

# Corretor Gramatical Para o Emacs

Aluno: Thiago Maciel Batista (5749922)

Supervisor: Marcelo Finger

4 de junho de 2010

# Sumário

<b>1</b>	<b>Resumo da Monografia</b>	<b>3</b>
1.1	PLN . . . . .	3
1.2	CoGrOO . . . . .	3
1.3	Emacs usando o CoGrOO . . . . .	4
<b>2</b>	<b>Objetivos do Trabalho</b>	<b>4</b>
<b>3</b>	<b>Atividades já Realizadas</b>	<b>5</b>
<b>4</b>	<b>Cronograma de atividades</b>	<b>5</b>
<b>5</b>	<b>Estrutura esperada da monografia</b>	<b>6</b>

# 1 Resumo da Monografia

## 1.1 PLN

As linguagens normalmente compreendidas e manipuladas por computadores são formais e precisas, diferentemente de linguagens humanas, que possuem características que dificultam ao computador compreendê-las, como por exemplo presença de ambiguidade, dependência do contexto, características culturais etc. Processamento de linguagem natural (PLN) é uma área da computação que pesquisa e desenvolve algoritmos para processar uma linguagem natural, de modo a torná-la compreensível para um computador. Dessa forma, um computador será capaz de interpretar e gerar textos em linguagens naturais, tornando a comunicação homem-máquina o mais natural possível.

PLN é uma área em plena evolução e que ainda tem muito a evoluir, mas já existem muitas aplicações que utilizam os conceitos de PLN. Alguns exemplos da utilidade de PLN são, extração de informação, recuperação de informação, tradução automática, geração e interpretação de linguagem natural, resumos de textos e correção de textos.

## 1.2 CoGrOO

Um exemplo de um programa que utiliza PLN é o CoGrOO. Um corretor gramatical para a língua portuguesa do Brasil totalmente acoplável ao OpenOffice.org desenvolvido por alunos da Universidade de São Paulo. O CoGrOO utiliza análise sintática, morfológica e regras gramaticas para detectar erros em textos. O processo feito pelo CoGrOO para a análise gramatical de um texto é dividido em fases, que são executadas uma após a outra. Atualmente, o CoGrOO é composto por módulos, sendo que, cada módulo realiza uma fase do processo. Abaixo temos os módulos do CoGrOO.

- SentenceDetect - Divide o texto em sentenças
- Tokenizer - Identifica os tokens.
- ProperNameFinder - Identifica nomes próprios no texto.
- ContractionSparator - Gera uma lista de tokens e suas contrações.

- Pretagger - Gera uma lista de CogrooTokens com nomes próprios e contrações
- Postagger - Devolve uma lista de etiquetas morfológicas associadas aos Cogrootokens.
- Rulesapplier - Aplica as regras gramaticais à lista de CoGrOOTokens e gera uma lista de erros

O CoGrOO está em evolução e atualmente está sendo desenvolvido o CoGrOO UIMA, uma nova versão do corretor que utiliza a arquitetura padronizada do Apache UIMA, um aplicativo para manipulação de dados não estruturados.

### 1.3 Emacs usando o CoGrOO

Apesar de ser feito, a princípio, para o OpenOffice o CoGrOO pode ser acoplado a outros aplicativos. Aproveitando essa mobilidade do CoGrOO, vamos acoplá-lo ao editor de texto Emacs que, apesar de ser um editor de texto popular, ainda não possui um corretor para a língua portuguesa acoplado a ele.

## 2 Objetivos do Trabalho

Este trabalho tem como objetivo estudar formas de acoplar o corretor gramatical CoGrOO ao editor de textos Emacs. Usaremos duas versões do CoGrOO, a última versão do sem utilizar o UIMA e o CoGrOO UIMA, com a finalidade de testar qual versão se adapta melhor ao projeto. Ao final do acoplamento, teremos o Emacs usando o CoGrOO para analisar e encontrar erros em textos digitados pelos usuários.

O CoGrOO processará um trecho de um texto enviado pelo Emacs a procura de erros e devolverá ao Emacs os erros encontrados e sugestões para correção. O emacs poderá usar os erros apontados pelo CoGrOO para mostrar ao usuário onde ocorreram erros no texto, de forma que o usuário possa identificá-los e arrumá-los. Para isso, o Emacs destacará trechos do texto onde foram detectados erros.

Após finalizarmos a parte de acoplar o CoGrOO ao Emacs, temos a intenção de disponibilizar o aplicativo para os usuários. Além disso, pretendemos

usar opiniões e sugestões, em relação a erros encontrados no aplicativo e possíveis melhorias, para proporcionar uma evolução no projeto de acordo com as necessidades dos usuários.

### 3 Atividades já Realizadas

Até agora, as seguintes tarefas foram realizadas:

- Estudo da estrutura interna do Emacs.
- Estudo da linguagem de programação Emacs Lisp.
- Estudo de conceitos básicos de PLN.
- Tentativa de fazer o Emacs Lisp se comunicar com o CoGrOO.
- Estudo do JLinjer.
- Estudo do Apache UIMA.
- Estudo do CoGrOO.

### 4 Cronograma de atividades

- Junho-Julho de 2010 - Terminar de acoplar o CoGrOO ao Emacs.
- Junho-Julho de 2010 - Testar o Emacs com duas versões diferentes do CoGrOO, a última versão do CoGrOO sem usar UIMA e versão usando o anotador Apache UIMA.
- Junho-Julho de 2010 - Disponibilizar o corretor para os usuários.
- Junho-Novembro de 2010 - Utilizar opiniões dos usuários para melhorar o aplicativo.
- Junho-Novembro de 2010 - Estudar mais sobre o Emacs e Emacs Lisp.
- Junho-Novembro de 2010 - Estudar conceitos e algoritmos de PLN.
- Setembro-Outubro de 2010 - Preparar a apresentação.
- Junho-Novembro de 2010 - Elaboração/Escreva da monografia.

## 5 Estrutura esperada da monografia

- Definição de PLN.
- Utilidades de PLN.
- Como utilizar PLN para corrigir textos.
- Algoritmos de PLN pertinentes ao projeto.
- Descrição do corretor gramatical CoGrOO.
- Tecnologias estudadas e aplicações das mesmas no projeto.
- Como foi feito o acoplamento do CoGrOO ao Emacs.
- Como o Emacs utiliza o corretor para fazer a correção de textos.
- Erros encontrados por usuários.
- Sugestões de usuários para melhorias.
- Estado atual do aplicativo com melhorias sugeridas, erros corrigidos e possíveis erros que permanecem.
- Resultados de testes.
- Conclusões.

## Referências

- [1] Robert J. Chassell. An introduction to programming in emacs lisp. Disponível em: [http://www.gnu.org/software/emacs/emacs-lisp-intro/html\\_node/index.html](http://www.gnu.org/software/emacs/emacs-lisp-intro/html_node/index.html)., Acesso em : 2010.
- [2] William Colen. Cogroo corretor gramatical para openoffice.org-guia do desenvolvedor. Disponível em: <http://csl.ime.usp.br/cogroo/en/node/5>., Acesso em: 2010.
- [3] Apache UIMA Development Community. Apache uima documentation. Disponível em: <http://uima.apache.org/documentation.html>., Acesso em: 2010.

- [4] Robert Krawitz, Bil Lewis, Dan LaLiberte, Richard M. Stallman, and Chris Welty. Gnu emacs lisp reference manual. Disponível em: [http://www.gnu.org/software/emacs/manual/html\\_node/elisp/index.html](http://www.gnu.org/software/emacs/manual/html_node/elisp/index.html)., Acesso em : 2010.
- [5] Xah Lee. Xah's emacs lisp tutorial. Disponível em: <http://xahlee.org/emacs/elisp.html>., Acesso em: 2010.
- [6] Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. MIT press, second edition, 1999.