Evaluating the Impact of Morphosyntactic Ambiguity in Grammatical Error Detection

Arantza Díaz de Ilarraza, Koldo Gojenola and Maite Oronoz
Department of Computer Languages and Systems
University of the Basque Country
{jipdisaa, koldo.gojenola, maite.oronoz}@ehu.es

Abstract

We present a study of the impact of morphological and syntactic ambiguity in the process of grammatical error detection. We will present three different systems that have been devised with the objective of detecting grammatical errors in Basque and will examine the influence of ambiguity in their results. We infer that the ambiguity rate in the input to an error detection tool can have a considerable impact on the quality of the system.

Keywords

Grammatical error detection, morphosyntactic ambiguity

1 Introduction

The relationship between ambiguity and error detection has been mentioned in very few occasions [3, 13]. Similarly to most NLP areas, the development of tools for grammatical error detection and correction finds ambiguity as a main obstacle for the design of efficient and accurate systems. Typically the errors accumulated through the linguistic analysis make difficult the process of detecting grammatical errors. Birn [3] states the following relation between ambiguity, linguistic analysis and grammar error detection.

> "The relationship between disambiguation and grammar error detection is intricate. On the one hand, . . . disambiguation is a prerequisite for any effort at precise error detection. On the other hand, a grammar error may disturb the disambiguation, . . . and this in turn may disturb the error detection"

In this paper we will study this statement over three systems designed for the detection of errors ranging from a restricted and limited context (errors in date expressions and complex postpositions) to the more general case of agreement errors between verb and sentence elements. We will concentrate on morphological and syntactic ambiguity:

- Morphological ambiguity: each word-form can receive multiple morphological analyses, e.g. noun/verb is a typical example of categorial ambiguity. For agglutinative languages there are additional sources of ambiguity (number, case, ...). This poses a problem for grammatical error detection/correction.
- Syntactic ambiguity: this is typically added on top of morphological ambiguity. For example, it is important to exactly know whether an NP is the subject or the object of a verb in order to detect agreement errors.

Level	Linguistic features	Method
M1	POS	CG + HMM
M2	POS + SubPOS	CG + HMM
M3	POS + SubPOS + Case	CG + HMM
M4	All in MORFEUS	CG

Table 1: *Disambiguation levels in* EUSTAGGER.

The remainder of this paper is organized as follows. Section 2 comments on the general morphosyntactic analyzer for Basque, and its parameterization to investigate the impact of ambiguity in error detection. Section 3 will present the experiments with different degrees of ambiguity in three settings. Section 4 compares our work with related systems, ending with the main conclusions.

2 Linguistic resources and parameterization of ambiguity

For the analysis of the input texts, we will use the Basque shallow syntactic analyzer [1]. Instead of using a general purpose analyzer, an alternative approach to error analysis could be the development of specially tailored resources for error processing, but as the creation of tools (morphological analyzer, tagger,. . .) is a very expensive task, we decided to use the one within our reach, and perform the necessary adaptations to deal with ill-formed sentences.

Let us focus in the parts related to ambiguity:

- *Morphosyntactic disambiguation* (linguistic and stochastic disambiguation in figure 1). After morphosyntactic analysis (MORFEUS), the tagger/lemmatiser EUSTAGGER obtains the lemma and category of each form and also performs disambiguation using the part of speech (POS), fine grained part of speech (SubPOS) and case. Disambiguation is performed by linguistic rules using Constraint Grammar (CG) [16] and stochastic rules (HMMs) [9]. Table 1 shows the parameterizable disambiguation levels in EUSTAGGER.

International Conference RANLP 2009 - Borovets, Bulgaria, pages 155–160

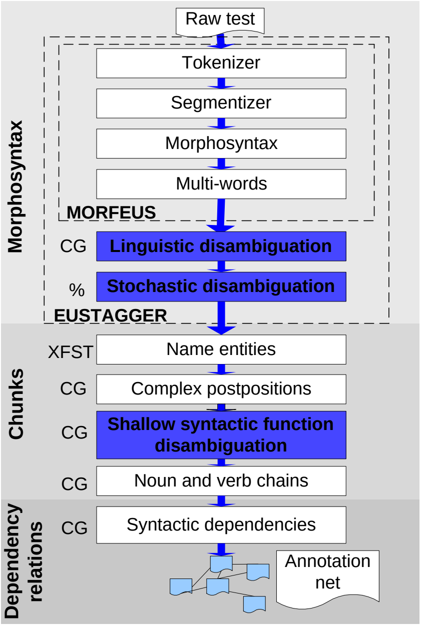

Fig. 1: *The syntactic analyzer for Basque.*

- *Shallow syntactic function disambiguation.* This is carried out in two levels:
  - S1: functions related to noun chunks (@<CM, ...) and functions of verbal chunks (@+FMAINVERB, ...) are disambiguated[1].
  - S2: functions included in S1 and main syntactic functions (@SUBJ, @OBJ...) are disambiguated.

For local error detection only morphosyntactic disambiguation will be used. In the case of agreement errors, however, both morphosyntactic and syntactic function treatment are necessary. For that reason, the experiments will use combinations of morphosyntactic and shallow syntactic function disambiguation. For example the combination M1-S2 indicates the morphosyntactic level M1, and the syntactic disambiguation level S2.

3 Estimation of the impact of ambiguity in Error Detection

We have divided grammatical errors into two groups depending on the context they occur. On the one hand, we will treat "local syntactic errors" that appear in windows of five-six consecutive words following the linear order of a sentence, that is, they usually occur within phrases or chunks. Several tools based on finite-state automata or transducers, such as Constraint Grammar, The Xerox Finite State Tool [17] or Foma [14] can be used to detect these types of errors. On the other hand, there is a group of grammatical errors that needs more sophisticated techniques. For example, in the detection of agreement errors, the elements to be analyzed (verb, subject, object and indirect object) may appear far from each other in the sentence. In the particular case of Basque, they could appear in many different positions due to its free constituent order of nominal elements with respect to the verb. We call these types of errors "global syntactic errors". We will use *Saroi* [6], a tool that we have developed to allow the definition of declarative rules for the detection of errors in dependency-trees. Although, in local syntactic errors correction has also been implemented, the results presented in this paper concern error detection in all cases.

With regard to the corpus used, we use not only general error corpora, but also "correct" corpora. The latter will allow us to test the systems negatively, that is, we will test systems' behavior in respect to false alarms. We think that this approach is interesting for a good evaluation of automatic error detection.

3.1 Local Syntactic Errors: Date Expressions and Complex Postpositions

Errors in date expressions and complex postpositions can be deemed as representative of local syntactic errors. Despite their similarity because their context for detection is limited to a few consecutive words, they also have important differences:

- *Date expressions.* These structures are hardly ambiguous. For example, the following succession of elements is almost always a date:

  `[ place name, ]`^{2} `year month day`

  An example of this structure is, *"1995eko maiatzaren 15"* (15th of May, 1995). It is incorrectly written because in Basque the day number after a month in genitive case must take a case mark. In erroneous date constructions, it is usual to find 2 or 3 errors in the same structure.

- *Complex postpositions.* Postpositions in Basque play a role similar to that of prepositions in languages like English or Spanish, so that, postposition suffixes are attached to the last element of the noun phrase. We have treated those postpositions that are formed by a suffix followed by a lemma (main element) that can also be inflected:

etxearen			*gainetik*		
etxe	+	*-aren*	*gain*	+	*-etik*
(house)		(of the)	(top)		(from the)
from	the	top	of	the	house

  Frequently the incorrect uses of some complex postpositions can have the same form as correct uses of adverbs or names. This makes postpositions morphosyntactically and semantically very ambiguous. Usually, erroneous constructions contain an unique error.

[1] @<CM: modifier of the noun carrying case. @+FMAINVERB: finite main verb.

[2] [ ] symbols indicate that the place name is optional.

3.1.1 Date expressions

We have performed the detection and correction of grammatical errors in date expressions [8] using finite-state transducers (FSTs). Finite-state constraints, encoded in the form of automata and transducers by means of the Xerox Finite State Tool (XFST), are applied to the morphosyntactic analysis of dates. The system is composed of two groups of FSTs, one for error detection and the other one for the generation of correct dates. The system deals with the nine most frequent error types occurring in dates in Basque.

The system was evaluated in a "test list" (items where month-names appear) of 658 sentences (411 for **Dev**elopment and 247 for **Test**), including correct dates, incorrect dates, and also structures similar to dates. Those sentences were extracted from a corpus composed of i) 267 essays written by students and ii) texts from newspapers (presumably correct), more than 500,000 words altogether. Only detection was evaluated, as the generation of correct date expressions guarantees the correction of all the errors in the expression even if not all of them were detected.

In order to evaluate the impact of morphosyntactic ambiguity, we have analyzed the corpora considering different levels of disambiguation (see table 1). Table 2 shows the results. Without disambiguation (WD), or using M1, M2 and M4 disambiguation levels, values of 95.9% recall[3] and 97.8% precision[4] are reached in the development corpus. The system gives 92.1% recall and 89.7% precision over the test corpus (247 test items) in WD, M1 and M2. However, the detection goes down when using the major number of features for disambiguation (M3 includes POS, SubPOS and case) obtaining 76.3% and 87.7% precision and recall in the test corpus. The reason for this reduction is the removal of the analyses needed for error detection, and in consequence, the decrease in the number of detected errors (75 from 93 in the development corpus and 29 from 35 in the test one). There are no changes in the false alarm rate.

3.1.2 Complex Postpositions

We designed and evaluated a set of rules, based on the Constraint Grammar (CG) formalism to detect errors in complex postpositions, constructions that are semantically and syntactically ambiguous [7]. For the description of the incorrect structures, apart from morphosyntactic and syntactic features, the rules were extended with several classes of semantic restrictions (animate nouns, names representing places, ...). For error correction we applied a morphosyntactic generator that uses information extracted from the incorrect structure and from correction schemas.

Being a local error, for the evaluation of this structure we used again "test lists", but in this occasion we made a distinction between those extracted from an error corpus (994,658 word-forms), and those obtained from a "correct" corpus composed of newspapers (8,207,919 word-forms).

The first experiment was carried out again, without disambiguating the analyzed texts. Table 3 shows the evaluation results. In the error corpus we obtained a recall of 81.6% and a precision of 96% (development) and 65% recall and 67% precision (test). The corpus composed of newspapers is lexically richer than the error corpus, and in consequence, the semantic variability of some complex postpositions was higher. That causes an increment of the false alarm rate, leaving the precision in values ranging from 40% to 42%.

	Newspapers Corp.		**Error Corp.**	
	Dev	**Test**	**Dev**	**Test**
Sentences	26679	17786	3884	2590
Errors	-	-	60	29
Undetected	-	-	11	10
Detected	30	24	49	19
False alarms	45	33	2	9
Recall	-	-	81.6%	65%
Precision	40%	42%	96%	67%

Table 3: *Complex Postpositions. Evaluation.*

As the goal of the present work is to analyze the impact of the ambiguity in grammar error detection and not the error detection task itself, we decided to use the biggest corpus for this experiment, in this case the **Dev** corpus. Table 4 shows the evaluation results. Although much variability in the results could be observed, still the option that makes a deeper morphosyntactic disambiguation gives the worse results. The precision falls from 96% in the WD option, to 84.9% in M3 due to the appearance of more false alarms. This may be caused because a deeper disambiguation can remove the correct interpretation of a word-form, which can then be flagged as incorrect.

	Error Corpus. Development				
	Detect	Undetect	FA	Recall	Precision
WD	49	11	2	81.6%	96.0%
M1	43	17	4	71.6%	91.48%
M2	43	17	4	71.6%	91.48%
M3	45	15	8	75.0%	**84.9%**
M4	42	18	3	**70.0%**	93.3%
Sentences	3884				
Errors	60				

Table 4: *Impact of ambiguity in postpositions.*

3.2 Global Syntactic Errors: Agreement

Agreement errors in Basque are very frequent. Finite verbs agree with the subject, object or indirect object of the sentence. These elements can appear in any order in the sentence, and each of them must agree with the verb in case, number and person. This is a source of many syntactic errors, considerably higher than in languages with a more reduced kind of agreement, as English or Spanish.

3.2.1 *A tool for inspecting dependency trees*

For the detection of agreement errors we applied *Saroi*, a system that is used to apply query-rules

[3] recall = correctly detected errors/all errors

[4] precision = correctly detected errors/(correctly detected errors + false alarms)

	Development					Test				
	Detect	Undetect	FA	Recall	Precision	Detect	Undetect	FA	Recall	Precision
WD	93	4	2	**95.9%**	**97.8%**	35	3	4	**92.1%**	**89.7%**
M1	93	4	2	**95.9%**	**97.8%**	35	3	4	**92.1%**	**89.7%**
M2	93	4	2	**95.9%**	**97.8%**	35	3	4	**92.1%**	**89.7%**
M3	75	4	2	77.3%	97.4%	29	3	4	76.3%	87.7%
M4	93	4	2	**95.9%**	**97.8%**	34	3	4	89.5%	89.5%
Sentences	411					247				
Errors	97					38				

Table 2: *Impact of ambiguity in date error detection.*

to dependency-trees. *Saroi* has as input a group of analysis-trees and a group of rules, and obtains as output the dependency-trees that fulfill the conditions described in the rules. Its main objective is the analysis of linguistic phenomena in corpora. Figure 2 shows an example of a rule that detects the error in the dependency-tree of figure 3. In the sentence the subject *zentral nuklearrak* (nuclear power station), in absolutive case, and the auxiliary verb, *dute*, which needs the subject to be in ergative case, do not agree. Specifically the first rule asks that the current word (which should be the main verb) has a subject, and this subject has a modifier (which contains the grammatical case). The verb has an auxiliary verb as dependent (linked by the auxmod dependency arc), which is transitive. If these two conditions hold, then the auxiliary verb and the subject should agree in case. If they do not, then an agreement error occurs.

```
AGREEMENT_SUBJ_CASE_N_NK
(
Detect (
      @!ncsubj!ncmod∼ &
      @!auxmod.type == ‘transitive’ &
      @!ncsubj!ncmod.case != @!auxmod.nork.case
      )
)
```

Fig. 2: *Example of a rule.*

Saroi uses as input the result of the partial syntactic analyzer (see section 2), in which the relations between the elements of the sentence are ambiguously represented. *Saroi* constructs all the set of non ambiguous trees starting from an initially ambiguous tree (see figure 4). The error-detection rules are applied to the full set of dependency-trees. Having in mind the errors accumulated in the analysis chain, we choose a conservative approach: we decide that an agreement error occurs in a sentence when an error detection rule matches all the analysis-trees.

3.2.2 Experiments

Due to morphosyntactic and syntactic ambiguity, a number of trees ranging from 1 to more than 100 is generated for each sentence. In addition, several difficulties must be taken into account:

- NP ellipsis is common in Basque. This makes it difficult to know if a sentence is correct or not, as there may be several ellided elements.

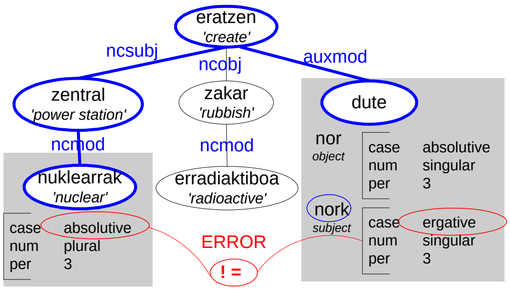

Fig. 3: *Dependency-tree for the sentence* *Zentral nuklearrak zakar erradiaktiboa eratzen dute *(*Nuclear power station create radioactive rubbish).*

- The syntactic analyzer obtains partial analyses and, therefore, not all the elements of the sentence appear in the dependency-trees due to lack of coverage, increasing false alarms.

We decided that in agreement error detection the best option for disambiguation should be chosen before starting the evaluation because to test the rules with all the possible disambiguation options is too time consuming. Considering all the disambiguation combinations, the best criteria should be the ones that:

- Detects the higher number of errors in ungrammatical sentences.
- Gives the lower number of false alarms in grammatical sentences.
- Generates the lower number of analysis trees for each sentence (efficiency).

Our strategy to obtain the best disambiguation option was to chose first the morphosyntactic disambiguation level, and then we selected the best option for syntactic disambiguation.

In order to choose the best morphosyntactic disambiguation level we selected a set of 10 ungrammatical sentences and their respective corrections, which were analyzed with the eight disambiguation combinations (see table 5). The combinations generating the lower number of trees, with acceptable detection and false alarm rates were those making the deepest morphosyntactic disambiguation (M3-S1 and M3-S2).

Next, we aimed at selecting the best syntactic function disambiguation level. We soon realized that the grammar that assigns the dependency-relations to grammatical texts needed of relaxation when applied

Disambiguation combinations								
	M1-S1	M2-S1	**M3-S1**	M4-S1	M1-S2	M2-S2	**M3-S2**	M4-S2
Number of trees	67,7	67,7	**27,8**	46,7	22,11	22,11	**11,6**	10,33
Errors in ungrammatical	5	5	**6**	6	5	5	**6**	6
False alarms in grammatical	0	0	**1**	1	0	0	**1**	0

Table 5: *Looking for the best morphosyntactic disambiguation-combination.*

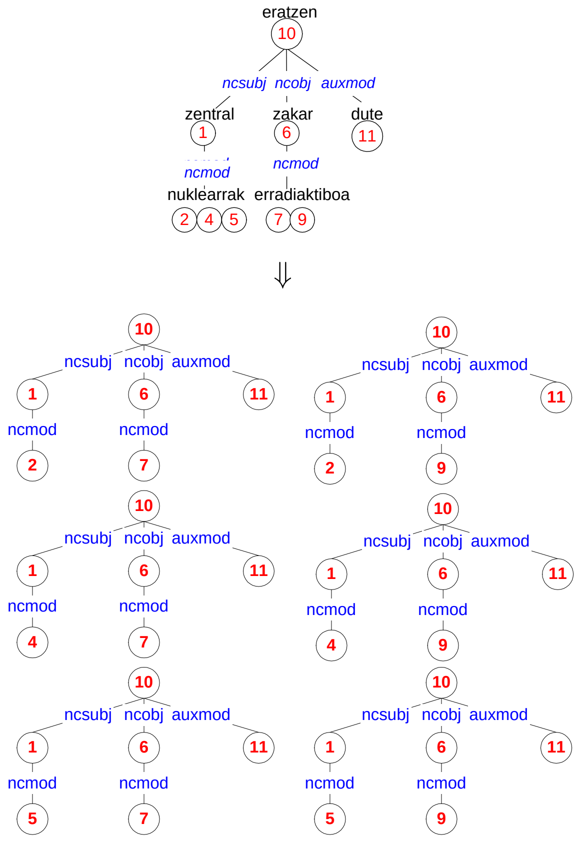

Fig. 4: *Ambiguous dependency tree and the corresponding non-ambiguous trees.*

to ill-formed texts. For example, in the incorrect sentence "*nik ez nago konforme" (*I do not agree*), the word "nik" (*I*) was not tagged as *subject* because it carries the ergative case, and the auxiliary verb "nago" asks for a subject in absolutive case. We experimented relaxing all the conditions referred to the type of auxiliary verb in the rules assigning *subject*, *object* and *indirect object* relations. In a second experiment we used a corpus of 75 sentences containing an agreement error and 75 of their corrections. The sentences were analyzed with the following combinations: M3-S1-Relaxed, M3-S1-NotRelaxed, M3-S2-Relaxed and M3-S2-NotRelaxed. Table 6 shows that the best results were obtained with the M3-S2-Relaxed option. In this experiment we reach interesting conclusions related to error detection:

- In 76.9 % of the cases (20 out of 26), the error was not detected because dependency-relations were incorrectly assigned.
- Sometimes the error was detected due to an incorrect analysis. A false detection occurs.

Opposite to what happened in local error detection, in this case the combination with the best results was the one that disambiguates most.

The work carried out in agreement error detection shows us that when the dependency-relations are incorrectly tagged, error detection is very difficult. The improvement of the syntactic analyzer will bring as a result a better error detection.

4 Related work

To choose the more appropiate approach to face up the problem of grammatical error detection is not a trivial decision. In this section we review some error detection approaches, and at the same time we try to justify our choice of using knowledge-based techniques, as opposite to statistic-based ones.

In our opinion, for error types related to the omission, replacement or addition of elements, empirical approaches are suitable. For example, in [19] and [5] machine learning techniques are used to detect errors involving prepositions in non-native English speakers. Although both English prepositions and Basque postpositions have in some part relation with semantic features, postpositions are, in our opinion, qualitatively more complex, as they are distributed across two words, and they also show different kinds of syntactic agreement, together with a high number of variants. A deeply studied area using machine learning techniques is that of the "context-sensitive spelling correction" [12, 4]. Izumi *et al.* [15] use empirical techniques to detect omission- and replacement-type grammatical and lexical errors in Japanese learners of English. Bigert and Knutsson [2] prove that precision in error detection is significantly improved when unsupervised methods are combined with linguistic information.

The error types we are working with are in all cases related to agreement. Linguistic features of several elements belonging to phrases or sentences must be compared in order to be able to detect the potential errors. This is one of the reasons why we decided to use a knowledge-based approach. Similar methods have been used for grammatical error detection using approaches based on context free grammars (CFG) or finite state techniques. In the first case, for analyzing ungrammatical sentences by means of CFGs, the "relaxation" of some constraints in the grammar has been necessary [18, 11], or error grammars have been developed [10]. When finite state techniques are used, error patterns encoded in rules are applied to the analyzed texts. We follow the second approach as rules encoded using CG, XFST or the query-rules of *Saroi* are applied to the linguistic analysis of the texts.

Disambiguation combinations				
	Relaxed		NotRelaxed	
	M3-S1	M3-S2	M3-S1	M3-S2
Errors in ungrammatical (EE)	42	**40**	36	34
False alarms in grammatical (FA)	23	**16**	18	15
Real detection (EE - FA)	19 (25.33%)	**24 (32%)**	18 (24%)	19 (25.33%)

Table 6: *Looking for the best syntactic function disambiguation-combination.*

5 Conclusions and future work

In this work we have presented the impact of morphosyntactic and syntactic ambiguity across three different types of error detection systems. Two of the systems detect and correct local syntactic errors, and the last one detects global syntactic errors. The results of the experiments show that the influence of morphosyntactic ambiguity in grammatical error detection is undeniable. We can assert that it is not always true that “it is obvious that disambiguation is a prerequisite for any effort at precise error detection”. In local syntactic error detection the best results have been obtained when the analyzed texts are not disambiguated (in the case of dates the same results are achieved if some types of disambiguation are performed). The reason is that before the disambiguation process starts, all the set of interpretations for each word is within our reach, both “correct” and “incorrect” interpretations. When disambiguation is performed, sometimes the interpretations we are interested in, are removed. We must bear in mind that disambiguation rules are generally written having grammaticality in mind. In the case of global syntactic error detection, nevertheless, the best results are obtained when the deepest disambiguation is used at morphosyntactic level, and also at syntactic level. In our opinion, this phenomenon is due to the explosion in the number of generated trees when “all” the possible ambiguity is considered. We think that each kind of error type asks for a specific study of the influence of ambiguity, specially when using knowledge-based techniques.

In all the presented systems one of the main goals is to process real texts with high precision error detection, minimizing false alarms, which are the main bottleneck in current grammar checking systems.

Acknowledgments

This research is supported by the Basque Government (IT-397-07).

References

[1] I. Aduriz, M. Aranzabe, J. M. Arriola, A. Díaz de Ilarraza, K. Gojenola, M. Oronoz, and L. Uria. A cascaded syntactic analyser for Basque. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 5th International Conference CICLing2004, Seoul, Korea, February 15-21*, volume 2945 of *Lecture Notes in Computer Science*, pages 124–134. Springer-Verlag GmbH, 2004.

[2] J. Bigert and O. Knutsson. Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In *Romand 2002 (Robust Methods in Analysis of Natural language Data)*, Frascati, Italy, 2002.

[3] J. Birn. Detecting grammar errors with Lingsoft’s Swedish grammar-checker. In *Proceedings from the 12th Nordiske datalingvistikkdager*, Department of Linguistics, Norwegian University of Science and Technology (NTNU), 2000.

[4] A. J. Carlson, J. Rosen, and D. Roth. Scaling up context-sensitive text correction. In *Proceedings of the Thirteenth Innovative Applications of Artificial Intelligence Conference (IAAI-01)*, pages 45–50, Menlo Park CA, 2001.

[5] M. Chodorow, J. Tetreault, and N.-R. Han. Detection of Gramatical Errors Involving Prepositions. In *4th ACL-SIGSEM workshop on Prepositions*, Prague, 2007.

[6] A. Díaz de Ilarraza, K. Gojenola, and M. Oronoz. Design and development of a system for the detection of agreement errors in Basque. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 6th International Conference CICLing2005, Mexico City, Mexico, February 13-19*, volume 3406 of *Lecture Notes in Computer Science*, pages 793–803. Springer-Verlag GmbH, 2005.

[7] A. Díaz de Ilarraza, K. Gojenola, and M. Oronoz. Detecting Erroneus Uses of Complex Postpositions in an Agglutinative Language. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, Manchester, 2008.

[8] A. Díaz de Ilarraza, K. Gojenola, M. Oronoz, M. Otaegi, and I. Alegria. Syntactic error detection and correction in date expressions using finite”-state transducers. In *Proceedings of Finite”-State Methods and Natural Language Processing (FSMNLP07)*, Postdam, Germany, 2007.

[9] N. Ezeiza, I. Aduriz, I. Alegria, J. Arriola, and R. Urizar. Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *COLING 1998*, Montreal, 1998.

[10] J. Foster and C. Vogel. Good reasons for noting bad grammar: Constructing a corpus of ungrammatical language. In *Pre-Proceedings of the International Conference on Linguistic Evidence: Empirical, Theoretical and Computational Perspectives*, Tübingen, Germany, 2004.

[11] K. Gojenola and K. Sarasola. Aplicación de la relajación gradual de restricciones para la detección y corrección de errores sintácticos. In *Actas del X congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural, (SEPLN)*, volume 4, Córdoba, Spain, 1994.

[12] A. R. Golding and D. Roth. A Winnow-Based Approach to Context-Sensitive Spelling Correction. *Machine Learning*, 34(1-3):107–130, 1999.

[13] S. Hashemi. Detecting Grammar Errors in Children’s Writing: A Finite State Approach. In *Proceedings of the 13th Nordic Conference on Computational Linguistics (NoDaLiDa’01)*, Uppsala, Sweden, 2000.

[14] M. Hulden. Foma: a Finite-State Compiler and Library. In *Demo in the proceedings of EACL*, 2009.

[15] E. Izumi, K. Uchimoto, T. Saiga, T. Supnithi, and H. Isahara. Automatic error detection in the Japanese learners’ English spoken data. In *ACL ’03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 145–148, Morristown, NJ, USA, 2003.

[16] F. Karlsson, A. Voutilainen, J. Heikkila, and A. Anttila. *Constraint Grammar: Language-independent System for Parsing Unrestricted Text.* Prentice-Hall, Berlin, 1995.

[17] L. Karttunen, T. Gaál, and A. Kempe. Xerox finite state tool. Technical report, Xerox Research Centre Europe, 1997.

[18] R. Teixeira Martins, R. Hasegawa, M. D. G. Volpe Nunes, G. Montilha, and J. Osvaldo Novais De Oliveira. Linguistic issues in the development of ReGra: A grammar checker for Brazilian Portuguese. *Natural Language Engineering*, 4(4):287–307, 1998.

[19] J. Tetreault and M. Chodorow. The ups and downs of preposition error detection in esl writing. In *Proceedings of Coling,*, Manchester, 2008.