of them for the creation of syntactic analysers for Basque. The first one was developed using a unification-based formalism (Aldezabal *et al*., 2004) (6) and the second one was based on the Constraint Grammar formalism (Aduriz, 2000) (1). For the development of Xuxeng, we have chosen the second formalism, that is to say, the syntactic analysis chain used by Karlsson's Constraint Grammar (henceforth CG) (Karlsson *et al*., 1995) (11).

At present, we are working on the creation of a robust syntactic analyser by implementing it in sequential rule layers. In most of the cases, these rule layers are materialised in different grammars written in CG. Each analysis layer gets the output of the previous one and enriches it with further information. Figure 4 shows the architecture of the mentioned system.

The parsing process starts with the outcome of the morphosyntactic analyser, called MORFEUS (Alegria *et al*., 2003) (7), which was created following the two-level morphology and it deals with the parsing of all the lexical units of a text, both simple words and multiword units (CLU- Complex Lexical Unit). From the obtained

results, grammatical categories and lemmas are disambiguated. The disambiguation process is done by means of the linguistic rules of CG and the stochastic rules based on Markovian models (Ezeiza *et al*., 1998) (10) with the aim of improving the parsing tags in which the linguistic information obtained is not accurate enough.

Next, the shallow syntactic analysis is carried out using the tagger/lemmatiser, named EUSLEM. Afterwards, the system defines entity names and postpositional phrases. For the recognition and categorisation of entity names (person, organisation and location) we have created a combined system. Firstly, the system applies a grammar that has been developed using the finite state technology (FST), which detects the entity names from the morphological information. Then, entity names are classified through the application of a heuristic, which combines both textual information and gazetteers (Alegria *et al.*, 2003) (8).

Another characteristic of Basque different to other languages is its postpositional system. In this phase, our system recognises the postpositional phrases that consist of a case suffix and an independent word. For example: *itsasoari buruz ('about the sea') —itsaso(sea)+ari buruz(about the)—*. This type of postpositional phrase is taken into account in the recognition of noun chains.

The next layer identifies both verb and noun chains using CG rules. Our grammar recognises continuous and non-continuous verb chains and simple and coordinated noun chains.

The layers of the shallow parsing are already accomplished and at present we are working on the deep syntactic analysis. The aim of the deep syntactic analysis is to establish the dependency relations between the components of the sentence. This way we can detect more complex error-types. This process is also performed by means of CG rules.

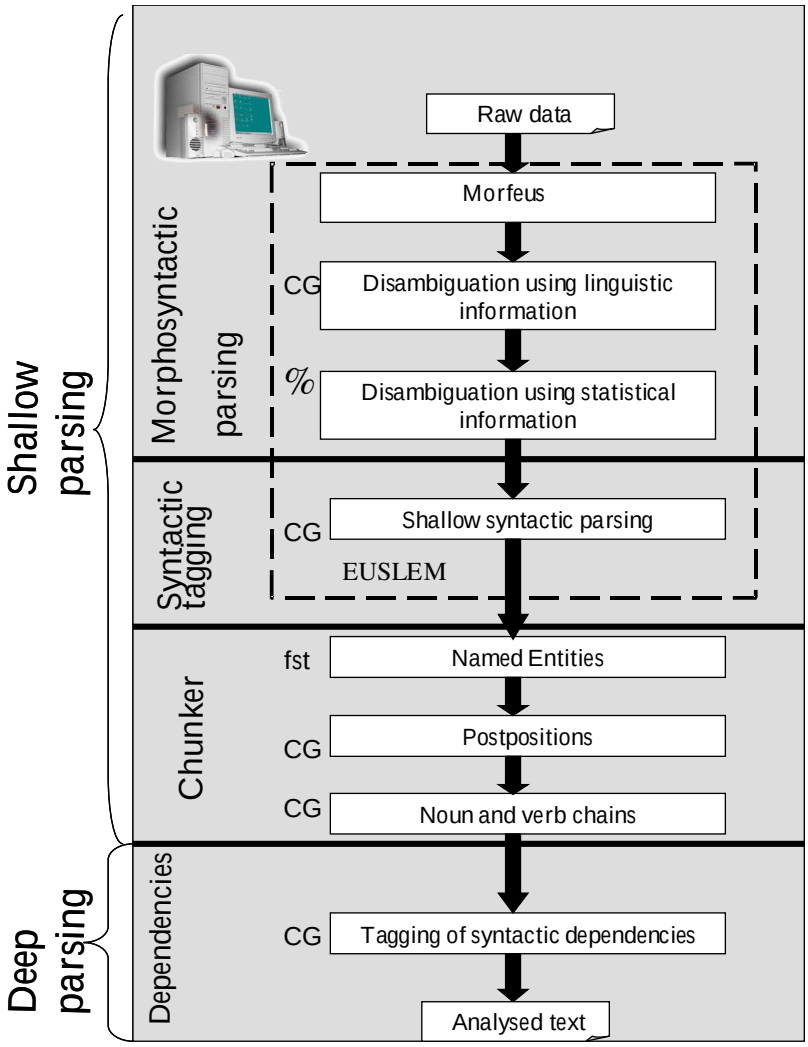

Figure 4: Progressive layers in the syntactic analysis.

3.2.- Error detection

Previous work has been done in the error detection field (Aduriz *et al.*, 2002) (2). Before starting with the definition of error detection rules, we prepared an environment for analysing errors. These are the steps followed in this process:

1. First, we carried out a classification of possible error-types divided into five main categories:
   - Spelling errors
   - Morphological, syntactic or morphosyntactic errors
   - Semantic errors
   - Punctuation errors
   - Errors due to the lack of standardisation of Basque

   Each category was subcategorised at the same time (in all, there are 55 subcategories in this application) in order to make as an accurate classification as possible.
2. Then, we validated and optimized this classification with the help of experienced Basque language teachers and proofreaders of newspapers or publishing houses.
3. After that, we designed and implemented a digital resource to be used as a repository of information of linguistic errors (Aduriz *et al.*, 2002) (2). This resource consists of a database, named ERREUS, and a Zope interface (Latteier *et al.*, 2001) (12) and it lets linguists and experts in this subject introduce, via Internet, any error found in a corpus (along with its corresponding information).
4. Once the classification was completed, the defined error-types were analysed to see whether they should be corrected, when and how. To do so, we divided the identified errors in three main groups, taking into account the information needed to detect them. We firstly grouped the errors that do not need any linguistic information to be detected —such as some punctuation errors or style proposals—, in the second group we classified those errors needing the result of the shallow parsing —data errors or postposition errors, for example—, and in the third group those errors that need the deep syntactic analysis —such as agreement errors.

We started working with the first group of errors to get the first prototype of the grammar checker. These errors seemed to be the easiest ones to treat since they do not need any linguistic information to be detected. For this purpose, we got the grammar API of Microsoft® (*cgapi.h*), which is implemented using Visual C. We also made some simple Visual Basic programs for an easy integration of the detection modules/programs in the API. However, this process was not as easy as expected because of the insufficient documentation about the API. As a consequence, we had to use the documentation of *csapi* (the Microsoft Common Speller API for Office Spell Checking) found in http://support.microsoft.com. This one is wider and provides useful information for the grammar API. We got some satisfactory results and at present we are able to underline the errors detected in a text as well as to give correction proposals.

Figure 5 shows an example of a style error. The message shown by the system is just a brief description of the error. But if the user wants further information (as to see both correct and incorrect examples), he/she can go to the help menu of the grammar window. All the information concerning the error (the error identifier tag, its category in the error classification, a brief and a wide description as well as correct and incorrect examples) is stored in an XML document. XML documents have been designed in a way to offer a direct link to the ERREUS data-base for future applications/utilities. We can consider this document the inventory of errors, and this is read from the API.

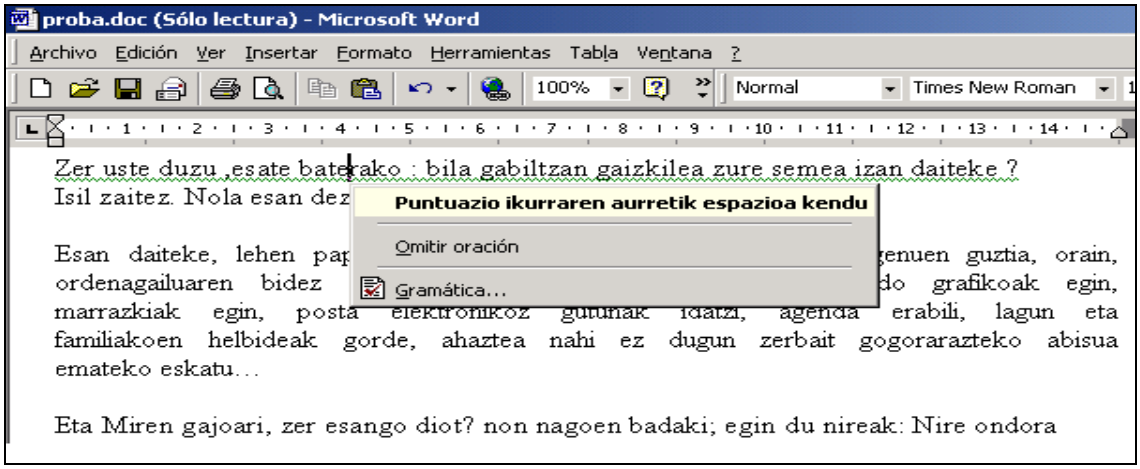

Figure 5: The message in the box warns that the space before the colon should be eliminated.

The second version of the style corrector uses the shallow syntactic analysis and treats those errors that certainly need some linguistic information but can be detected using simple patterns in small detection windows (two or three words). In this second phase, we are about to integrate in Microsoft Word® all the steps of the shallow parsing —the morphosyntactic analysis, the syntactic tagging and the chunkers— in order to detect these error-types. As these applications are made to run on Linux, we had to prepare a Windows® runtime integrating all the layers of the shallow parsing. We firstly got a Windows® runtime of Morfeus, the morphological

(2) Aduriz I., Aldezabal I., Aranzabe M., Arrieta B., Arriola J., Atutxa A., Díaz de Ilarraza A., Gojenola K., Oronoz M., Sarasola K., Urizar R. 2002. The design of a digital resource to store the knowledge of linguistic errors. *DRH2002 (Digital Resources for the Humanities). Edinburgo.*

(3) Agirre E., Arregi X., Artola X., Díaz de Ilarraza A., Sarasola K., Soroa A. 1999. Un Diccionario activo vasco-castellano en un entorno de escritura. *VI Simposio Internacional de Comunicacón Social. Santiago de Cuba.*

(4) Aldezabal I., Alegria I., Ansa O., Arriola J., Ezeiza N. 1999. Designing spelling correctors for inflected languages using lexical transducers.*Proceedings of EACL'99, 265-266. Bergen, Norway.*

(5) Aldezabal I., Ansa O., Arrieta B., Artola X., Ezeiza A., Hernández G., Lersundi M. 2001. EDBL: a General Lexical Basis for the Automatic Processing of Basque. *IRCS Workshop on linguistic databases. Philadelphia (USA).*

(6) Aldezabal I., Aranzabe M., Atutxa A., Gojenola K., Sarasola K., 2004. Patrixa: A unification-based parser for Basque and its application to the automatic analysis of verbs. *Inquiries into the lexicon-syntax relations in Basque. University of the Basque Country, Bilbao.*

(7) Alegria I., Aranzabe M., Ezeiza A.,Ezeiza N., Urizar R. 2003. Robustez y flexibilidad de un lematizador/etiquetador. *VIII Simposio Internacional de Comunicación Social. Santiago de Cuba.*

(8) Alegria I.,Balza I.,Ezeiza N.,Fernandez I.,Urizar R. 2003. Named Entity Recognition and Classification for texts in Basque. *II Jornadas de Tratamiento y Recuperación de Información, JOTRI, Madrid. Spain.*

(9) Díaz de Ilarraza A., Sarasola K., A. Gurrutxaga, I. Hernaez, N. Lopez de Gereñu. 2003. HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities. *Workshop on NLP of Minority Languages and Small Languages. TALN. Nantes.*

(10) Ezeiza N., Aduriz I., Alegria I., Arriola J.M., Urizar R. 1998. Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages *COLING-ACL'98, Montreal.*

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

(12) Latteier A., Pelleitier M., *The Zope Book*, New Riders. ISBN: 0735711372. July, 2001.

(13) Petek B., 2000. Funding for research into human language technologies for less prevalent languages, Second International Conference on Language Resources and Evaluation (LREC 2000). Athens, Greece.

(14) Williams B., Sarasola K., Ó´Cróinin D., Petek B. 2001. Speech and Language Technology for Minority Languages. Proceedings of Eurospeech.