Procesamiento del Lenguaje Natural, núm. 43 (2009), pp. 41-48 recibido 1-05-2009; aceptado 5-06-2009

Determiner errors in Basque: Analysis and Automatic Detection

Errores en el uso de determinantes en euskera: Análisis y Detección Automática

Larraitz Uria, Bertol Arrieta, Arantza Díaz de Ilarraza, Montse Maritxalar, Maite Oronoz*
Faculty of Computer Science, Donostia
University of the Basque Country
(*)maite.oronoz@ehu.es

Resumen: En este artículo presentamos un estudio realizado para analizar el uso incorrecto de los determinantes en textos escritos en euskera. El análisis exhaustivo de esta tipología de errores (a través de los ejemplos recopilados) ha sido la base para la detección automática de los mismos. La recopilación y el análisis de errores son imprescindibles para el desarrollo de un corrector gramatical para el euskera y para la creación de sistemas inteligentes de enseñanza de lenguas asistida por ordenador (ICALL).
Palabras clave: Análisis de errores, errores en determinantes, detección automática.

Abstract: In this paper we present the work carried out to deeply study the nature of determiner errors in written Basque. The collected error examples have led us to a more exhaustive analysis which has been essential for the automatic detection of the exhibited phenomena. The analyzed and stored data are necessary for the development of a grammar checker for Basque and Intelligent Computer-Assisted Language Learning (ICALL) systems.
Keywords: Error analysis, determiner errors, automatic detection.

1 Introduction

Within the Natural Language Processing (NLP) research field, we are working on the automatic treatment of grammatical errors with two main aims: the development of a grammar checker for Basque and the study of the learning process of the language in order to create intelligent computer-assisted tools based on learners' needs. Despite the fact that all error types have to be exhaustively analyzed, in this article we focus on the analysis of the incorrect use of determiners in Basque.

Determiner errors are relatively common among language learners. The Basque learner corpora collected in the last years have provided us with enough data to perform a deep study of this error phenomenon. Based on that information, we have created a grammar to automatically detect some types of determiner errors using the Constraint Grammar (CG) formalism (Karlsson *et al*., 1995).

Although CG was created for disambiguation, it has been also used to limit and detect local grammatical errors[1] in many languages due to its mapping abilities. For instance, agreement errors into phrases and word order errors in Catalan are detected by means of CG. This error grammar is used in ALLES, an advanced long-distance language learning system (Schmidt *et al*., 2004) and in PrADO, an error checker for native writers (Badia *et al*., 2004). The grammar checker for Swedish, *Grammatifix* (Birn, 2000; Arppe, 2000), consists of around 650 rules integrated in the CG module to detect 26 error types such as non-agreement into the elements of the noun phrase or lack of coherence in verb phrases. And in the *Grammar Checker for Norwegian* (NGC) (Johannessen, 2002), which is based on the system developed in *Grammatifix*, pattern-rules developed by means of CG are also applied for the automatic detection of some errors.

Language knowledge is explicitly encoded in the rules created for automatic error detection and that information is also useful for the linguistic diagnosis of errors, which is often

[1] Syntactic errors that occur at phrase level.

ISSN: 1135-5948

necessary in computer-aided language learning environments. Therefore, finite-state techniques (such as Constraint Grammar and Xerox Finite-State Tool) and context-free grammars are appropriate approaches for automatic error detection, mainly when the rules have been written for educational purposes.

The first step in the methodology followed for error analysis was the creation of a general, hierarchical and dynamic error classification. The error categories make possible the linguistic diagnosis of the manually annotated examples. In fact, the linguistic diagnosis is, in our case, the starting point for the automatic error treatment.

The paper is organized as follows: Section 2 deals with the main characteristics and problems concerning the use of determiners in Basque. Section 3 presents the general error classification defined to categorize Basque determiner error instances, particularly focusing on the category corresponding to determiner errors. Section 4 describes the manual error annotation carried out for this study. Section 5 tackles the automatic detection of determiner errors in Basque, using Constraint Grammar. Section 6 presents the results obtained in the evaluation of the rules. And finally, some conclusions and future work are outlined in section 7.

2 The use of determiners in Basque

Basque is an agglutinative language in which most words are formed by joining morphemes together and it is said to be a free-word-order language because the order of the phrases in a sentence can vary. On the contrary, the order of the elements that constitute the noun phrase (NP) is fixed: nouns head the NPs, adjectives follow the nouns and determiners (articles and demonstratives) follow the [Noun + Adj] groups; other modifiers such as possessive phrases, postpositional phrases, relative clauses and most quantifiers always precede the nouns. From the point of view of generative linguistics, the determiner, in general, appears in the last position of the NP, in some cases agglutinated to a word, and it takes the entire NP as its complement, constituting the Determiner Phrase (DP) (Laka, 1996). The following examples show a few types of correct determiners and DP structures:

- the singular and plural definite articles: *-a* / *-ak* (the English 'the'), which in Basque are suffixes to nouns and adjectives:

[[[*haurraren*]$_{GEN}$ *jostailu*]$_{NP}$ ***-a/-ak***]$_{DP}$
child of toy(s) the
'the toy(s) of the child'

- the singular and plural indefinite articles: *bat* ('one') / *batzuk* ('some, ones')

[[[*haurraren*] *jostailu*]$_{NP}$ ***bat / batzuk***]$_{DP}$
child of toy(s) a/some
'a / some toy(s) of the child'

- the demonstratives: *hau / hori / hura* ('this/that') / *hauek / horiek / haiek* ('these/those'):

[[[*haurraren*] *jostailu*]$_{NP}$ ***hau/hori/hura***]$_{DP}$
child of toy(s) this/that/these/those
'this/ that / these / those toy(s) of the child'

However, depending on some characteristics of the DP, the use of determiners may vary. Below some correct and incorrect examples of Basque DPs are showed:

- Arguments require a determiner:

*emakume**a** etorri da* ('the woman has arrived')
emakumeØ** *etorri da* ('woman has arrived')

- Predicates in copular sentences require the definite article *-a*:

Anne ona da ('Anne is good')
Anne onØ** *da* ('Anne is good')

- A list of indefinite quantifiers[2] (such as *zenbait* 'some'; *hainbat* 'many, much'; *gutxi* 'few, little'; *asko* 'many') cannot co-occur with any determiner in the same phrase:

***Zenbait** gizonØ* ('some man')
***Zenbait** gizona* ('*some a man')

Hainbat** liburu**k ('many books')
Hainbat** liburu**ak ('*many the books')

[2] Indefinite quantifier are a type of determiner.

These examples show some characteristics of correct and incorrect uses of determiners. Determiner errors are quite common in written Basque, especially in learner corpora, due to the mentioned morphosyntactic variations and the standardization process in Basque. Therefore, Aldabe *et al*. (2007) agree that a deep study of these phenomena is interesting and necessary in our language community.

3 Error classification for Basque

In order to collect, categorize and annotate the errors detected in written texts, a descriptive error taxonomy for determiner errors in Basque was defined. This classification is part of a database created to store error instances. Together with the erroneous examples and linguistic information, technical information for the automatic treatment of errors and psycholinguistic data to carry out different studies in the field of ICALL are also stored in this database (Aldabe *et al*., 2006). In fact, a well-organized and complete repository of errors is a relevant basis for these two research fields we work on.

The error taxonomy consists of six general linguistic categories: spelling errors; lexical errors; morphological and syntactic errors; semantic errors; punctuation errors and style 'errors'. Each category is further divided into more specific subcategories such as determiners, verbs, prepositions, declensions, etc. This paper is focused on the category of determiner errors, which has been specified, redefined and completed based, mainly, on the error instances detected in the annotated corpora. However, theoretical information provided by Basque grammars (Laka; Zubiri and Zubiri, 1995) has been also considered.

Error type	**Error tag**
Deletion of DET	D_DET
Addition of DET	A_DET
Repetition of DET	R_DET
Wrong Order of DET	WO_DET
Wrong DET	W_DET
Definiteness / Indefiniteness	DI_DET
Organic *-a*	ORG_A
Ambiguous cases	DET_ANB

Table 1. Main subcategories of the types of determiner errors and their corresponding tags.

The category of determiner errors consists of eight subcategories (Table 1): deletion of the determiner, when necessary; addition of the determiner, when not necessary; repetition of the determiner in the DP; wrong order of the determiner; definite/indefinite names after certain determiners, when they should be indefinite/definite; deletion of the so-called organic *-a*[3] as if it were the singular definite article *-a*; and ambiguous cases (DPs that are correct/incorrect at phrase level but not at sentence level). Each subcategory contains more specific subcategories that are not described here, and its corresponding error tag.

4 Manual annotation of the corpora

Once the error categories and subcategories were defined, we carried out the manual error annotation of the corpora, according to the tags specified in the classification. In Figure 1 we present three examples of different types of determiner errors annotated in the corpus with their corresponding labels, which delimit the erroneous phrase.

Deletion of DETs (D_DET)
*<**D_DET**>*kotxe*<**D_DET**> *erosi nuen*
car I bought
'I bought **car**'

Repetition of DETs (R_DET)
Euskal Herria* <R_DET**>*nazioa bat*<**R_DET**> *da*
Basque country nation a one is
the Basque Country is '**a one nation**'

Wrong order of DETs (WO_DET)
*<**WO_DET**>*asko lagun*<**WO_DET**> *joan ziren*
many friend went
'**friend many**' went

Figure 1. Examples of manually annotated error instances.[4]

In the manual annotation 788 determiner error instances have been tagged in a 113,290 words learner corpus, consisting of texts written by

[3] Although it is sometimes mixed up with the singular definite article *-a*, the organic *-a* is not the article but part of some lemmas.

[4] The corresponding correct examples of these sentences are: *Kotxea erosi nuen; Euskal Herria nazio bat da; Lagun asko joan ziren*.

learners of Basque of different language competence levels (beginners, intermediate and advanced[5]). The rate of determiner errors in the annotated corpus is of 1.99%.[6] Table 2 shows the number of errors detected in each language level. As expected, the lower the language level, the higher the number of errors:

Language level	nº of errors/ nº of phrases
Beginners	2.87%
Intermediate	2.18%
Advanced	1.43%
All levels	**1.99%**

Table 2. Percentages of the annotated errors in each language level.

Below, Table 3 shows the number of errors manually annotated per each error type. The most annotated error types belong to the D_DET and R_DET categories (66.74%):

Error code	Nº of errors
D_DET	327 (41.49%)
A_DET	67 (8.50%)
R_DET	199 (25.25%)
WO_DET	34 (4.31%)
W_DET	13 (1.64%)
DI_DET	11 (1.39%)
ORG_A	101 (12.81%)
DET_ANB	36 (4.56%)

Table 3. Number of errors per each error type in the manual annotation.

The linguistic diagnosis of the annotated error examples is usually the starting point for the automatic error treatment.

5 Automatic detection of determiner errors

After the manual annotation was finished, we began to work on the automatic detection of the annotated errors, taking into account the characteristics of each categorized error type as well as the analysis provided by the morphosyntactic analyzer. The rules written for the automatic treatment of determiner errors are based on the Constraint Grammar (CG) formalism (Karlsson *et al*., 1995; Karlsson, 1990). As determiner errors occur at noun and determiner phrase levels, they are considered local errors. Our experience in the use of CG, its suitability for the detection of local grammatical errors and the possibility this formalism offers for linguistic diagnosis have led us to use it.

In order to detect five types of determiner errors, 85 rules have been written. Figure 2 shows an example of a rule. For each rule, the error type, the corresponding classification category, the linguistic description of the error, information of the error source, at least an example of each error type and the rule itself are defined. The example rule indicates that the &ERROR_RDET3_1 error tag must be applied if i) the target noun is a common, absolutive and singular noun; ii) the target noun is not a word containing the organic *-a*; iii) the target noun is followed by a definite, singular, absolutive and non-finite determiner.

Error Type
Two determiners in the same Noun Phrase.
Category
Repetition of Determiners.
Error Description
If a singular indefinite determiner comes after a noun ending with a determinative suffix, the phrase is not correct. A noun phrase cannot take either two determiners or a determiner and a quantifier.
Source of Information
Zubiri's Grammar (1995) and learner corpora.
Examples
*Mendi***_*bat*_** ikusten dut.
*mountain ***_*a one*_*** (***_*a one*_*** mountain)
CG based Rule[7]

```
MAP (&ERROR_RDET3_1) TARGET N IF
            (0 COM AND ABS AND SING)
            (NOT  0 ORGA)
          (1 DEF-DET AND SING AND ABS AND NF);
```

Figure 2. Example rule.

[5] When we collected the texts, the language level of each student was already specified.

[6] As determiner errors occur at phrase level, in order to get the percentage of this error type we have taken into account the number of phrases in the corpus (39.546), which has been calculated automatically.

[7] The abbreviations used in the rule mean: N = noun; COM = common; ABS = absolutive; SING = singular; ORGA = organic *-a*; DEF-DET = definitive determiner; NF = non-finite.

Below, Table 4 displays the number of rules written per each treated error type. Neither the number of subcategories analyzed within each error category nor the complexity that each error type implies is equal, and therefore, the number of rules per each treated error category is different. As most of the errors (66.74%) belong to the D_DET and R_DET error categories, the highest number of rules (90.58%) has been written to detect these types of errors. Besides, although all determiner errors have been manually annotated, the automatic detection of all of them has not been carried out yet. In this first approach, we have treated the most common error types.

Error type	**nº rules**
D_DET	19
A_DET	4
R_DET	58
WO_DET	1
DI_DET	3
Total	85

Table 4. Number of rules per each treated error type.

In the analysis of determiner errors, we have found some problems or causes that make difficult the automatic detection of certain errors. For example, the analysis provided by the morphosyntactic analyzer is not always correct; in some cases words have not been correctly disambiguated and do not have the analysis we expect in that context. Besides, as the tagged texts are language learners', there are many spelling errors (6.91%) which make difficult the automatic detection of grammatical errors. Moreover, some phenomena are ambiguous: some phrases or structures can be incorrect at phrase level but correct at sentence level, and vice versa.

Therefore, not all the categories and subcategories defined in the error classification are automatically detectable using only rule based methods; in some cases, other techniques (such as subcategorization, semantics or machine learning, for example) will be also necessary.

6 Evaluation

In order to evaluate the rules written for the automatic detection of some determiner errors, we have defined the following metrics:

- *true positives* (*tp*): wrong instances which are marked wrong by the rules;
- *true negatives* (*tn*): correct instances which are marked correct by the rules;
- *false negatives* (*fn*): wrong instances marked as correct by the rules;
- *false positives* (*fp*): correct instances which are marked wrong by the rules (also called *false alarms*).

The precision and recall of the rules are obtained as follows:

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Precision measures the correctness of the detected errors; recall measures the number of errors detected out of the errors which should have been detected.

In order to carry out this evaluation, learner and native speaker corpora have been used. Learner corpora are necessary to evaluate the number of errors detected as well as the number of the generated false positives or false alarms; native speakers' texts are useful mainly to evaluate the number of false alarms. As regards the learner corpora, it has been split in two parts: development (75%) and test (25%) corpora. The evaluation we present in this article corresponds to the analysis of four error types. Table 5 and 6 present the results obtained in the test corpus. As regards the different error types detected in the learner corpora, the results are displayed in Table 6.

	tp	**fp**	**fn**	**Total**	**Precision**	**Recall**
System	60	72	74	132	45.45%	44.78%

Table 5. Results obtained by the system in the test part of the learner corpora.

From the chosen error types 134 error instances were manually annotated and the system has flagged 132.

Error	Precision	Recall
D_DET	37.8%	43%
R_DET	50%	42.8%
WO_DET	83.3%	45.4%
DI_DET	30%	60%

Table 6. Results of the evaluation, per each treated error type.

In all, out of the 132 flagged errors, there have been 72 (54.54%) false alarms. Nevertheless, 57 (79.16%) of them could be avoided because of the reasons listed below:

- In the tagged learner corpora there are many spelling errors (6.91%) which are not recognized by the morphosyntactic analyzer and are, consequently, analyzed as unknown words. 26 (36.11%) cases out of the 72 false alarms are of this type,. And they could be avoided correcting the spelling errors before applying the grammar checker.
- Apart from spelling errors, in learner corpora there are usually rare structures, although all the words might be correctly written. In this type of sentences, where phrases are not well structured, we have detected 17 (23.61%) false alarms, in this case, errors different to determiner ones.
- The analysis provided by the morphosyntactic analyzer is not always correct, which can also generate false alarms. In fact, out of the 72 false alarms, 14 (19.44%) are of this type. Nevertheless, these examples provide us with feedback to improve the morphosyntactic analyzer.

Table 7 shows the different cases of the detected false alarms:

Case	False Alarms
Unknown words	26 (36.11%)
Rare structures	17 (23.61%)
Incorrect analysis	14 (19.44%)
"Real" false alarms	15 (20.83%)
Total	72

Table 7. Cases which have caused false alarms in learner corpora.

If we do not take into account those false alarms generated in unknown words, rare structures or incorrectly analyzed words, the results would be the following ones (Table 8):

	tp	fp	fn	Total	Precision	Recall
System	60	15	74	75	80%	44.78%

Table 8. Results obtained by the system taking into account only "real" false alarms.

Moreover, the rules have been also evaluated in a 53,658 words corpus composed of texts of a Basque newspaper written in standard language. In all, 55 determiner errors have been flagged (0.33% of the phrases).[8] Table 9 displays the details of the results:

Case	nº of cases
Unknown words	14 (25.45%)
Incorrect analysis	12 (21.81%)
"Real" false alarms	26 (47.27%)
Real errors	3 (5.45%)
Total	55

Table 9. Number of errors and false alarms detected in native speaker corpora.

As in learner corpora, precision rate can be increased if spelling errors are corrected or the morphosyntactic analyzer is redefined and improved before carrying out the automatic detection of grammar errors. In general, we consider more appropriate to keep false positives or false alarms to a minimum, at the cost of failing to identify some grammatical errors. In fact, if the system provides users with too many false alarms, they might have doubts about their language knowledge.

7 Conclusions and future work

In this paper we have presented the work carried out for detecting the incorrect use of determiners in Basque unrestricted texts as well as the evaluation of the rules created for their automatic detection.

As determiner errors are quite common (mainly among language learners' writings), it is interesting to carry out the analysis and the automatic detection of this phenomenon.

[8] In these texts 16.434 phrases have been automatically detected.

A grammar, based on the CG formalism, has been written to automatically detect some types of determiner errors. The rules have been then evaluated in language learners' and native speakers' corpora. The results obtained are rather satisfactory and have been useful to analyze the problems or difficulties that the automatic error detection implies. The data collected and the results obtained will be necessary for the development of the grammar checker for Basque as well as for the study of the learning process of the language.

Without using semantic and pragmatic information, not all the determiner errors manually tagged are automatically detectable. In addition, on the one hand, possible spelling errors, rare structures as well as the incorrect analysis provided by the morphosyntactic analyzer can hinder the automatic detection of grammar errors; on the other hand, the inherent complexity and ambiguity of the language make automatic error detection more difficult. Thus, although CG is considered an appropriate technique for the detection of some local errors occurring at phrase level, other methods will also have to be used together with this CG formalism.

The data collected in the annotation process of this study is interesting for two different purposes: to provide specific data and examples for creating rules to be integrated in a grammar checker; and to show the kinds of problems Basque learners have with the use of determiners. This way, the stored information, apart from being used for automatic error treatment, is also useful for psycholinguistic studies related to the learning and teaching process of Basque.

The present research has been very interesting to redefine the categories of determiners in the initial error classification; to get annotated learner corpora; to collect error instances in the database mentioned in section 3 as well as to give a step further in the development of the grammar checker for Basque. Taking into account the results obtained in this analysis, we have to design the best strategy to avoid as many false alarms as possible so that the output of the checker is good enough for the users.

Although in this paper we have presented only the analysis of determiner errors, we consider the methodology followed in this study suitable for the analysis of other error types. As future work, we will continue analyzing other types of errors to enrich the work carried out in the field of grammatical error detection for Basque. Besides, different error detection systems, which have been carried out using several detection techniques (Oronoz, 2009), will be integrated in a single system so that the grammar checker we are developing for Basque is able to detect different error types.

References

Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Niebla I., Oronoz M., Uria L. 2006. The Use of NLP tools for Basque in a multiple user CALL environment and its feedback. TAL & ALAO workshop. TALN 2006. Leuven, Belgium. In Proceedings of the 13th Conference Sur Le Traitement Automatique des Langues Naturelles. Volume 2. p.: 815-824; ISBN: 2-87463-024-1.

Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Oronoz M., Uria L., Amoros L. 2007. Learner and Error Corpora Based Computational Systems. In Corpora and ICT in Language Studies: PALC 2005, J. Walinski, K. Kredens & S. Gozdz-Roszkowski (eds.), Peter Lang. Vol. 13, 2007. ISBN 978-3-631-56099-0.

Arppe, A. 2000. Developing a grammar checker for Swedish. In Proceedings of the 12th Nordic Conference in Computational Linguistics, Nodalida´99. Department of Linguistics, Norwegian University of Science and Technology, Trondheim, pp. 13-27.

Badia T., Gil A., Quixal M., and Valentín O. 2004. NLP-enhanced error checking for Catalan unrestricted text. In Proceedings of the fourth international conference on Language Resources and Evaluation, LREC 2004, pp. 1919–1922.

Birn, J. 2000. Detecting grammar errors with Lingsoft's Swedish grammar checker. In Proceedings of the 12th Nordic Conference in Computational Linguistics, Nodalida´99. Department of Linguistics, Norwegian University of Science and Technology, Trondheim, pp. 28-40.

Johannessen J.B., Hagen K., and Lane P. 2002. The performance of a grammar checker with deviant language input. In Proceedings of the 19th international conference on

Computational linguistics, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Karlsson F. 1990. Constraint Grammar as a Framework for Parsing Running Text. In Procs. CoLing'90. In Procs. 14th International Conference on Computational Linguistics, ICCL.

Karlsson F., Voutilainen A., Heikkilä J. and Anttila A. (eds.). 1995. Constraint Grammar. A Language-Independent System for Parsing Unrestricted Text. Berlin and New York: Mouton de Gruyter.

Laka, I. 1996. A Brief Grammar of Euskera - The Basque Language. University of the Basque Country, Office of the Vice-Rector for the Basque Language, ISBN: 84-8373-850-3. http://www.ehu.es/grammar/

Oronoz, M. 2009. Euskarazko errore sintaktikoak detektatzeko eta zuzentzeko baliabideen garapena: datak, postposizio-lokuzioak eta komunztadura. PhD thesis. Computer Languages and Systems.

Schmidt P., Garnier S., Sharwood M., Badia T., Díaz L., Quixal M., Ruggia A., S. Valderrabanos A., J. Cruz A., Torrejon E., Rico C. and Jimenez J, 2004. ALLES: Integrating NLP in ICALL Applications. Proceedings of the fourth international conference on Language Resources and Evaluation, LREC 2004. 1919-1922. Lisbon, Portugal.

Zubiri, I. and Zubiri, E. 1995. Euskal Gramatika Osoa. Bilbo. Didaktiker SA.