CoGrOO
Um sistema de análise gramatical para a língua portuguesa

William D. Colen M. Silva
Eng. Computação (Poli – 2006)

Desenvolvedor CoGrOO – 2004 - presente

william.colen@gmail.com

CoGrOO
Um sistema de análise gramatical para a língua portuguesa

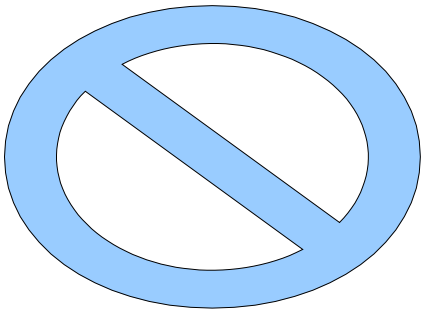
- Agenda
  - Breve motivação sobre o campo PLN
  - Apresentação do CoGrOO
  - História do projeto
  - Breve análise dos componentes
  - Desempenho dos componentes
  - Demonstrações
  - Evolução e propostas

Aplicações PLN

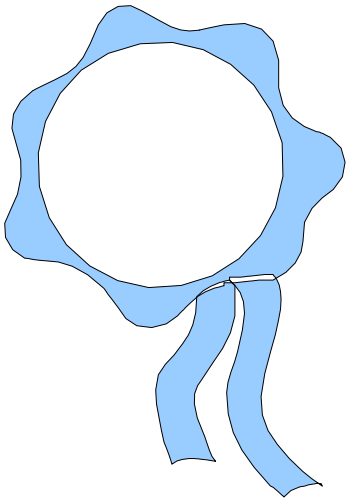
- Tradutores automáticos
- Corretores ortográficos e gramaticais
- Buscadores (Web)
- Ferramentas para Web Semântica
- Indexadores para BI
- Sugestões de compras
- ...

Busca na WEB

Busca por palavra chave



Qual o significado de “consensuar”?
Quando nasceu Napoleão?
Quantas toneladas de café foram produzidas em 1901?
O que os cientistas pensam quanto a ética de experiencias com células tronco?



Boa parte do conhecimento humano está em documentos difíceis de serem interpretados por computadores

Informação:
- Alto valor
- Mais atualizada

Mas..
- Muito ruido
- Semântica oculta
- Busca ineficiente

Apresentação do CoGrOO

- Primeiro
- Único
- Mais de 26 mil downloads (contando apenas da versão 2.0 em diante)
- Usado por empresas estatais e privadas. Algumas empresas tem ele instalado em milhares de máquinas.
- Reconhecido localmente como sendo um dos esforços mais importantes para o desenvolvimento do BrOffice.org

Apresentação do CoGrOO

- Foi o primeiro corretor gramatical integrado ao OOo do mundo.
- Segundo mais utilizado (perde apenas para o Language Tool, que suporta inglês).
- Reconhecido pela Linguateca.
- O projeto Golfiño (Galego) foi criado a partir do CoGrOO.
- Recebemos pedidos para criar versões do CogrOO para português de Portugal e para o Espanhol.

Apresentação do CoGrOO

- Hospedado pelo SourceForge
- Licença LGPL
- Fácil instalação e uso
- Atualizações frequentes
- Apoio da comunidade

História do projeto

- 2004
  - Idealização do projeto (Menezes, Kinoshita, Lais e Neto)
  - Edital de Software Livre
  - Apoio da FINEP e Metrô de S. Paulo
- 2005
  - Desenvolvimento de diversos protótipos e das regras

História do projeto

- 2006
  - Integração com o BrOffice.org
  - CoGrOO 1.0
  - Apoio Google SoC: desenvolvimento API do corretor gramatical no Core do OOo
  - Projeto de conclusão de curso – nasce o futuro CoGrOO 2.0

História do projeto

- 2007
  - Lançamento do CoGrOO 2.0
  - Google SoC – integração com o Abiword
  - Novos projetos de conclusão de curso: porte para inglês e espanhol
  - Google doa uma licença do Treebank

História do projeto

- 2008
  - Projeto carente de colaboradores e de apoio
  - Esforços individuais:
    - Estudos para uma nova arquitetura
    - Renovação para suportar o OOo 3.0

História do projeto

- 2009
  - Grande revisão e correção de bugs
  - Aceitação cada vez maior do projeto pela comunidade contrasta com a falta de apoio
  - Inicio das conversas com o CCSL

CoGrOO 1.0

Características

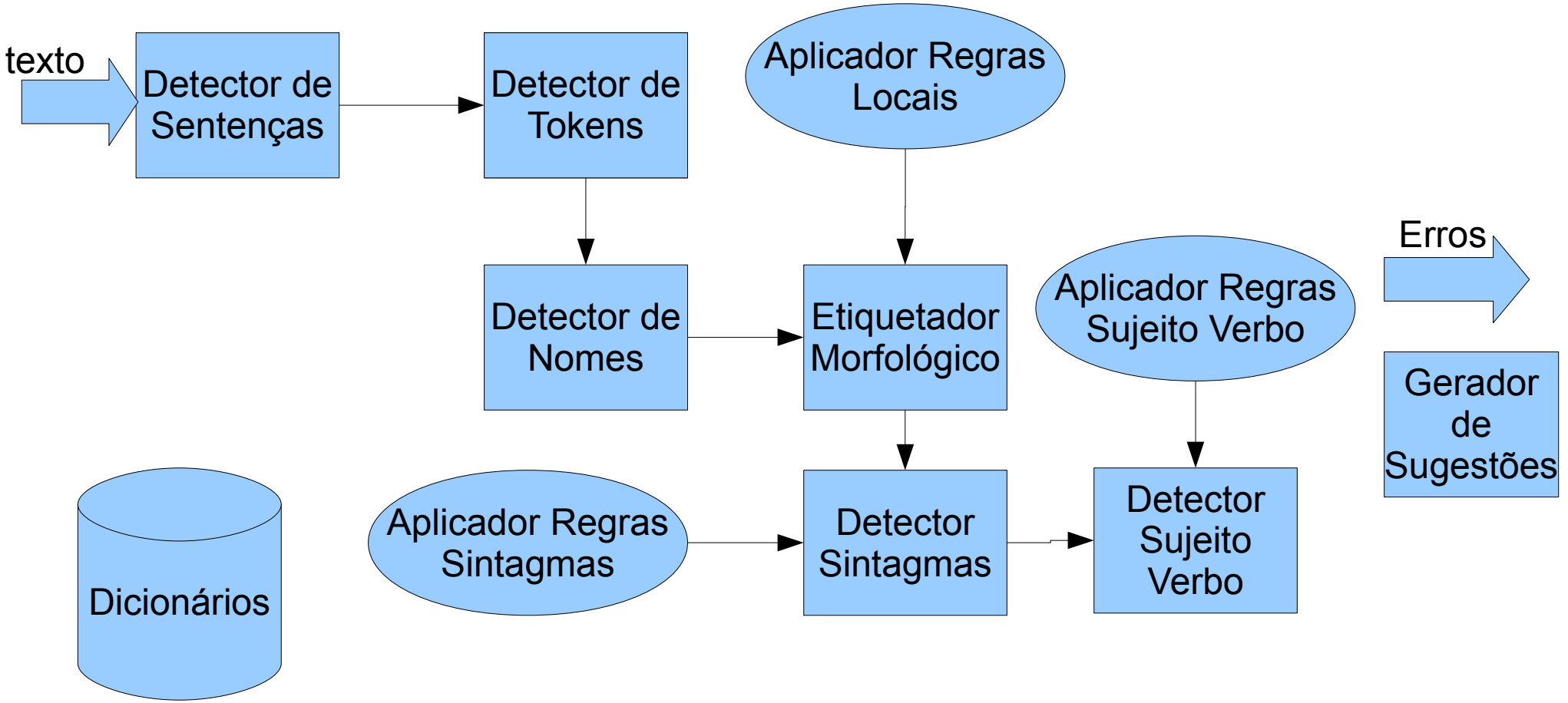
- Desenvolvido em Perl
- Treinamento - Corpus CentenFolha 1.0 modificado
- Etiquetador morfológico: método Trigrama
- Chunker e Shallow Parser: método N-Grama
- Detector de sentenças e de palavras: identificação de padrões e outras heurísticas
- Totalmente desenvolvido pela equipe (não usa bibliotecas terceiras)

CoGrOO 2.0

Características

- Desenvolvido em Java
- Treinamento - Corpus CentenFolha 1.0 modificado (mesmo do CoGrOO 1.0)
- Todos os módulos usam o método da Máxima Entropia
  - Heurísticas adotadas no CoGrOO 1.0 foram transformadas em *features* no CoGrOO 2.0
- Usa bibliotecas terceiras: OpenNLP, Maxent, Stemplator

Análise da Arquitetura e do Desenvolvimento



Dicionários

- Dicionários de palavras com classificação morfológica
  - casa: [verbo casar] [substantivo feminino singular]
- Dicionários de relacionamentos entre palavras
  - meninas → menino → menino meninos menina meninas
- Dicionário de abreviaturas

Problema Fundamental: Resolver ambiguidades

- Detecção de limites de palavras/sentenças
  - “Sr. Silva estava jogando futebol.”
  - “O computador novo custará R$ 2.500,00.”
- Ambigüidades nos sentidos das palavras
  - “Nada como voltar para *casa*!” (substantivo)
  - “Ele se *casa* na semana que vem.” (verbo)

Separador de Sentenças

- Entrada:
  - [Ele foi procurar uma casa. Ele vai se casar com a Srta. Maria.]
- Saída:
  - [Ele foi procurar uma casa.]
  - [Ele vai se casar com a Srta. Maria.]
- Desafio:
  - Decidir se marcas de fim de linha estão separando linhas no contexto. Exemplo "Srta."

Separador de Tokens

- Entrada:
  - [A Sra. Maria, esposa do Sr. José, trouxe-nos frutas.]
- Saída:
  - [A][Sra.][Maria][,][esposa][do][Sr.][José][,][trouxe][-nos][frutas][.]
- Desafio
  - Além dos espaços muitos outros símbolos podem separar *tokens* na frase. Exemplo "José, trouxe-nos" são quatro *tokens*.

Etiquetador Morfológico

- Entrada:
  - [Ele foi procurar uma casa.]
- Saída:
  - [Ele, pronome pessoal masculino 3ª pessoa singular]
  - [**foi**, verbo ir passado 3ª pessoa do singular]
  - [procurar, verbo procurar no infinitivo]
  - [**uma**, artigo indefinido feminino singular]
  - [**casa**, substantivo feminino singular]
  - [., ponto final]
- Desafio
  - Muitas palavras de mesma grafia podem ser classificadas de diferentes formas de acordo com o contexto em que estão. Por exemplo "casa", que pode ser substantivo ou verbo (casar).

Agrupador

- Entrada:
  - [Ele, pronome pessoal masculino 3ª pessoa singular]
  - [foi, verbo ir passado 3ª pessoa do singular]
  - [procurar, verbo procurar no infinitivo]
  - [uma, artigo indefinido feminino singular]
  - [casa, substantivo feminino singular]
  - [., ponto final]
- Saída:
  - [Ele, sintagma nominal masculino 3ª pessoa singular]
  - [foi procurar, sintagma verbal 3ª pessoa singular]
  - [uma casa, sintagma nominal feminino 3ª pessoa singular]
  - [., ponto final]
- Desafio
  - Encontrar seqüências que poderiam ser tratadas como elemento único. Exemplo "uma casa".

Analisador Sintático Simples

- Entrada:
  - [Ele, sintagma nominal masculino 3ª pessoa singular]
  - [foi procurar, sintagma verbal 3ª pessoa singular]
  - [uma casa, sintagma masculino feminino 3ª pessoa singular]
  - [., ponto final]
- Saída:
  - [Ele, sujeito]
  - [foi procurar, verbo]
  - [uma casa, sintagma nominal feminino 3ª pessoa singular]
  - [., ponto final]
- Desafio
  - Identificar entre os sintagmas quais compõem sujeito e verbo.

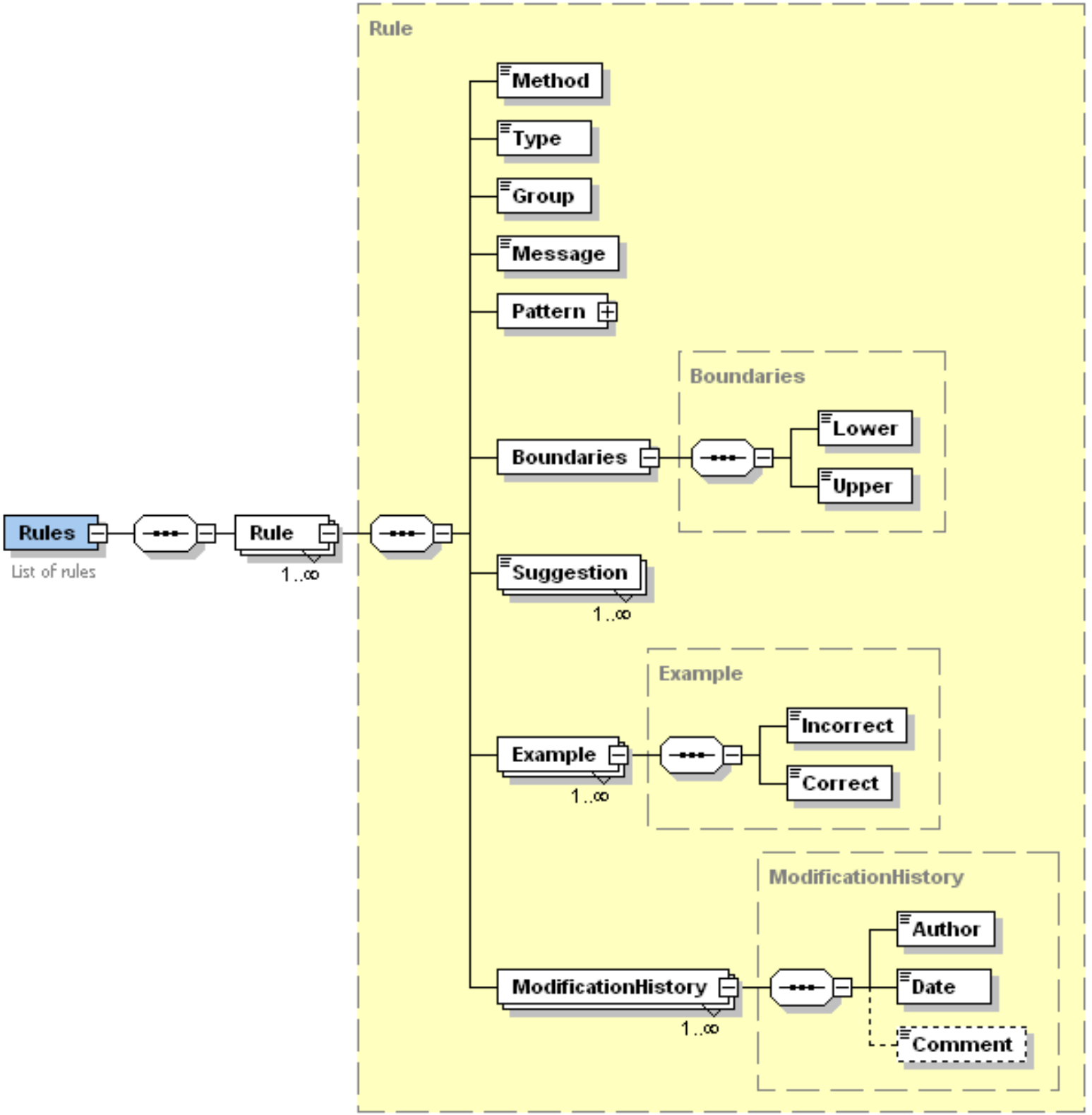
Análise da Arquitetura e do Desenvolvimento

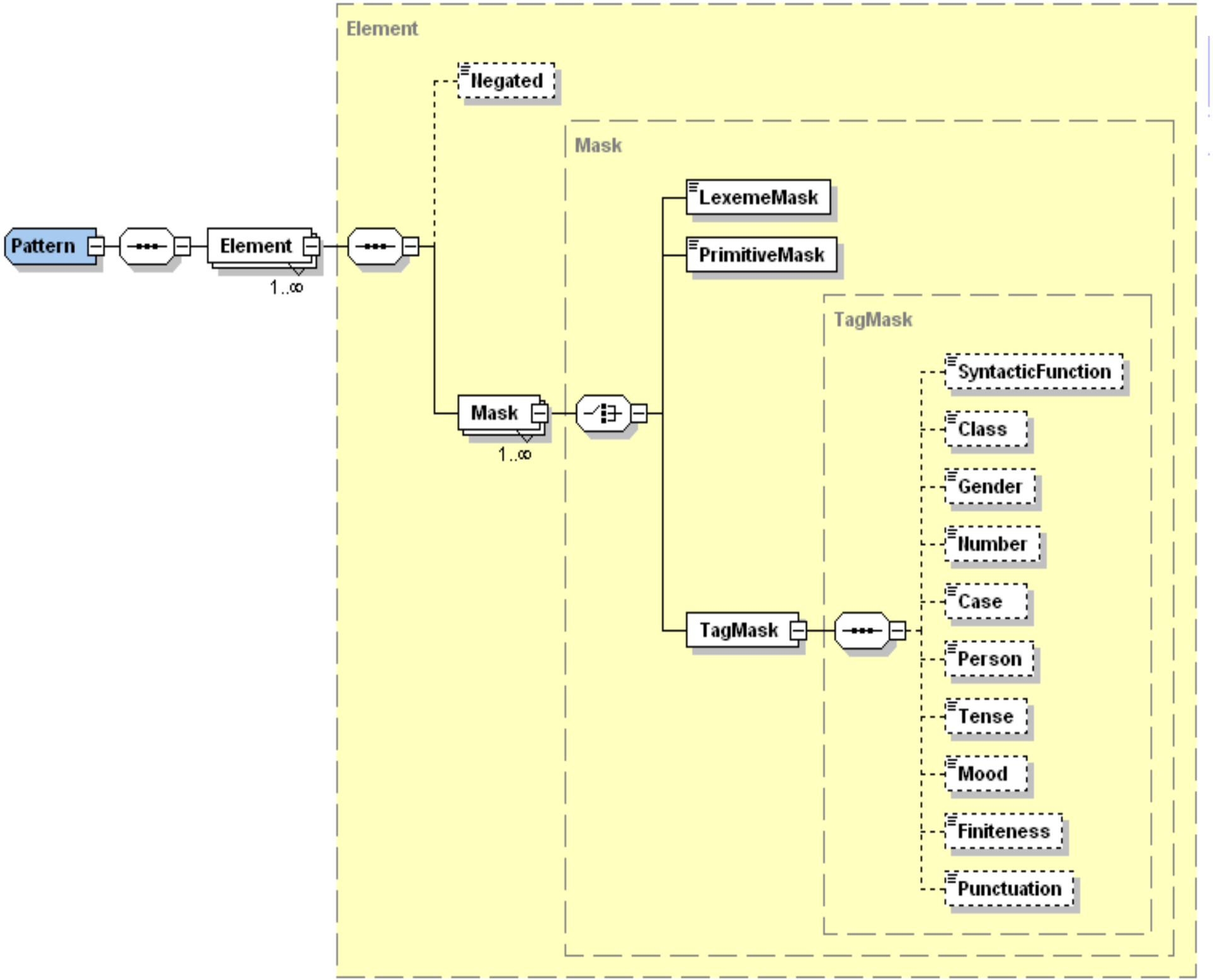
Tipos de erros:

- colocação pronominal
- concordância nominal
- concordância entre sujeito e verbo
- concordância verbal
- uso de crase
- erros comuns da língua portuguesa falada

Regras

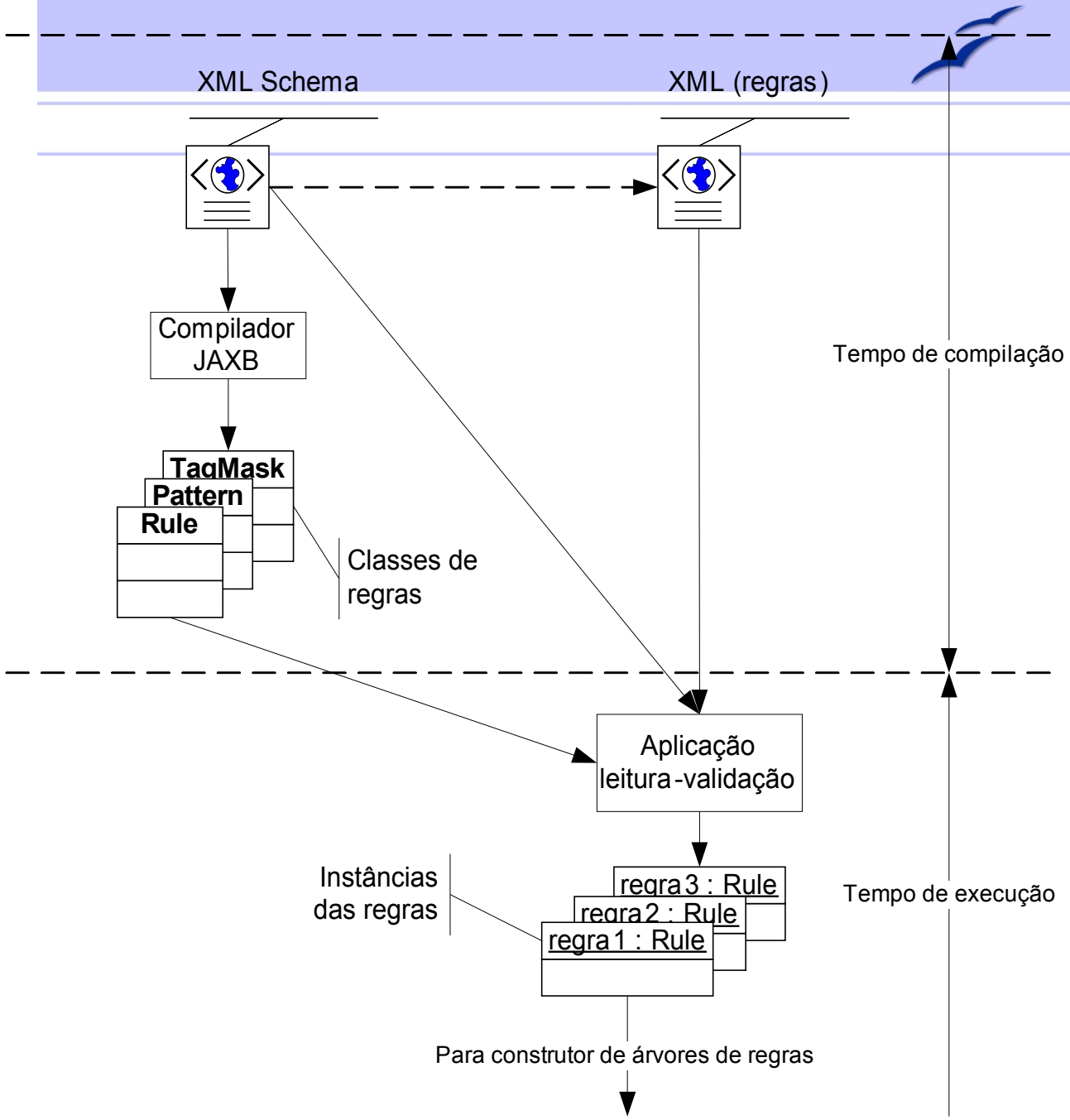
- Análise de desvios por padrões
- Estrutura de regra
  - Método
  - Mensagem
  - Padrão
    - Exemplo: artigo masculino plural + substantivo masculino singular
  - Modelos genéricos de sugestão
- Descritos em arquivo XML e validados por um XSD

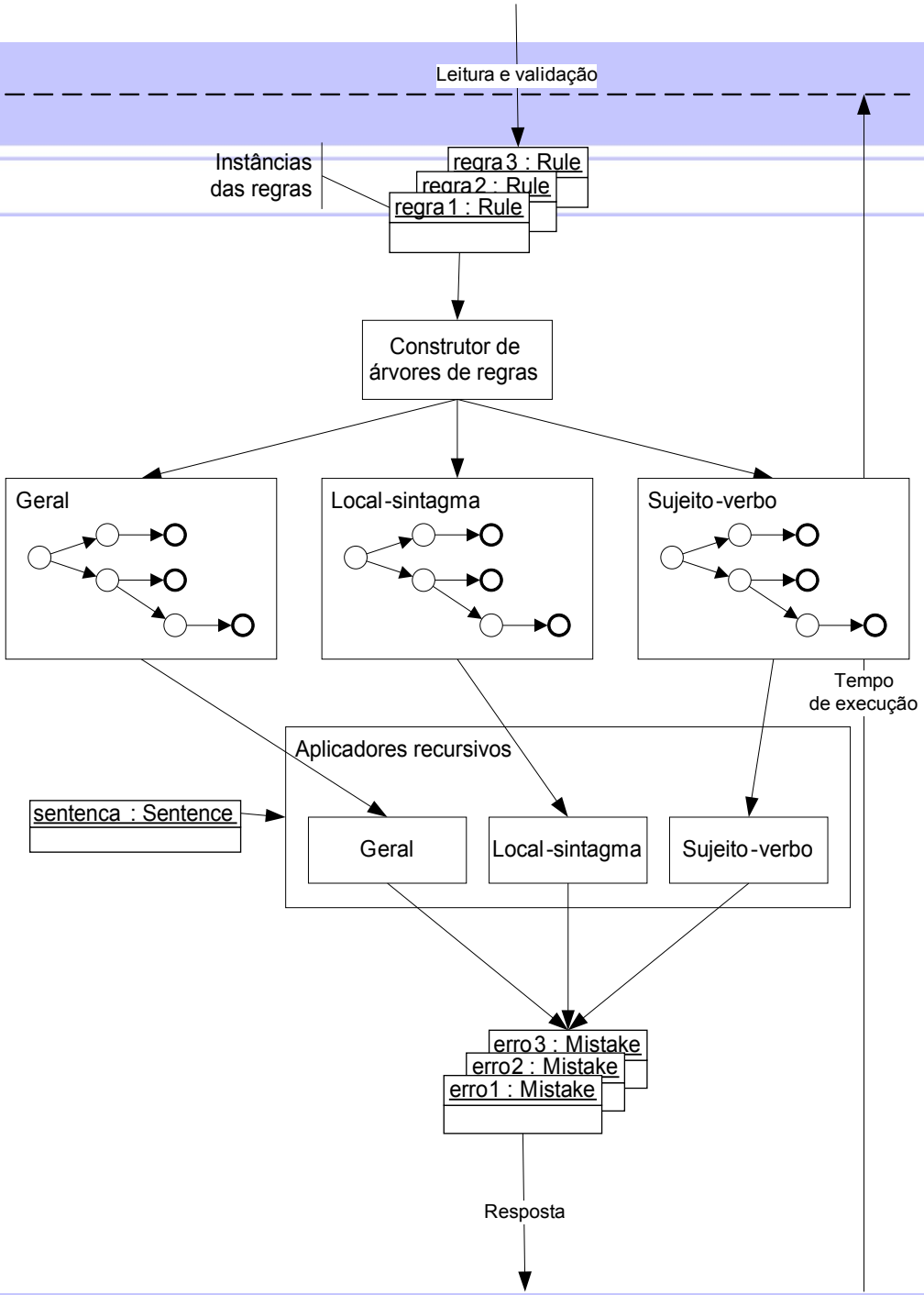

Rule
Method
Type
Group
Message
Pattern
Boundaries
Lower
Upper
Boundaries
Rules
List of rules
Rule
1..∞
Suggestion
1..∞
Example
Incorrect
Correct
Example
1..∞
ModificationHistory
Author
Date
Comment
ModificationHistory
1..∞


Element
Negated
Mask
LexemeMask
PrimitiveMask
TagMask
SyntacticFunction
Class
Gender
Number
Case
Person
Tense
Mood
Finiteness
Punctuation
Pattern
Element
1..∞
Mask
1..∞
TagMask

Regras

- Árvores
  - Criação de árvores de busca a partir dos padrões das regras
- Aplicadores
  - São algoritmos recursivos que fazem a busca com base nas árvores e na sentença processada pelo CoGrOO


XML Schema
XML (regras)
Compilador
JAXB
TagMask
Pattern
Rule
Classes de
regras
Tempo de compilação
Aplicação
leitura-validação
Instâncias
das regras
regra3 : Rule
regra2 : Rule
regra1 : Rule
Tempo de execução
Para construtor de árvores de regras

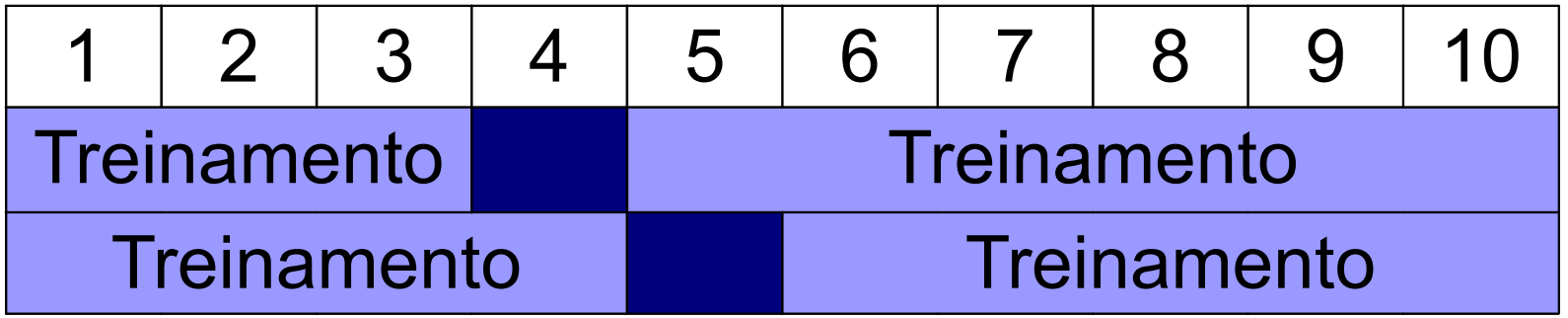

Leitura e validação
Instâncias
das regras
regra 3 : Rule
regra 2 : Rule
regra 1 : Rule
Construtor de
árvores de regras
Geral
Local-sintagma
Sujeito-verbo
Tempo
de execução
Aplicadores recursivos
sentenca : Sentence
Geral
Local-sintagma
Sujeito-verbo
erro 3 : Mistake
erro 2 : Mistake
erro 1 : Mistake
Resposta

Sistemas Desenvolvidos

- Treinamento
- Refinamento dos parâmetros de treinamento
- Teste de desempenho
- Teste das regras
- Visualizador gráfico
- Servidor RPC
- Servidor XML
- Integração com o OpenOffice.org

Desempenho

Testes de Desempenho



Desempenho

- Tokenizer - 98,74%
  - Considera a sentença
- Name Finder – 90,11%
  - Considera a sentença
- Tagger – 96,05%
  - Considerado cada token
- Chunker – 77,25%
  - Considera a sentença
- Shallow Parser – 68,80%
  - Considera a sentença

Demonstração

- Visualizador
- Saída dos testes
- Teste de regras
- Mostrar arquivo de regras
- BrOffice.org

O Futuro do CoGrOO

Existe muito ainda para ser explorado...

Comunidade (Colaboradores Web)

- Página que possibilita experimentar o CoGrOO e seus módulos pela Web
- Página que possibilita escrever e testar regras online – regras poderiam ser submetidas para a equipe avaliar
- Página que aplica o corretor sobre textos extraídos do Wikipédia – interface permitiria que o colaborador determinasse a causa do erro (dicionário, etiquetador)
- Página que permite entrar com texto livre para cadastrar mal funcionamento do corretor

CoGrOO como ferramenta Linguística

- Determinar casos em que os módulos do corretor poderiam ser úteis para pesquisadores da área de linguística
- Criar ferramentas linguísticas para pesquisadores

Módulos e dicionário

- Reescrever módulos que apresentam baixa performance
- Implementar módulos de resolução de correferências e análise semântica
- Revisar o dicionário léxico

Corretor Gramatical

- Tratar casos de baixa performance da análise (possível erro)
- Criar corpus de erros e treinar um módulo para detectar esses erros
- Ampliar as regras

UIMA

Apache UIMA (Unstructured Information Management Architecture)

- UIMA é uma plataforma para construção de aplicações que lidam com Linguagens Naturais. É um esforço para padronizar análise de conteúdo, sejam textos contidos em emails, blogues, páginas Web, ou até mesmo em imagens e vídeo
- Oferece ferramentas para facilitar o desenvolvimento de aplicações e interoperabilidade
- Adotado como padrão OASIS

Criar uma biblioteca de componentes para o UIMA, todos os módulos do CoGrOO se tornariam automaticamente reusáveis

Discussão....

Bibliografia

W.D.C.M. Silva, M. Suzumura, F.W. Gusukuma, D.A.M. PIRES, “Corretor Gramatical Acoplável ao OpenOffice.org - CoGrOO 2.0”, Monografia de conclusão do curso de Engenharia da Computação, Escola Politécnica, Universidade de São Paulo, Brasil, 2006.

OpenNLP, open-source framework to develop natural language applications (http://opennlp.sourceforge.net, Acesso em: 25 de março 2009.).

UIMA, Unstructured Information Management Architecture (http://incubator.apache.org, Acesso em: 25 de março 2009.).

OASIS, Open Standards for the Information Society (http://www.oasis-open.org, Acesso em: 25 de março 2009.).

KINOSHITA, J. ; SALVADOR, L. N. ; MENEZES, C. E. D. ; SILVA, W. D. C. . CoGrOO an OpenOffice grammar checker. In: Workshop on Intelligent Text Categorization and Clustering, 2007, Rio de Janeiro. Anals of 7th International Conference on Intelligent Systems Design and Applications , ISDA 2007. Rio de Janeiro, 2007.