CENTRO UNIVERSITÁRIO FEEVALE
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

NLPSearch
Framework para integração de Sistemas de PLN e API's de Mecanismos de Busca

por

LEANDRO FUSSIGER
leandro.fussiger@gmail.com

Anteprojeto de Trabalho de Conclusão

Rodrigo Rafael Villarreal Goulart
rodrigo@feevale.br

Novo Hamburgo, Outubro de 2006.

Sumário

Dados de Identificação

Área de Estudo: Inteligência Artificial e Processamento da Linguagem Natural.

Título provisório do trabalho: NLPSearch - Framework para Integração de Sistemas de PLN e API's de Mecanismos de Busca

Orientador (a): Ms. Rodrigo Rafael Villarreal Goulart

Identificação do aluno:

Nome: Leandro Fussiger

Telefones:

Celular: 9135-7210

Residencial: 3527-0647

Comercial: 3598-9801 Ramal 2112

E-mail: leandro.fusiger@gmail.com

Resumo

Nas últimas décadas os estudos relacionados com o processamento da linguagem natural têm se desenvolvido a passos largos. Os pesquisadores têm focado seus esforços em diferentes fenômenos da linguagem, dentre os quais se destacam os estudos realizados sobre anáforas, sumarização automática e o tratamento de diálogos.

Uma forma de integrar alguns destes conceitos é o desenvolvimento de mecanismos para o tratamento desses fenômenos da linguagem. Estes mecanismos geralmente utilizam alguma coleção de textos para o seu processamento, intitulados Corpus.

Para montar um corpus é necessário reunir diversos textos sobre o assunto escolhido, o que acaba consumindo um tempo considerável. Para realizar esta tarefa podemos utilizar a larga base de dados armazenada na web. Para isto precisamos utilizar algumas API's (Application Programming Interface) disponibilizadas por alguns mecanismos de busca, como o Google, Yahoo e Technorati.

Para utilizar os conceitos de PLN aliados com as API's dos mecanismos de busca propõem-se a construção de um framework para desenvolvimento de ferramentas para PLN que utilizem a web como corpus.

Sendo assim este trabalho tem por objetivo o estudo destas tecnologias (PLN e API's) para a construção de um framework que contemple esta necessidade. E posteriormente o desenvolvimento de ferramentas através deste framework.

Palavras Chave: PLN, API's, Framework.

Motivação

A investigação cientifica de métodos e técnicas para o processamento da linguagem natural têm se desenvolvido muito nas últimas décadas, e especialmente nos últimos anos, com esforços focados em diferentes fenômenos da linguagem. O processamento de anáforas (GASPERIN, 2003), sumarização automática (PARDO, 2003) e o tratamento de diálogos (NUNES, 1999) são alguns exemplos do emprego de recursos computacionais e o conhecimento de lingüistas na construção de sistemas inteligentes que consigam tratar de forma individualizada os fenômenos da linguagem. Surge então a necessidade de desenvolver mecanismos para integração desses esforços, aliados a uma larga base de dados como a internet. Para propor um sistema que integre estes conceitos de forma a ampliar sua utilização em diferentes contextos lingüísticos, este trabalho propõe a construção de um framework para o desenvolvimento de aplicações de PLN que utilizem a Web como corpus.

Na década de 50 a comunidade cientifica começava com os primeiros estudos sobre PLN. Segundo Nunes (1999, p. 7) as primeiras investigações institucionalizadas sobre o Processamento da Linguagem Natural começaram a ser desenvolvidas no início da década de 50, depois da distribuição de 200 cópias de uma carta, conhecida como *Weaver Memorandum*, escrita por Warren Weaver, então vice-presidente da Fundação Rockfeller e exímio conhecedor dos trabalhos sobre criptografia computacional. Nessa carta, divulgada em 1949, Weaver convidava universidades e empresas, interessados potenciais, para desenvolver projetos sobre um novo campo de pesquisa que ficou conhecido como ‘‘tradução automática’’, ‘‘tradução mecanizada’’ ou simplesmente MT (abreviação do inglês “Machine Translation”).

Em 1960 foram desenvolvidas as primeiras formalizações do significado em termos de redes semânticas, os primeiros tratamentos computacionais das gramáticas livres de contexto e a criação dos primeiros analisadores. Na década de 70 veio à consolidação dos estudos do PLN com a implementação de parcelas das primeiras gramáticas e analisadores sintáticos e a busca de formalização de fatores pragmáticos e discursivos. Em 1980 começava a sofisticação dos sistemas com o desenvolvimento de novas teorias lingüísticas motivadas pelo estudo do PLN. Nos anos 90 surgiam os

sistemas baseados em representação do conhecimento com o desenvolvimento de projetos de sistemas de PLN complexos que buscavam a integração dos vários tipos de conhecimentos lingüísticos e extralingüísticos e das estratégias de inferência envolvidas nos processos de produção, manipulação e interpretação de objetos lingüísticos (Nunes, 1999).

Hoje em dia encontramos ferramentas que implementam recursos de PLN, elas podem ser classificadas em:

- Tagger, Morphological Analyzer - Analisadores morfológicos e etiquetadores;
- Stemmer - Forma canônica da palavra. Ex.: A forma canônica de gatos e gatas é gato;
- Parsers - Analisadores sintáticos;
- Corpus Tools - Ferramentas para Manipulação de Corpus (coleção de textos sobre um determinado assunto cujo conteúdo possui etiquetas).

As ferramentas de linguagem da Xerox Research Centre Europe (XEROX, 2006) são exemplos na categoria Tagger, Morphological Analyzer, através delas é possível escolher dentre diversas línguas para realizar a análise morfológica ou a tokenização, o resultado é apresentado na própria página do centro de pesquisa. O Snowball (SNOWBALL, 2006) é um Stemmer que possui módulos em várias línguas, sendo uma delas o português. Através de vocabulários ele processa as palavras informadas e mostra as palavras na sua forma canônica. Na categoria de Parsers em língua portuguesa destaca-se o Portuguese VISL (VISUAL, 2006). Neste parser o usuário informa a frase e solicita que o programa realize o parsing ou analise sintática da frase, então é apresentada uma janela com o resultado, conforme figura 1, onde cada palavra é classificada conforme sua sintaxe. O JBootCat (ROBERTS, 2006) é uma ferramenta que se enquadra na categoria de Corpus Tools. Através desta ferramenta o usuário informa o caminho onde será salvo o corpus, as palavras que deseja encontrar, o motor de busca (somente o Google por enquanto) e solicita que a ferramenta busque os resultados na web. Com os resultados obtidos o usuário poderá gerar um corpus.

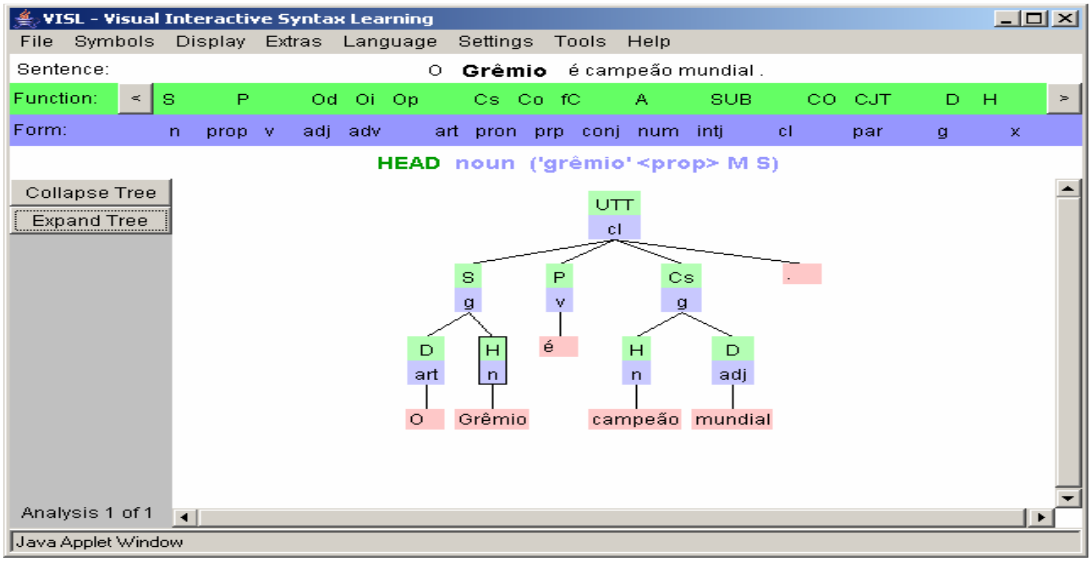

Figura 1 – Parser VISL

Para montar um corpus que utilize a web como base de dados é necessário utilizar API's de motores de busca. O Google e o Yahoo que possuem as maiores bases de dados para pesquisas na web disponibilizam suas API's para desenvolvimento de ferramentas, assim como a Technorati que possui a maior base de blogs cadastrados na web.

Com a API do Google (GOOGLE, 2006) é possível desenvolver ferramentas para realizar pesquisas na sua base de dados. Para isso é necessário obter uma chave de pesquisa na página do Google através do registro de uma conta neste serviço. Este serviço possui algumas limitações tais como a possibilidade de realizar somente 1000 buscas por dia, 10 palavras por busca e 10 resultados por busca. Os resultados obtidos na busca são devolvidos para a ferramenta no formato XML.

A API do Yahoo (YAHOO, 2006) segue os mesmos conceitos da anterior, como a necessidade de um cadastro no site para receber a chave de pesquisa, porém os números quanto a pesquisas são consideravelmente maiores. É possível realizar até 5000 buscas por dia com até 50 resultados por busca. Os resultados são retornados através de um formulários XML ou JSON, ou ainda em PHP.

Outra API que também necessita de cadastro no site para obter a chave de pesquisa é a API da Technoratti (TECHNORATI, 2006). Nesta API não é informado

um limite de buscas por dia, somente é possível saber que cada pesquisa retorna até 100 resultados utilizando um formulário XML ou RSS.

A utilização de frameworks para o desenvolvimento de aplicações de PLN é uma proposta natural para integração de ferramentas para o processamento de texto como as anteriormente citadas. Os frameworks estão cada vez mais presentes na área de desenvolvimento de software, pois são a maneira pela qual os sistemas orientados a objetos conseguem a maior reutilização (GAMMA, 2000, p. 43).

Na literatura podem ser encontradas muitas definições para frameworks orientados a objetos. Segundo Silva (2000, p. 45) à abordagem de frameworks orientados a objetos utiliza o paradigma de orientação a objetos para produzir uma descrição de um domínio para ser reutilizada. Sendo assim um framework é uma estrutura de classes inter-relacionadas, que correspondem a uma implementação incompleta para um conjunto de aplicações de um domínio, sendo que esta estrutura de classes deve ser adaptada para a geração de aplicações especificas. Segundo Pree (1995, p. 54) um framework é uma coleção de classes abstratas e concretas que representam um subsistema, estas classes (abstratas e concretas) podem ser estendidas ou adaptadas para construir um novo subsistema. Para D'Souza e Wills (1998, p. 340) um framework pode ser interpretado como sendo um pacote de templates, um pacote desenvolvido para ser importado com substituições. Ele é entendido para prover uma nova versão, baseada nas substituições realizadas. Segundo Gamma (2000, p. 42) um framework predefine os parâmetros de projeto, de maneira que o projetista/implementador da aplicação possa se concentrar nos aspectos específicos da sua aplicação.

Este trabalho propõe o desenvolvimento de um framework para integração de ferramentas de PLN e APIs de mecanismos de busca. A próxima seção apresenta os objetivos do projeto.

Objetivos

Objetivo geral

Desenvolver um framework para o desenvolvimento de sistemas que integrem Sistemas de PLN e API's de mecanismos de busca.

Objetivos específicos

- Pesquisar na literatura definições e pesquisas relacionadas à Frameworks, pesquisas relacionadas à API's de mecanismos de busca na Web e sistemas de PLN consolidados;
- Desenvolver a proposta de Framework para integração de sistemas de PLN e API's de mecanismos de busca;
- Desenvolver um sistema baseado na proposta de Framework;
- Avaliar os resultados obtidos;

Metodologia

Para atingir os objetivos propostos este trabalho será dividido em etapas, descritas a seguir:

1. Será realizada uma revisão bibliográfica sobre a utilização de frameworks Orientados a Objetos, onde serão destacadas as principais vantagens de se utilizar este conceito;
2. Fazer um estudo sobre as principais API's de Mecanismos de Busca disponibilizados para os desenvolvedores terem acesso às bases de dados destes mecanismos, apontando quais recursos cada um disponibiliza e quais as suas limitações;
3. Realizar uma revisão bibliográfica sobre o Processamento da Linguagem Natural, mostrando suas origens e suas principais características;
4. Fazer um estudo sobre algumas ferramentas relacionadas ao PLN, comentando sobre suas principais características e limitações;
5. Redigir o texto do Trabalho de Conclusão I;
6. Fazer a modelagem dos recursos do framework utilizando alguns artefatos da UML;
7. Desenvolver o framework;
8. Implementar um sistema simplificado utilizando o framework desenvolvido;
9. Realizar testes para avaliar os resultados obtidos;
10. Revisão e correção do texto para entrega;
11. Apresentação para a banca avaliadora.

Cronograma

Abaixo consta a distribuição cronológica das etapas definidas na metodologia.

Trabalho de Conclusão I

Etapa	Agosto			Setembro			Outubro			Novembro			Dezembro		
1															
2															
3															
4															
5															

Trabalho de Conclusão II

Etapa	Janeiro			Fevereiro			Março			Abril			Maio			Junho		
6																		
7																		
8																		
9																		
10																		
11																		

Bibliografia

GASPERIN, Caroline; GOULART, Rodrigo; VIEIRA, Renata. **Uma ferramenta para resolução automática de correferência** In: *Anais do XXIII Congresso da Sociedade Brasileira de Computação, IV ENIA*. Campinas-SP: SBC, 2003, v.7. p.163 – 172.

PARDO, Thiago A.S; RINO, Lucia H.M.; NUNES, MariaG.V. (2003). **NeuralSumm: Uma Abordagem Conexionista para a Sumarização Automática de Textos**. In *Anais do IV Encontro Nacional de Inteligência Artificial* – ENIA, p. 1-10. Campinas-SP, Brasil. 2 a 8 de Agosto.

NUNES, Maria G. V. et al. **Introdução ao Processamento das Línguas Naturais**. In: Notas Didáticas do ICMC. São Carlos – SP. 1999, n. 38, 91p.

GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. **Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos**. Porto Alegre: Bookman, 2000. 364p.

JOHSON, Ralph. **Frameworks = (componnents + patterns)**. Communications of the ACM, New York: v. 40, n. 10, p. 39 – 42, Out. 1997. Disponível em: <www.idi.ntnu.no/grupper/su/courses/dif8901/papers2003/P-r22-johnson97.pdf>. Acesso em: 29 ago. 2006.

PREE, Wolfgang**. Design Patterns for Object-Oriented Software Development**. ACM Press Books e Addison-Wesley Publishing Company, 1995. 272p.

D'SOUZA, Desmond F.; WILLS, Alan C. **Objects, Components, and Frameworks with UML: The Catalysis Approach**. Massachusetts: Addison-Wesley Longman, Inc., 1998. 787p.

XEROX Research Centre Europe. **Research – Content Analisys: Language Tools**. Disponível em: http://www.xrce.xerox.com/competencies/content-analysis/demos/portuguese. Acesso em: 29 ago. 2006.

SNOWBALL. Disponível em: http://www.snowball.tartarus.org/. Acesso em: 29 ago. 2006.

VISUAL Interactive Syntax Learning. **Portuguese VISL**. Disponível em: http://visl.sdu.dk/visl/pt/parsing/automatic/trees.php. Acesso em: 29 ago. 2006.

ROBERTS, Andrew. **JBootCat**. Disponível em: http://www.andy-roberts.net/software/jbootcat/index.html. Acesso em: 29 ago. 2006.

GOOGLE. **Google Soap Search API (beta)**. Disponível em: http://www.google.com/apis/reference.html#2_5. Acesso em: 29 ago. 2006.

YAHOO. **Yahoo! Search Web Services**. Disponível em: http://developer.yahoo.com/search/. Acesso em: 29 ago. 2006.

TECHNORATI. **Technorati API Libraries**. Disponível em: http://www.technorati.com/developers/tools/libraries.html. Acesso em: 29 ago. 2006.