Integrated set of tools for robust text processing

Fernando Sánchez-León †
Flora Ramírez Bustamante †
Thierry Declerck ‡

† Laboratorio de Lingüística Informática
Facultad de Filosofía y Letras
Universidad Autónoma de Madrid
28049 Madrid
Spain
E-mail: {fernando, flora}@maria.lllf.uam.es
http://www.lllf.uam.es

‡ DFKI GmbH
Language Technology Lab
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
Germany
E-mail: declerck@dfki.de
http://www.dfki.de

abstract

Corpus annotation practices for most major languages have moved the focus, within the NLP research community, on low-level linguistic analysis. This trend is reinforced by the relative failure of grammar based applications and the need for fast and robust document processing for tasks as varied as information indexing or language correctness checking.

This paper describes a set of tools for morpho-syntactic annotation of Spanish texts, based both on well known public domain tools (emerging from MULTEXT) and proprietary technologies (as Constraint Grammars). Besides, a complete bunch of new specific modules ranging from morphological analyzers to form, typographical and morpho-syntactic checkers have been integrated in this NLP tool.

1. Overview

This paper describes an integrated tool for linguistic annotation of running texts. The tool has been built upon some public domain as well as proprietary modules with the idea of building a generic morpho-syntactic annotator/tagger for Spanish texts. In order to demonstrate its usability for specific NLP applications based on the resulting linguistic annotation, typographical and grammar checking procedures of a low-level linguistic nature have been successfully integrated within the core components of the tool. This integration work showed some interesting properties in the sense that the checking application built upon the morpho-syntactic annotator/tagger can be used to remedy to some extent intrinsic weaknesses of some components of the tools, thus improving the quality of the core components and consequently also of the integrated application.

This leads to an interesting discussion: to which extent can form checking not only be built upon morpho-syntactic tools but also interact with them? This question is addressed along the sections of the paper, but further experiments still have to be done in order to provide for qualified results. Another aspect which deserves attention and more work concerns the relationship between grammar checking and high level linguistic analysis. But the paper here will concentrate on achieved work done in the context of low-level linguistic processing.

2. Tokenization

NLP systems perform tokenization of natural language applying, generally, the same techniques used in the

development of compilers for programming languages. This situation is far from perfect as a means to get the complexity (and ambiguity) of natural language and hence a number of tokenization errors are produced by any such tokenizer. This is precisely the approach taken by the MULTEXT segmenting tool, `mtSeg`, upon which the tool's tokenizer has been built[1].

Those segmentation errors potentially affect the quality of the subsequent morphological analysis and disambiguation. Besides, punctuation errors in the input are interfering with the recognition of word boundaries (one example being the use of commas not respecting the correct spacing around them).

As a remedy for this situation, we aim at interpolating text analysis modules with subsets of typographical, lexical and grammar checking rules which will operate modularly and co-operatively with each of these modules, rather than performing all verification tasks after morpho-syntactic analysis (although at the moment this is the current procedure). This co-operative approach is based on the anticipation of common errors in input texts and its ultimate goal is to perform a *text normalization* during which restricted sets of wrong wordforms and inadequate lexical units, which are not stored usually in general purpose lexicons, can be as well recognized[2]. Text normalization would improve the results from text analysis and verification tasks.

3. Morphological analysis and guessing

Morphological analysis is performed via fullform lexicon lookup. This fullform lexicon is created by a morphological generator which consults a lemma-based lexicon containing linguistic as well as lexicographic information (definition, etymology, usage, etc.). The generator is a `perl` script that interprets paradigm information and applies a cascade of morphographemic changes to input lexemes according to concatenating conditions[3]. The interesting idea with respect to spelling and grammar checking is that the generator is capable of relaxing certain morphographemic rules and produce a set of fullforms that account for common cognitive errors in inflectional morphology. These errors, given their string distance from the correct form, are not captured by standard spelling checkers.

Morphological analysis is defined, basically, as lookup in the fullform lexicon (which is realized as a hash). However, a thorough treatment of other morphological phenomena, ranging from derivational morphology (both prefixes and suffixes, and both category changing and so called 'appreciative' morphology) for major categories to enclitic pronouns to verbal forms, allows the system to analyze many more running forms. The latter are specially important since they constitute some 0.4 to 0.6% of text words in any document. All information relative to these components is declaratively expressed and, thus, easily modifiable by the potential user.

Besides this, the analyzer (again a `perl` script) is capable of guessing morpho-syntactic information for unknown words (those not included in the lexicon and not having a possible analysis according to morphological analysis modules) taking into account suffix information, typographic form and morpho-syntactic context.

4. Disambiguation

Morpho-syntactic disambiguation is performed by means of a *Constraint Grammar* developed for Spanish[4]. This formalism combines three operations in order to perform surface-syntactic analysis of morphologically analyzed unrestricted text: namely, context sensitive disambiguation, assignment of clause boundaries and assignment of surface-syntactic functions. However, our current implementation is based only on the first of these operations, the reason being our desire to restrict ourselves to linguistic information usually present in other taggers[5]. This approach allows us to use a less powerful formalism and assert assump-

[1] A description of this segmenter can be found in `http:www.lpl.univ-aix.fr/projects/multext/`.

[2] With wrong wordforms we are not referring to error recognition on character level, work usually done by typographical spellcheckers, but rather to cognitive errors caused by lack of competence. Certain kind of wrong wordforms is far away from being detected by commercial spellcheckers because of the string/character distance between the wrong and correct forms. Besides, this kind of products either does not detect special words which should be forbidden in written texts or are not capable of guiding the user towards correctness in writing. In some way, this task could be related to that of controlled language, if the ultimate goal of the tool is to meet the style requirements of a given user having his/her own style book.

[3] This implementation has been easily migrated to a two-level and unification-grammar based morphological generator such as `mmorph`, developed as part of the MULTEXT tools for linguistic processing of corpora (Petitpierre and Russell, 95).

[4] We thank Lingsoft, Inc., and specially Prof. Karlsson, for their permission to use the Constraint Grammar Development Package.

[5] In fact, the information used by the CG module is the same proposed and used for a port to Spanish of the well known Xerox stochastic tagger (Cutting et al., 1992; Sánchez-León, 1995; Sánchez-León and Nieto-Serrano, 1997).

tions on it that can be easily migrated to other formalisms.

The major concern during development has been that of speed of grammar construction rather than recall preservation. The idea, already demonstrated by (Chanod and Tapanainen, 1995), is that a constraint-based tagger that surpasses a statistical tagger in accuracy can

Table 1: Results on a fragment of *Bélver Yin*

# Words	4106[1]				
Grammar	None[2]	1–2	(1–2)+3–4	(1–4)+1–2	(1–4+1–2)+F
#Analyses	6366	4949	4400	4394	4106
Ambiguity	1.55	1.21	1.07	1.07	1
Errors	9	16	59	59	114
Recall	99.78	99.61	98.56	98.56	97.22
Precision	64.36	82.64	91.98	92.1	97.22

[1]Including punctuation signs.
[2]Indicates text after ambiguous morphological analysis. Grammars are numerated from 1 to 4, being F the final force grammar.

be implemented in the same amount of time used to bootstrap the stochastic model. In this context, only seven weeks have been allocated to the CG development. Because of this, some of the phenomena to be dealt with have been implemented in a less than perfect version.

Constraint knowledge has been organized into four different grammars, ranging from full reliability to more heuristic rules and/or partial treatment. Once the treatment of a certain phenomenon is refined, some of the rules concerning its disambiguation can be promoted to upper grammar levels. Currently, the former two grammars make reliable statements using local and non-local information, respectively. On the other hand, the latter two are mainly heuristic, ranging from heuristical disambiguation using some context information to disambiguation for certain common lexical items even if the context cannot be proven, respectively. Besides, since our desire is to have a completely disambiguated output, a final grammar is used, taking decisions simply on ambiguity classes without looking at context. The former two grammars contain about 250 rules while the latter two contain 120 rules. With this configuration of grammars results obtained on a previously unseen fragment of a modern novel can be seen in table 1.

5. The grammar checking application

The grammar checking application integrated within the previous tools, called CON-TEXT [6], provides writing support including checking of typographical, lexical cognitive and morpho-syntactic errors, as agreement in nominal phrases, as well as checking of wrong sequences of wordforms relative to spelling errors in quasi-homographs, and contextually bound errors, as bound prepositions or case marking prepositions.

CON-TEXT catches errors with a uniform technique which, depending on the error type, is fully based on the morpho-syntactic or tokenization information attached to sequences of wordforms. With this information CON-TEXT delimits the contextual components and the structure of a great variety of error situations.

Although CON-TEXT is a checker of the low-level type and its syntactic capabilities are based on shallow analysis, it is able to recognize errors of the low- and high-level type. As (Kettunen, 1996) demonstrates, the demarcation line between them is not clear cut, and sometimes low-level errors (as typographical ones) can encounter high-level problems, and high-level errors (as agreement ones) can be solved in purely sequential terms (see Ramírez Bustamante, Sánchez-León and Declerck, 1997, and Ramírez Bustamante and Sánchez-León, 1996a for a discussion on this issue)[7].

Checking rules account for incorrect sequences of possibly ambiguous (morpho-syntactic) tags, lexical items or typographical marks. Thus, CON-TEXT is a typographical and morpho-syntactic checking appli-

[6] This project was funded by the Dirección General de Investigación of the Comunidad de Madrid, ref. 05C/002/96 and it has produced two prototypes: the one we are presenting here and another one integrated in `Windows95/NT,` implemented with finite state techniques.

[7] Within spelling errors, omission or addition of written accent constitutes one of the most common spelling errors in Spanish. Commercial spelling checkers do not detect this kind of errors when it is a legal word (e.g. *el*_article, the, vs *él*_pronoun, he). These spelling errors, however, interfere with the syntactic level of description considering that they usually convey a change of category, what will produce a parsing failure in a grammar checker of the high-level type which carries out a full syntactic analysis.

cation for Spanish, based on hand-crafted rules (it is currently implemented in `perl`), and the error detection technique is fully based on the error anticipation technique.

Most of the checking rules make an extensive use of the provided linguistic information, thus giving some linguistic motivation for the description and detection of errors at a 'pre-processing' or 'shallow' level. They describe erroneous sequences of tags, classes of tokens, and/or wordforms within the sentence boundaries. Besides this, they demonstrate to what extent ambiguous morpho-syntactic descriptions provide anchor points for grammar error detection. Working with ambiguous morpho-syntactic descriptions constitutes, in our opinion, a sound basis for avoiding the noise derived from other components in text analysis, since:

- the error rate from a tagger is a potential source for precision degradation in grammar error detection, and
- a tagger could process sequences of erroneous wordforms belonging to ambiguity classes (homographs) in such a way that the errors are finally deleted from the text and no longer detectable.

For these reasons, we have explored to what extent the grammar checking can be performed on ambiguously tagged text. However, where this methodology clearly falls short in order to improve recall (especially when agreement errors are at stake), some rules have been devised that work under the assumption that the text has been (partially) disambiguated. This is done by means of relaxing certain category ambiguities contained in the ambiguity class of a focussed word. This methodology does not ensure the modularity and economy of the whole system, since tasks are not clearly distributed and some processes, as disambiguation, will have to use similar reasoning to do their job afterwards. In fact, lowering the level of error processing does not resolve the two main problems of grammar verification: the level of the linguistic description to which a given error belongs is not always the level at which it can be (optimally) processed, and every error category requires a different technique, handling different linguistic information, which is not always available at the level where the error will be processed[8]. For instance, whilst checking agreement errors needs disambiguation, checking error sequences of wordforms does not. For the former error type, checking should be performed after disambiguation, but for the latter checking can be performed before, and this will help the tagger to perform a more reliable disambiguation of the elements surrounding the error.

In the future we aim at performing verification tasks in different steps by breaking the sequentially chained steps presently used in our text analysis and allowing a *forward/backward emulation process*, which will enable the whole system to improve the results from text analysis and checking. This new approach will provide the system with the ability to perform checking tasks gradually, at different stages, interacting co-operatively with the texts analysis tools.

For the time being only local rules are implemented and they are distributed over four detection modules, corresponding to the categories of errors actually covered by the application: typographical and/or punctuation errors, lexical cognitive errors, (local) agreement errors and error sequences of wordforms which are quasi-homographs (paronyms). A high level of precision was reached in the error detection work, because of the process of ambiguity classes and subclasses. This means that the detection of errors is done not only in non-ambiguous contexts, but also in certain contexts which, although ambiguous, contain an ambiguity class or subclass where there is no category able to block the detection of the error.

The modules can be optionally activated as well as the display of messages. Messages associated with rules are kept in external files and they are parametrized, as well as the rules which flag them, into two main basic types: those flagged because of the detection of an error, and those flagged because of the detection of a sequence of tags or wordforms whose correction could normalize the document. The former type of errors is considered severe errors. Some examples of this error type are described in section 5.1. The latter errors are specially concerned with style weaknesses which are not, strictly speaking, errors since they are well spread over the language. However, it has been considered that a message about the correct form could help users to make a decision about their documents. Examples of these are foreign structures or literal translations mostly from English and/or French (e.g. *la carta a enviar*, the letter to be sent, *esponsorizar*, to sponsor), and wordforms belonging to irregular inflectional paradigms which have been normalized by analogy (e.g. *déficit* is an invariant form, but the form *déficits*, deficits, is quite common).

Messages are based on a heuristic diagnosis of the detected error sequence. They are in the majority of cases adequate for each error situation, but when they

[8] As mentioned above, spelling errors can trigger problems in the phrase structure (i.e. **tú coche*, 'you car'), or in complex verb forms (i.e., **a ido*, 'to gone'). It is not necessary to arrive to the syntactic level of processing in order to detect this kind of errors. However, word checking level is not the place where the error situation can be caught. On the other hand, whilst the detection of most of the errors on agreement needs a syntactic processing, where the feature relaxation technique is the best one for diagnosis, some of them can be processed with local information available at the level of the morpho-syntactic analysis. See also (Oliva, 1997) for a similar hypothesis.

are not, they indicate tendencies which allow users to make their own diagnoses and corrections[9].

A corpus of naturally occurring errors has been elaborated, extracting them manually from published texts and prescriptive dictionaries of Spanish. They were distributed according to a typology built on the GramCheck model (Ramírez Bustamante and Honrado, 1994) and partly modified to account for the new error types covered by CON-TEXT. This is an important feature, as CON-TEXT aimed at avoiding the automatic generation of error data applying «corruption» rules to detect erroneous lexical units or syntactic structures. This methodology, used by the commercial checkers we tested (see section 5.2.), has the disadvantage of producing a great number of false errors detections, because error rules are produced automatically without linguistically motivated contextual restrictions.

5.1. The handling of the error types

Rules for typographical errors are anchored to existing punctuation/typographical marks in the text, thus it does not help in grammatical placement of punctuation marks. They detect errors concerning spaces, unbalanced punctuation marks, combination of punctuation signs...

Agreement errors are anchored to the morpho-syntactic descriptions provided by the morphological analysis. Rules describe local error contexts of sequences of tags. Morphological ambiguities are selected or rejected wrt. the ambiguity class they belong to, and the contexts where they appear. If all the morpho-syntactic descriptions of a given wordform belong to a nominal lexical category, they are selected; otherwise, they are rejected if they do not appear in a context which solves the ambiguity (for example, an ambiguity class of the type verb/noun can be disambiguated in a prepositional context). This selection ensures that detection is done over non ambiguous nominal categories or ambiguous ones within this macroclass, so that non tractable ambiguous sequences are rejected (e.g. *los*_pronoun_masc_pl/art_masc_pl *enfoque*_verb_1P_pres_subj/verb_3P_pres_subj/noun_masc_sing, 'he/she focus them, the foci'). Since Spanish is a language with a high level of homographs, this mixed strategy allows an adequate balance between recall and precision[10].

The rules responsible for the detection of erroneous sequences of lexical categories in restricted contexts are anchored to paronyms since these words are precisely a common source of errors (e.g. *se*_pronoun vs *sé*_verb vs *el*_article vs *él*_pronoun, *ha*_verb vs *a*_preposition). They describe the occurrences of a set of lexical categories which either at the left or at the right should be satisfied for a given context.

Additionally, some lexical pattern errors including the lack of an expected preposition in limited contexts are detected (e.g. **informar que*, to inform that, **inferior que*, inferior than...), as well as wrong sequences of categories (like locative adverb and possessive pronoun instead of preposition and oblique pronoun, e.g. **alrededor mío*, around mine). The error grammar contains 225 rules based on our typology of errors.

Finally, the lexical module of CON-TEXT aims at overcoming the shortcomings of spelling checking based on 'minimal edit distance'. On the other hand, it provides the system with the ability to recognize 'problematic' lexical units which are normally accepted by commercial products as correct forms. The lexicon contains 1100 wordforms of this error category.

5.2. Evaluation

Preparation of the benchmark corpus

For the purpose of the evaluation of the checking tool, four raw texts have been extracted from different electronic versions of newspapers (*ABC*, *El País*, totalizing 44922 words) and submitted to the morphological analysis. Additionally, a text containing a collection of sentences, gathered from newspapers, including errors of different nature (spelling, typographical, agreement, error sequences of lexical categories, unexpected input...) has been composed (12287 words). This text

[9] It is important to note that CON-TEXT is not able to diagnose, in the strict meaning of this word in the context of grammar checking, given that it is not able to show the exact word where the error is located. This limitation is crucial for certain error sequences regarding suggestion adequacy, that is, the way in which the system proposes a diagnosis for an error sequence. For instance, CON-TEXT is able to detect that an article cannot be followed by a verb. However, there are error situations like *el*_article *numero*_1P_present_indicative, 'the number', where there are several possible corrections, the best one being the replacement that requires the minimal number of changes to transform one word into another: *el*_article *número*_noun, 'the number', - one change for correction -; *él*_3P_pronoun *numeró*_3P_past_ind, he numbered, - two changes -, and *él*_3P_pronoun *numera*_3P_pres_ind, he numbers, – two changes –. For the time being, CON-TEXT does not correct the errors that it finds. This means that users should be, in some way, sensitive to specific language problems, in the sense that a suggestion message indicating that two words cannot be in one sequence could be opaque to some users, even if the message points to possible spelling errors in the sequence at hand.

[10] Therefore, there exist serious problems and limitations in the detection of agreement errors, for instance, in reduced clauses. In the sentence *salieron de la* ***clase juntos*** (they leave the class together) the two words in bold are checked. These problems lead to the necessity of deeper levels of analysis in order to detect this kind of sequences, which, on the other hand, are not so frequent in texts.

provides the base for the description of the test suite. Those texts were not used during the elaboration of the rules.

Such texts has been checked with CON-TEXT as well as with two other commercial products: the grammar checker of Word 7 by Microsoft and WordCorrect, a grammar and spelling checker for Spanish, developed

Table 2: Results on texts from electronic versions of newspapers

# Words		44922							
		HUMAN		CON-TEXT		Word 7		WordCorrect	
#Errors		116		67		157		266	
Typograph.	**Non-typogr.**	77	39	49	18	64	93	117	149
Number of Real Errors				49	14	48	16	50	25
#Total Real Errors				63		64		75	
Recall				54,31%		55,17%		64,65%	
Precision				94,02%		40,12%		28,19%	

Table 3: Results on the test suite

# Words		12287							
		HUMAN		CON-TEXT		Word 7		WordCorrect	
#Errors		320		124		148		133	
Typograph.	**Non-typogr.**	133	187	85	39	66	82	55	78
Number of Real Errors				85	35	57	41	45	45
#Total Real Errors				120		98		90	
Recall				37.5%		30,6%		28,1%	
Precision				96,77%		66,2%		67,6%	

by DGC - Desarrollo Gramatical Computarizado, from Barcelona.

Considering error coverage, the recall is concerned with how many of the errors occurring in the input text are identified as errors. Precision is concerned with what percentage of all the flags produced by the system are indeed real errors.

Results and Discussion

One can see in table 2 and 3 that for the time being, taking also into consideration the small set of implemented rules, CON-TEXT has a rather low recall rate (37–54%), contrasting with its precision rate (94–96%)[11]. The approach followed is a safe one, as mentioned above, and adopts linguistically motivated detection mechanisms. Thus the highest value for recall will be correlated to the performances permitted by the 'low-level' strategy adopted and to the amount and quality of available linguistic information. This approach shows complementary results to those provided by commercial tools. So for example, the Word 7 checking facility, in checking the texts form newspapers has a recall of 55% but a precision of only 40.12%. WordCorrect, on the contrary, has a higher recall (64.6%), but its precision is quite low (28.1%). It is interesting to note that while CON-TEXT maintains a compromise between recall and precision, commercial products show an unbalanced result wrt. this compromise: the lowest percentage in recall, the highest percentage in precision, and viceversa, as it can be check in table 2 and 3. On the other hand, the high number of errors detected by these products demons-trates that they apply the above mentioned «corruption methodology» which lead to the detection of false error situations.

That means that CON-TEXT, although with a very lim-

[11] The results in the tables below concerning CON-TEXT in the `UNIX` version are similar to the ones obtained with the `Windows95/NT` version.

ited set of rules, can provide reasonable results. Besides, detection of typographical errors is carried out with more accuracy than non-typographical errors. The number of agreement errors caught and the precision of this type of rules are both much lower than in the other type of errors[12]. This fact stresses that, although the results are promising, this strategy is insufficient for the treatment of more complex syntactic relations. In this sense, deeper linguistic annotation levels are needed which can be intertwined with similar, more abstract, error patterns.

6. Concluding remarks

We have presented tools for morpho-syntactic annotation and analysis for Spanish texts. Those tools combine wellknown and largely available tools and resources and proprietary technologies. The tool proposed could as well be further designed towards a workbench for the development of fast and robust NLP applications. Evaluation of disambiguation done on the base of the tool showed extremely satisfying results. Also very promising was the integration of a tool for form checking, which is still in the development phase. During the integration phase very interesting questions were raised about the interaction of morpho-syntactic annotation/analysis and form checking. Ongoing and future work will give some answers about the possible (balancing) interaction.

Further investigation and implementation will concern the integration of the integrated tool into a high-level language analysis. We think in particular on the use of the ALEP platform, since within this platform a syntax checker for Spanish has already been implemented, using among others the techniques of feature relaxation (see Ramírez Bustamante and Sánchez León, 1996b). The work on checking in that project was limited to error types whose detection needs a high-level analysis. We will have to further investigate to which extent error detection at the low level of linguistic analysis can support and speed the overall task of grammar checking.

7. Bibliography

Chanod, J.-P. and Tapanainen, P. Tagging French ─ comparing statistical and constraint-based methods. In *Proceedings of the EACL-95*, University College, Belfield, Dublin, Ireland, 1995.

Cutting, D., J. Kupiec, J. Pedersen and P. Sibun. A Practical Part-of-Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 133─140, Trento, Italy, 1992.

Kettunen, K. Low-level Typographical Spellchecking: A proposal. In *Computers and the Humanities*, 30, pages 77─84, 1996.

Oliva, K. Techniques for accelerating a grammar-checker. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 155-158, Washington, USA, 1997.

Petitpierre, D. and G. Russell. MMORPH ─ The MULTEXT Morphology Program. MULTEXT deliverable report for the task 2.3.1, ISSCO, University of Geneva, February 1995.

Ramírez Bustamante, F. and A. Honrado. *New insights into the ordered catalogue of grammar errors and style weaknesses in Spanish.* GramCheck Del 1, MLAP 93/11.

Ramírez Bustamante, F., F. Sánchez-León. Is linguistic Information enough for grammar checking? In *Proceedings of the First International Workshop on Controlled Language Applications*, CLAW '96, pages 216─228, 1996.

Ramírez Bustamante, F., F. Sánchez-León. GramCheck: A Grammar and Style Checker. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 175─181, 1996.

Ramírez Bustamante, F., F. Sánchez-León, T. Declerck. Grammar Checking and Preprocessing in ALEP? In *Proceedings of the 3rd ALEP User Group Workshop*, pages 71─80, 1997.

Sánchez-León, F. *Spanish tagset for the CRATER project*. CRATER internal document. Available as `http:xxx.lanl.gov/cmp-lg/ps/cmp-lg/9406023`.

Sánchez-León, F. and A. Nieto-Serrano. Retargeting a tagger. In R. Garside, G. Leech and T. McEnery (eds.) *Corpus Annotation. Linguistic Information from Computer Text Corpora*. London, Longman, 1997.

[12] The interaction between the different modules of the tool is critical, given that an incomplete implementation of the lexicon could provoke false detection of errors, since the guesser annotates unknown words with several tags.