CoGrOO: a Brazilian-Portuguese Grammar Checker based on the CETENFOLHA Corpus

Jorge Kinoshita[1], Laís do Nascimento Salvador[2], Carlos Eduardo Dantas de Menezes[3]

[1]Universidade da São Paulo (USP), Escola Politécnica
São Paulo – SP – Brasil
jorge.kinoshita@poli.usp.br
[2]Unifacs – Universidade de Salvador
Nuperc
Salvador – Bahia - Brasil
lais@unifacs.br
[3]Centro Universitário SENAC
São Paulo, SP – Brasil
carlos.edmenezes@sp.senac.br

Abstract

This paper describes an ongoing Portuguese Language grammar checker project, called CoGrOO[1]-Corretor Gramatical para OpenOffice (Grammar Checker for OpenOffice), based on CETENFOLHA, a Brazilian Portuguese morphosyntactic annotated Corpus. Two of its features are highlighted: - hybrid architecture, mixing rules and statistics; - free software project. This project aims at checking grammatical errors such as nominal and verbal agreement, "crase" (the coalescence of preposition "a" (to) + definitive singular determiner "a" yielding "à"), nominal and verbal government and other common errors in Brazilian Portuguese Language. We also present some empirical results based on the implemented techniques.

1.Introduction

This paper describes an ongoing Brazilian Portuguese Language grammar checker project, called *CoGrOO-Corretor Gramatical para OpenOffice* (Grammar Checker for OpenOffice), based on the CETENFOLHA corpus.

In general, written texts are subject to errors such as:

- Spelling errors;
- Grammatical Errors: when grammatical rules are not observed, as for example in "Nós vai para casa". ("We goes home"). These errors relate to verbal and nominal agreement.
- Errors of Style: In Portuguese, according to the general rule, the subject precedes the verb. Depending on the context, it is very difficult to understand an inversion. For example: "bonitos eles são" (pretty they are) is a correct sentence in Portuguese; but in a more formal written style the expected sentence would be "eles são bonitos" (they are pretty).
- Semantic errors: such errors are strongly context dependent. For example "the truck eats bananas".

CoGrOO project aims at checking grammatical errors such as nominal and verbal agreement, "crase" (the coalescence of preposition "a" (to) + definite feminine singular determiner "a", yielding "à"), nominal and verbal government, misuse of the adjective "mau" (bad) and the adverb "mal" (badly), among other common errors which can be found in Brazilian Portuguese. In this project, two features are highlighted: - hybrid architecture, mixing rules and statistics; - a free software project.

The CETENFOLHA (Linguateca, 2005) is a Brazilian Portuguese morphosyntactic annotated corpus, based on journalistic essays, generally written in third person and having a much more formal style than a personal letter. This article describes the construction of this grammar checker based on CETENFOLHA.

The remainder of this paper is structured as follows: Section 2 describes the architecture of the checker and the error detection process; in Section 3, some results are presented; Section 4, the conclusion, presents the contributions of the implemented approach.

2.Architecture

The architecture of CoGrOO with its main modules is described in Figure 1. As we can see, in CoGrOO system, consecutive modules accomplish the sentence analysis:

(1)The first module is the Sentence Boundary Detector: it splits up the input text into sentences.
(2)The second module, the Part of Speech Tagger, receives a sentence and assigns morphological tags to its lexical itens.
(3)After tagging, the sentence is submitted to the Chunker, in which finds small noun phrases and verbal phrases are separately grouped,
(4)The noun and verbal phrases are then submitted to the Grammatical Relation Finder, which assigns grammatical relations to noun and verbal phrases, trying to establish whenever possible, the grammatical roles involved in each case (for instance, subject, verb, predicate).

As can be observed, in Figure 1, there is an error detector module for each step of sentence analysis. The grammar checker looks for two error types: local errors and structural errors. The local error rules are applied to a short sequence of words and tags. For the treatment of more complex errors like verbal agreement errors, we

[1] This project is sponsored by the FINEP, Public Call MCT/FINEP/FINEP-01/2003.

apply structural error rules in the output of grammatical relation finder module. In the structural rules, a sequence of words, tags and grammatical labels is used.

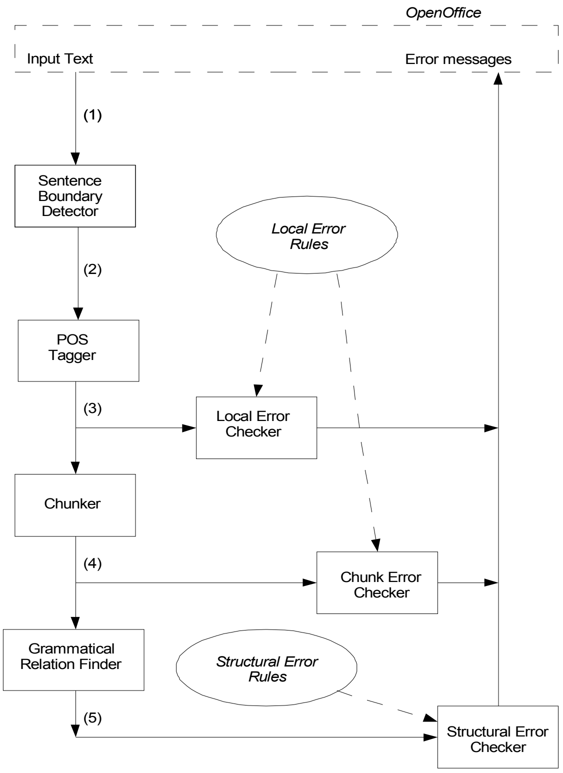

Figure 1 - Architecture of CoGrOO system

We used the corpus CETENFOLHA for the building of modules (2), (3) and (4). This corpus was automatically generated and thus, contains many errors. For instance: "Los" in "Los Angeles" is wrongly tagged as an object pronoun (example: "vou vê-LOS" / "I will see them") instead of Proper Noun. Despite the errors and style tendencies, we believe it was possible to extract many useful patterns for our modules.

In the following items we describe each system module with more details and how they accomplish the error-checking task.

2.1.Sentence Boundary Detector

This module prepares the input text to be analyzed by the Part-of-Speech Tagger. It marks sentence boundaries with special symbols. This module also consults a dictionary of abbreviations for dot disambiguation. Its output is a data structure for each sentence found in the text.

2.2.Part of Speech Tagger

This module assigns a morphological tag for each word of the sentence. The tagger follows the steps:

(1)assigns all the possible tags to each word of the sentence.

(2)defines the most probable tag for each word of the sentence, by inspecting its context.

Step (1) uses a dictionary that assigns possible tags for each word. This dictionary was generated by processing the CETENFOLHA annotated corpus. The process is in the following way: we counted how many times a word W was tagged as T. The most probable tag to word W is the one which appeared more times in the corpus. We also make use of a suffixes directory to handle words missing from the dictionary. The last 3 letters are searched for, and, in cases where the suffix cannot be found, we simply assigned the tag "singular noun" to the word, as this is the most recurrent tag in the Portuguese language.

In the second step we have to choose just one tag to each word based on an algorithm similar to Brill's tagger (Brill, 1992). For each word, the algorithm will replace the most probable tag by another tag, by inspecting the neighborhood, i.e., a sequence of of three tags (tag-trigrams). We have to choose a tag that most resemble the tag-trigrams from CETENFOLHA. We extracted the tag-trigrams from CETENFOLHA. We decided to use just the 80% more frequent trigrams. Just small patterns respond for the majority of the trigrams (Zipf's law).

After tagging words, the first set of error rules is applied. These rules are called local because they deal with a very short context of few words or tags to the left or to the right. A local rule consists of pattern and an error message. A pattern is a sequence of words or tags. If it is possible to find a pattern in the input text, then an error message is given.

Examples of error detection by this module is the use of the "crase" – which is the contraction between the preposition "a" (to) and the singular feminine definite article "a" (the) – before masculine words or verbs, or the use of the singular feminine inflection "–a" with the invariable adverb "meio".

An local rule has 2 components:

- the patern: a regular expression with words and part-of-speech tags;
- the message:
  - error type. Example: subject-verb agreement problem;
  - example of a correct sentence;
  - example of non adjusted use of the grammatical standard.

The tagger has a precision rate of 95%. It was built specially for this project, since we did not have access to an open source Brazilian Portuguese tagger.

2.3.Chunker

This module finds chunks, small parts of nominal and verbal phrases. Although our purpose is not to find whole Noun Phrases or Verb Phrases, we wrongly, called our chunks of NP and VP. The chunker is based on finite state machines that look for sequence of determiners, nouns, adjectives and pronouns for NP and sequence of verbs and adverbs for VP.

The chunker uses patterns extracted from the noun/verbal phrases in CETENFOLHA. These patterns were used to generate the finite state machines implemented in this module.

Some local error rules can also be applied to a chunk. In the case of NP, we check for number and genre agreement errors in determiners, noun and adjectives.

2.4.Grammatical Relation Finder

The objective of this module is to find grammatical relations such as subject-verb, verb-object or verb-preposition in the input sentence. So far, we have only implemented the code responsible for finding the subject-verb relation.

This is done by using a finite state machine that looks for patterns in the input sentence that is now annotated with tags and chunks. For instance, one pattern is "! NP VP", where "!" means "sentence beginning".

These patterns, composed of noun/verbal phrases and tags, were extracted from the CETENFOLHA. A pattern repeated many times indicates a valid grammatical relation. For instance: in the corpus when the pattern "! NP VP" occurs, NP is the subject of VP in 85% of cases.

After establishing grammatical relations, the structural error checker module can check many errors involving these relations, which are called structural errors. At the moment, only person and number agreement can be checked, based on syntactic tags between subject and verb. We hope to check for other errors such as misuse of prepositions as soon as other grammatical relation finding modules are created.

We also hope to check for errors that depend on the syntactic structure. Some local error rules are not so local as we previously thought. They must know syntactic structure. For instance: the verb "fazer" can indicate elapsed time. Example: "faz 20 anos" (it was 20 years ago). Generally, the verb "fazer" is in third person singular when followed by something that indicates time. We implemented this rule. It correctly detected an error in the sentence "fazem 20 anos", where the verb "fazer" is in third person plural. However, it wrongly detected an error in "Hoje eles fazem 20 anos de casados" (they made/fazem 20 years together). So, we reviewed this rule pattern in order to apply it only when the subject can not be detected.

3.Results

The grammar checker is implemented in the Perl language and the interface to the OpenOffice, in Java. In order to have a better evaluation of CoGrOO system, we created the Metrô corpus by collecting a data set from the site of the "Companhia do Metropolitano de São Paulo" (http://www.metro.sp.gov.br), a public transport company, known popularly as Metrô, in October 13th, 2005 (Uliano et al., 2006).

This corpus was created to evaluate how CoGrOO checker works on real texts and also to calibrate its overall performance by comparing CoGrOO to ReGra, a grammar and style checker for Brazilian Portuguese language (Nunes et al., 2000), that is used in Microsoft Office. In these experiments we used the version running in Microsoft Word 2000.

It is a corpus with 16,536 words and about 800 sentences; it's consisted of small pieces of texts, each one with about 4 paragraphs.

In this corpus, we have well written documents, probably analyzed by human revisers, since they had been published by an Internet content specialized agency. However, it was possible to find some mistakes in these texts. The reason for this is, probably, that no automatic revision tool was used, like CoGrOO or ReGra.

In this experiment, we use two parameters to evaluate the system performance:

- True positives: grammar errors correctly accused.
- False positives: accused grammar errors that do not exist.

In order to check these parameters, a human expert, a linguist, analyzed the corpus and detected 51 grammar errors. Moreover, we agree with 10 stylistic errors detected by ReGra (but there are 2 false positives too). Unlike Regra, CoGrOO system, as presented above, is purely a grammar checker system.

Table 1 shows the results of these experiments

GRAMMAR ERRORS	CoGrOO	ReGra
True positives	14	15
False positives	10	36

Table 1: Grammar errors detected by CoGrOO and ReGra

CoGrOO and ReGra had detected 7 common true positives (crase, agreement and punctuation errors). CoGrOO detected 8 true positives that Regra didn't detect, for instance, some nominal government errors. By the other hand, Regra system detected 7 true positives that were ignored by CoGrOO system., like some specific punctuation and agreement errors.

In a grammar checker system, each new rule can increase the amount of true positives detected, but it can also increase the amount of false positives, which is not desirable. By analyzing the messages emitted by ReGra, and also the correction suggestions, we can observe that it implements more rules than CoGrOO. Therefore, we achieved a better ratio between true and false positives than Regra in the Metrô corpus. CoGrOO has around 100 rules in its base.

For each grammar checker rule, we counted the number of true and false positives that it yielded when applied to the Metrô corpus. Rules with a low ratio were discarded or revised.

The stylistic errors module, yet to be done, will improve this ratio even more. We must confess that our rule set has been changing during Metrô corpus analysis due to that procedure of scoring rules, in order to select the best set of rules.

4.Conclusions and future work

CoGrOO project follows an hybrid architecture, mixing rules and statistics. Tagger, chunker and grammar relation finder modules were created in a slightly different fashion than what can be found in the literature. The tagger is based on previous works (Brill,1992; Daelemans et al, 1996; Menezes, 2000). The chunker and grammar relation finder are based on a finite state machine, but the patterns for the relation finder were statistically generated.

A similar approach to that of Naber's (2003) was adopted for local error rules, although Naber does not deal with sentence parsing, thus not being able to account for errors as subject-verb agreement.

Until now, we've been working with Metrô corpus, but we must apply CoGrOO to others corpora in order to evaluate our grammar checker.

Although we got a better ratio between true and false positives, we know that ReGra is better than CoGrOO, because it has a better Portuguese parser, a greater number of error rules and it also contains a stylistic error module.

We intend to implement a stylistic error module because these rules are easy to write (for example, to detect two consecutive spaces) and they have a good ratio between true and false positives.

We expect to make our code available around April, 2006. However, a prototype can be tested in http://cogroo.incubadora.fapesp.br.

Acknowledgements

We would like to thank to CoGrOO team members: Fábio M. Blanco, Fábio Wang Gusukuma, Marcelo Suzumura, Marcos Yoshio Okamura Oku, William Daniel Colen M. Silva and the linguist Sueli Caramello Uliano.

This research is supported by Finep - Financiadora de Estudos e Projetos -, Public Call MCT/FINEP/FINEP-01/2003. We also would like to thank the FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo - for hosting our source code (CVS) and site.

References

Brill, E. (1992) “A Simple Rule-Based Part Of Speech Tagger”, Proceedings of ANLP-92, 3rd Conference of Applied Natural Language Processing, Trento, Italy.

Daelemans, W., Zavrel, J., Berck, P. and Gillis, S. (1996) “MBT: A Memory-Based Part of Speech Tagger-Generator”, In: E. Ejerhed and I. Dagan (eds.) Proceedings of the Fourth Workshop on Very Large Corpora, Copenhagen, Denmark, 14-27.

Linguateca (2005) “CETENFolha”, http://www.linguateca.pt/CETENFolha/, last visited: March 2005.

Menezes, C.E.D. (2000) “Um método para a construção de analisadores morfológicos, aplicado à língua portuguesa, baseado em autômatos adaptativos”, Dissertação de Mestrado, Universidade de São Paulo, Brazil.

Naber, D. (2003) “A Rule-Based Style and Grammar Checker”, Diplomarbeit Technis Fakultät, Universität Bielefeld, Germany.

Nunes, M.G.V.; Martins,R.T.; Hasegawa, R.; Haber, R.R.; Montilha, G. “Relatório dos Testes Comparativos entre Diferentes Versões do Revisor Gramatical ReGra. (NILC-TR-00-8)”. June 2000, 08p. (available at http://www.nilc.icmc.usp.br/nilc/download/NILC00-8doc.zip)

Uliano, S.C.; Menezes, C.E.D.; Gusukuma, F.W. “Uma análise do CoGrOO, um Corretor Gramatical acoplável ao OpenOffice”. February 2006. (available at http://www.pcs.usp.br/~cogroo/papers/analise-cogroo-corpus-metro.html)