Universidade de São Paulo
Escola de Artes, Ciências e Humanidades

# Estudo de Técnicas de Extração de Informação a partir de textos da WEB

Victor Hugo Jabur Passavaz

Monografia apresentada à Escola de Artes, Ciências e Humanidades, da Universidade de São Paulo, como parte dos requisitos exigidos na disciplina ACH 2018 – Projeto Supervisionado ou de Graduação II, do curso de Bacharelado em Sistemas de Informação.

São Paulo, Janeiro de 2009

Universidade de São Paulo
Escola de Artes, Ciências e Humanidades

# Estudo de Técnicas de Extração de Informação a partir de textos da WEB

Victor Hugo Jabur Passavaz

**Orientação**

_____

Prof. Dr. Ivandré Paraboni

**Banca Examinadora**:

_____

Profª. Dr. Patrícia Rufino Oliveira

_____

Prof. Dr. Cândido Ferreira Xavier

São Paulo, Janeiro de 2009

# Agradecimentos

## A Deus

“Por me fazer mais forte nos momentos difíceis e por me ensinar que na vida, devemos ser verdadeiros e humildes para com todos os nossos irmãos. O mundo é a sua escola, para onde viemos compartilhar conhecimento e amor.”

## Aos Meus Pais

“Por todo o carinho e amor que tiveram comigo, fizeram-me um verdadeiro guerreiro para combater as injustiças da vida, sou eternamente grato por tudo o que me ensinaram, é maravilhosa a vida ao lado de vocês”

## Ao Meu Avô

“Meu eterno ídolo, por todos os momentos de alegria que me proporcionou ao longo desta linda estrada da vida, me ensinou que para amar ao próximo, é preciso abrir mão de nós mesmos, sem egoísmo, sem nenhuma pretensão. Sinto muitas saudades...”

## À Minha Noiva

“Muito obrigado pelo seu amor, sempre me amando mais do que a si mesma, tenho certeza de que Deus a colocou em minha vida, para me ensinar o seu verdadeiro significado. Juntos, viveremos os momentos mais lindos de amor...”

# Dedicatória

À comunidade EACH

É um orgulho para todos nós, alunos da EACH, participar de uma revolução que ocorreu na USP, com o surgimento de uma escola flexível e inovadora como a EACH. Temos ciência de que a aprovação e construção do projeto desta escola não foi nada fácil. Há vários anos a comunidade pobre da zona leste reivindicou por uma universidade renomada como a USP na zona leste.

Somos a primeira turma formada após todas estas reivindicações e lutas de nosso povo, devemos por tanto, agradecer a esta comunidade, pois sem ela, não estaria escrevendo hoje este documento. Muito Obrigado ! Cabe a nós alunos, valorizar o conhecimento que nos foi dado, gratuitamente, e devolver em forma de benefícios para a comunidade, disseminando conhecimento e fazendo o nome de nossa escola pelo mundo afora.

Agradeço a todos os professores, pela dedicação e carinho com que nos receberam em classe e compartilharam conosco a fundação desta maravilhosa escola. Uma nova semente foi plantada e já está dando ótimos frutos, tanto no mercado de trabalho, quanto na área de pesquisa científica.

Fica aqui os meus parabéns a toda a comunidade EACH, que a cada dia luta por fazer a escola crescer e que está proporcionando a formação superior de muitos jovens que antes não tinham condições de realizar tal sonho.

# Resumo

Este trabalho realiza um estudo de caso na área de Extração de Informação (EI) e implementação de um sistema EI subdividido em três módulos. Um módulo de obtenção de textos da internet (WEB), um módulo analisador morfossintático (Bick, 2000) responsável por preparar o texto e por fim o módulo responsável por extrair eventos do tipo "visita" de um conjunto textual, ie, extrair todos os trechos de textos do conjunto textual que contenham algo sobre "visita" e separá-los em tabelas contendo os campos (visitante, visitado e data da visita), estas "tabelas" serão chamadas de templates no decorrer deste trabalho.

A implementação foi feita utilizando-se a linguagem de programação Python, o kit de desenvolvimento para aplicações de Processamento de Linguagem Natural (PLN), o Natural Language Toolkit (NLTK), recursos estes que foram utilizados também no trabalho anterior (Passavaz, 2008).

## Palavras Chave

- Extração de Informação
- Linguagem Python
- Busca de Textos da WEB

# Índice

Agradecimentos ..... iii

Dedicatória ..... iv

Resumo ..... v

Lista de Figuras ..... vii

Lista de Tabelas ..... viii

Glossário ..... ix

**1. Introdução** ..... 1

**1.1 Objetivos** ..... 3

**1.2 Motivação** ..... 4

**1.3 Metodologia** ..... 4

**1.4 Estrutura do Trabalho** ..... 5

**2. Revisão Bibliográfica** ..... 5

**2.1 Introdução** ..... 5

**2.2 Definições em EI** ..... 7

**2.2.1 Tipos de Texto** ..... 7

**3. Trabalho Desenvolvido** ..... 8

**3.1 Módulo de Obtenção de Textos da Web** ..... 9

**3.2 Módulo de Extração de Eventos** ..... 11

**3.2.1 Introdução** ..... 11

**3.2.2 Objetivos do módulo de extração de eventos "visita"** ..... 12

**3.2.3 Diferenciação do Tipo de Extração** ..... 13

**3.2.4 Passo a Passo do algoritmo implementado (Módulo de Extração)** ..... 14

**4. Resultados** ..... 18

**5. Conclusões** ..... 24

**Apêndice** ..... 27

**Referências Bibliográficas** ..... 29

# Lista de Figuras

Figura 1 - Visão Esquemática de um processo de EI (FELDMAN) ..... 2

Figura 2 - Exemplo de Template que deve ser preenchido pelo Módulo 3 - EI ..... 3

Figura 3 - Exemplo de Texto Estruturado (Anotado) ..... 7

Figura 4 - Exemplo de Texto Semi Estruturado ..... 8

Figura 5 - Exemplo de Texto Não Estruturado (Livre) ..... 8

Figura 6 - Trecho contendo um evento "visita" a ser extraído ..... 12

Figura 7 - Template contendo o evento “Visita” ..... 12

# Lista de Tabelas

Tabela 1 - Índices de Confiança em EI pela conferência MUC (2005) ..... 6

Tabela 2 - Tabela de tipos de extratores criados ..... 14

Tabela 3 - Casos de Teste que compõem o “Conjunto de Testes” ..... 19

Tabela 4 - Total de atributos por categoria de frase ..... 19

Tabela 5 - Análise Sumarizada do Atributo "Visitante" ..... 20

Tabela 6 - Análise Sumarizada do Atributo "Visitado" ..... 20

Tabela 7 - Análise Sumarizada do Atributo "Data da Visita" ..... 21

Tabela 8 - Taxas de Acerto da Resolução de Pronomes ..... 21

Tabela 9 - Etiquetas utilizadas para a extração do evento visita ..... 28

# Glossário

| Número | Termo | Significado |
|---|---|---|
| 1 | Corpus / Corpora | Repositório de arquivos textos utilizados por aplicações PLN, geralmente acompanham kits de desenvolvimento ou são distribuídas em um pacote de arquivos, geralmente em vários idiomas e constituem arquivos de entrada para as aplicações PLN. |
| 2 | NLTK | Natural Language Toolkit – É um kit de desenvolvimento feito na linguagem Python e voltado especialmente para se trabalhar com Processamento de Linguagem Natural |
| 3 | Parser | Parser é um componente que serve para analisar a estrutura gramatical de uma entrada, que são segmentos de texto ou símbolos que podem ser manipulados. O parser pode ser um leitor que ajuda na conversão do arquivo para manipulação dos dados contidos no mesmo, é uma espécie de tradutor que converte uma dada entrada não estruturada em uma saída fácil de ser manipulada, como transformar um texto em XML por exemplo. |
| 4 | PLN | Processamento de Linguagem Natural |
| 5 | Python | É uma linguagem de programação. |
| 6 | Tagger | Um Tagger é um componente responsável por anotar ou fazer marcas em um determinado texto. Por exemplo, dado um texto "cru" de entrada, o tagger analisa gramaticalmente cada palavra do texto e faz uma marca. Frase de exemplo: O Armando caiu. O<artigoDefinido> Armando<nomeProprio> caiu <verbo>. |
| 7 | MUC | Message Understanding Conference – São conferências que formalizaram a tarefa de Extração de Informação e definiram métricas de avaliação dos sistemas de EI. |
| 8 | EI | Extração da Informação – Processo que visa extrair de um documento ou texto alguma informação relevante. |
| 9 | Texto Estruturado, Anotado ou Etiquetado | Texto que possui marcações bem definidas, neste trabalho serão assim chamados os textos que sofreram uma análise morfossintática, possuindo portanto marcações gramaticais. |

| | | |
|---|---|---|
| 10 | Template | É uma tabela que contém alguns atributos/colunas, neste trabalho os templates referem-se ao evento "visita" e possuem os atributos (visitante, visitado e data da visita). |
| 11 | Evento do tipo "visita" | Estes eventos aparecem ao longo de todo o trabalho e significam um template (tabela) que contém algum tipo de informação que envolva o termo "visitar". Por exemplo, Lula visitou o senado, ... A China foi visitada durante as eleições, a etc ... Estas tabelas, neste trabalho, contém os campos (visitante, visitado e data da visita) |
| 12 | HTML | Hyper Text Markup Language – É uma popular linguagem através da qual são desenvolvidas as páginas / websites da internet. |
| 13 | WEB | Abreviação de World Wide Web (WWW), muitas vezes a expressão WEB é utilizada como sinônimo de Internet. |
| 14 | PALAVRAS | É um trabalho de doutorado (Bick, 2000) que implementa um analisador morfossintático e neste trabalho foi aplicado de forma a etiquetar (preparar) os textos que serão utilizados para o processamento, será chamado de módulo 2 neste trabalho. |
| 15 | URL | Uniform Resource Locator – É um sinônimo para o endereço de uma página de internet, podemos dizer que http://www.usp.br é um endereço de internet ou então chamá-lo simplesmente de URL. |
| 16 | HTTP | Hypertext Transfer Protocol – É o protocolo de comunicação utilizado para trafegar os dados por intranets e pela internet. |

# 1. Introdução

Com o crescimento da WEB, cresce também o número de documentos eletrônicos à disposição das pessoas, sendo uma tarefa muito árdua a localização de um conteúdo desejado nestes documentos. Atualmente são utilizados buscadores na WEB tais como o Google e o Yahoo, que retornam um número grande de documentos contendo informações relevantes, mas o usuário ainda deve analisá-los manualmente para localizar a informação desejada. Este problema é conhecido como Sobrecarga de Informação. (Loh, Wives, Oliveira, 2000).

Uma possível melhoria nesta situação seria o uso de um sistema de Extração de Informação (EI) que converte texto não estruturado, ie, textos sem qualquer tipo de formatação, marcação ou análise de qualquer espécie, para uma forma que pode ser carregada em uma tabela de banco de dados. (Konchady, 2006). Desta forma, torna-se possível (ou mais fácil) tratar o conteúdo destes textos de forma automática.

Este trabalho realiza a implementação prática de um sistema de Extração de Informação (EI) subdividido em três módulos principais. O primeiro módulo é responsável por percorrer páginas de internet (WEB), obtendo delas textos não estruturados, i.e., ocorre um processo de remoção de tags HTML de modo que o texto fique isento de qualquer tipo de marcação ou organização, estes textos são armazenados em arquivos no disco rígido para futuro processamento. O segundo módulo é um analisador morfossintático, conhecido também como POS-TAGGER, este módulo não foi desenvolvido por este trabalho, foi utilizado o PALAVRAS (Bick, 2000) responsável por transformar os textos obtidos no primeiro módulo (textos não estruturados, Figura 5) em textos estruturados, ie, textos que possuem marcações bem definidas contendo análise morfológica e sintática, também são conhecidos como textos anotados ou etiquetados (Figura 3). O terceiro módulo realiza a extração de informação, recebendo para isso o texto estruturado (saída do módulo 2) e fornece como resultado um conjunto de

tabelas (templates) preenchidas com informações acerca do evento "visita" (nome do visitado, nome do visitante e data da visita).

Este trabalho mantém uma integração com o trabalho de conclusão desenvolvido em Junho (Passavaz, 2008) referente ao mecanismo de resolução de pronomes em terceira pessoa desenvolvido no trabalho anterior e utilizado neste trabalho para auxílio no processo de extração de informação (módulo 3), nos casos em que o nome do visitante / visitado é um pronome em terceira pessoa (Eles, Elas, Ele, Ela). Foram também utilizados do trabalho anterior os mesmos arquivos textos (Corpus Folha de São Paulo, caderno Brasil 1994), a linguagem de programação Python, incluindo também o kit NLTK.

O módulo responsável pela extração dos eventos visita (módulo 3) é o mais significativo neste trabalho, segue abaixo a Figura 1 que tenta explicar melhor o seu funcionamento. O módulo recebe como entrada um grande volume de textos estruturados (preparados pelo módulo 2) para serem processados, sua função é utilizar um conjunto de regras pré-determinadas (Rule Modules), como exemplo (Sujeito + Verbo visitar + Objeto Direto) para extrair dos textos uma grande quantidade de tabelas (templates) que contenham os campos (atributos) desejados, no caso deste trabalho é (nomes do visitante, nome do visitado e data da visita).

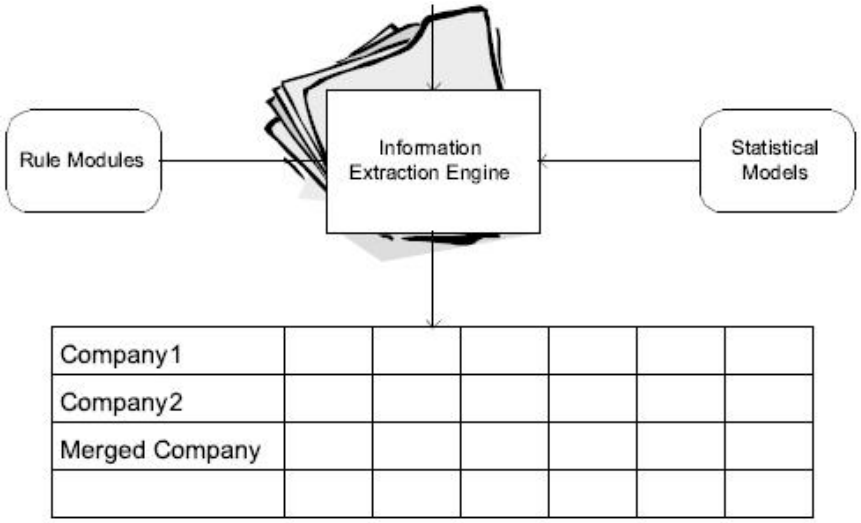


**Figura 1 - Visão Esquemática de um processo de EI (FELDMAN)**

## 1.1 Objetivos

O objetivo deste trabalho é o estudo de técnicas básicas de EI através da implementação de um sistema deste tipo, englobando um módulo para capturar textos da WEB, um segundo módulo para fazer o tratamento destes textos e um terceiro módulo responsável por extrair eventos do tipo visita.

Para que o trabalho fosse adequado ao cronograma (tempo) foram tomadas algumas medidas. Uma delas foi utilizar como módulo 2 o analisador PALAVRAS (Bick, 2000), que já fornece um analisador morfossintático com muitos recursos prontos. Outra medida tomada foi o desacoplamento dos módulos, ie, os textos obtidos pelo módulo de captura de textos na WEB não foram utilizados neste trabalho, foi utilizado um Corpus textual (caderno Brasil do jornal Folha de São Paulo do ano de 1994) previamente etiquetado pelo analisador PALAVRAS (Bick, 2000).

O módulo que recebeu maior tempo para desenvolvimento foi o módulo 3, que realiza extração de eventos do tipo "visita" e é o foco principal deste trabalho. O evento "visita" foi escolhido por possuir um grande número de ocorrências no Corpus e também por conter diversos atributos envolvendo este evento, dos quais foram escolhidos apenas os três de acordo com o exemplo abaixo:

*"...Henrique tem visitado outros presidenciáveis peemedebistas – esteve ontem com o governador Iris=Rezende , de Goiás , e vai hoje=à=noite a Curitiba ( PR ) , para se encontrar com o governador Roberto=Requião ...."*

| | |
|---|---|
| Visitante: | Henrique |
| Visitado: | presidenciáveis peemedebistas |
| Data da Visita: | - |

**Figura 2 - Exemplo de Template que deve ser preenchido pelo Módulo 3 - EI**

## 1.2 Motivação

A principal motivação de um sistema de EI é a necessidade de processamento automático de informações contidas em texto não estruturado em grandes volumes.

Sistemas de EI são usados, por exemplo, para examinar grandes quantidades de documentos na web e construir bancos de dados a partir deles. Por exemplo, anúncios classificados de imóveis, empregos, preços de produtos comercializados ( para comparação entre diferentes fornecedores etc ... ) podem ser examinados em sua forma textual e convertidos automaticamente em registros de uma base de dados estruturada.

## 1.3 Metodologia

Para o desenvolvimento deste trabalho, foi seguida a seguinte metodologia, numerada em ordem de execução:

1. Seleção da base de textos que será utilizada no trabalho.
2. Estudo de técnicas básicas de EI.
3. Estudo e implementação do módulo que realiza o download de textos da WEB.
4. Estudo sobre a extração de relações através do preenchimento de templates (pré-definidos).
5. Implementação do Sistema.
6. Testes e verificação dos resultados obtidos pelo algoritmo de extração.
7. Escrita da Monografia

### 1.4 Estrutura do Trabalho

O restante do trabalho está organizado conforme descrito abaixo:

Seção 2. Revisão Bibliográfica – Dividido em três partes:

2.1. Introdução – Trata de uma descrição geral sobre o que é Extração de Informação, informando alguns conceitos chave e alguns exemplos sobre como funciona uma aplicação deste tipo.

2.2. Definições em EI – Trata termos técnicos da área de EI para familiarização.

Seção 3. Trabalho Desenvolvido – Detalhamento de todo o trabalho desenvolvido. São dois módulos desenvolvidos e explicados nesta seção (módulo de extração de textos Web e módulo de Extração de Eventos do tipo "visita")

Seção 4. Resultados – Contém três seções: avaliação dos testes, exibição dos resultados e uma discussão sobre estes.

Seção 5. Conclusões – Contém uma avaliação geral deste trabalho, relatando os conhecimentos aprendidos durante sua execução, uma síntese das contribuições do mesmo e sugestões para trabalhos futuros.

## 2. Revisão Bibliográfica

### 2.1 Introdução

A área de Extração de Informação (EI) é uma sub-área de Processamento de Linguagem Natural (PLN) e procura resolver um dos problemas atuais que é a localização de uma informação específica em meio a crescente quantidade textual, como a WEB por exemplo. O objetivo da área é transformar textos semi-estruturados ou não-estruturados em dados estruturados que serão armazenados em um banco de dados. (Corrêa, 2003).

Deste modo, sistemas extraem automaticamente valores de atributos (como campos de um banco de dados), sem, no entanto realizar uma análise completa do conteúdo do texto. (Loh, Wives e Oliveira). A idéia principal é proporcionar à sociedade uma forma eficiente de se realizar buscas sem a necessidade de leitura de grandes quantidades textuais para que se encontre a informação desejada. Por exemplo, quando realizamos uma busca hoje em qualquer site de busca como Google, Yahoo, dentre outros, o sistema retorna páginas que contenham em algum ponto a expressão desejada, é necessário a leitura de grande parte do documento para se chegar à informação desejada.

Com textos previamente tratados, é possível a utilização da informação de forma mais eficiente. Isso permite, por exemplo, a comparação de preços de produtos de duas lojas virtuais, ou a descoberta de conhecimento usando Mineração de Dados a partir das tabelas de dados extraídos dos documentos. (Leitão; Torres; 2005).

A conferência MUC (Message Understanding Conference) realiza uma competição na área de EI que incentiva o desenvolvimento de técnicas através de conferências periódicas, sendo a mais recente a MUC 7 realizada em 1997. Segue abaixo uma tabela que a conferência publicou em seu site, contendo uma análise do estado atual da arte da área Extração de Informação (análise de 2005).

| Itens de Informação | Porcentagem de Confiança |
|---|---|
| Entidades | 90 % |
| Atributos | 80 % |
| Fatos | 70 % |
| Eventos | 60 % |

**Tabela 1 - Índices de Confiança em EI pela conferência MUC (2005)**

Segue a descrição de cada um dos itens de informação considerada categorias de competição na série de conferências MUC.

- Entidades: Define-se como entidades, segundo o MUC, nomes de pessoas, organizações, locais, datas, horas, porcentagens e valores monetários.

- Atributos: São características relacionadas a entidades. Cada tipo de entidade terá atributos específicos, por exemplo, uma entidade Pessoa terá atributos como, nome, idade, RG, CPF, dentre outros.

- Fatos: São relações entre as entidades, tais como, empregado-de, local_de, filho_de.

- Eventos: Define-se evento como uma atividade ou ocorrência de interesse, tal como o evento "visita" deste trabalho.

## 2.2 Definições em EI

### 2.2.1 Tipos de Texto

Técnicas apropriadas devem ser aplicadas dependendo do tipo de texto que um algoritmo EI estiver utilizando. Basicamente são 3 os tipos de textos considerados:

- **Texto Estruturado:** Também conhecido como marcado ou anotado, possui marcadores bem definidos que facilitam a extração de informação. Segue um trecho de texto estruturado utilizado por este trabalho, produzido com uso do etiquetador PALAVRAS (Bick, 2000).

```
O   [o] <artd> DET M S @>N
aumento    [aumento] <act> N M S @SUBJ>
de  [de] <np-close> PRP @N<
impostos    [imposto] <mon> N M P @P<
em [em] <sam-> <np-close> PRP @N<
```

**Figura 3 - Exemplo de Texto Estruturado (Anotado)**

- **Texto Semi Estruturado:** Este texto não chega a ser tão bem definido quanto o texto estruturado. Possui alguma regularidade, mas não respeita obrigatoriamente uma determinada organização. É o caso, por exemplo, do texto presente em páginas web geradas dinamicamente (eg, as informações sobre um produto em um site como Amazon). Nestes casos, o texto, embora de teor

variável, apresenta certas regularidades estruturais que podem facilitar o seu tratamento.

- 09/09/2008 - (Diretoria) Disponível material com o logotipo novo da EACH em comemoração aos 75 anos da USP, para inclusão nos documentos oficiais da USP
- 08/09/2008 - (Extensão) Lista de Selecionados para Oficina de Teatro na EACH
- 18/07/2008 - (Graduação) Divulgação da Atribuição Didática dos Docentes da EACH
- 01/07/2008 - (Extensão) Universidade Aberta à Terceira Idade - veja a programação para o 2o. semestre

**Figura 4 - Exemplo de Texto Semi Estruturado**

- **Texto Não Estruturado:** É o texto livre, escrito em linguagem natural, sem nenhuma organização. É este tipo de texto que é extraído da web pelo módulo de download de textos deste trabalho, por exemplo:

```
O analista de sistemas usa criativamente a tecnologia da informação para aquisição
de dados, comunicação, coordenação, análise e apoio às decisões gerenciais.
Seus conhecimentos são usados em funções organizacionais, como: contabilidade,
finanças, vendas e produção. Seu papel pode ser definido na articulação entre
recursos humanos e computacionais que interrelacionam a coleta, o armazenamento,
a recuperação, a distribuição e o uso de dados com o objetivo de eficiência
gerencial(planejamento, controle, comunicação e tomada de decisão) nas organizações.
```

**Figura 5 - Exemplo de Texto Não Estruturado (Livre)**

## 3. Trabalho Desenvolvido

Este capítulo descreve a implementação de um estudo de caso na área Extração de Informação (EI). A implementação foi desenvolvida na linguagem Python e está dividido em dois módulos:

1. Desenvolvimento de um Módulo de Extração de Textos da WEB a partir de um endereço raiz. Neste trabalho foi utilizado como teste o endereço do Jornal Folha de São Paulo (http://www.folha.uol.com.br).

2. Implementação de regras básicas de extração de eventos do tipo "visita", ie, extração de templates (tabelas) com os campos (nome do visitante, nome do visitado e data da visita) utilizando para isso um Corpus (Caderno Brasil, 1994 –

Folha de São Paulo) que já foi devidamente preparado pelo analisador morfossintático PALAVRAS (Bick, 2000).

## 3.1 Módulo de Obtenção de Textos da Web

O algoritmo desenvolvido aplica os seguintes passos para a extração dos textos da WEB:

**Passo 1 – Início da Busca de Textos a partir de uma URL Origem**

O algoritmo recebe como entrada dois dados, uma URL origem, ie, uma página de internet que será o ponto de início das buscas e recebe também um número inteiro que é o número de páginas WEB que devem ser retornadas. Os testes foram feitos para a URL (http://www.folha.uol.com.br).

```
listaPaginas = extrator.carregarConjuntoPaginas(http://www.folha.uol.com.br/, 10)
```

**Passo 2 – Captura de Dados da Página Inicial**

Neste ponto, ocorre uma captura de todos os links HTTP contidos na página inicial, juntamente com o texto e o título da página. Os dados são armazenados em um objeto da classe de nome PaginaHTML. Abaixo segue a chamada do método:

```
pagina = self.carregarPaginaWeb(url)
```

**Passo 3 – Iteração sobre os Links HTTP**

O programa itera sobre cada link da página inicial armazenando-o em uma pilha que não aceita repetição de links. O objetivo é iterar até que se consiga atingir o número de páginas solicitadas pelo usuário ou então até que todos os links da página inicial se esgotem.Ocorre a execução de um laço infinito que terminará apenas quando a pilha de links HTTP descobertos estiver vazia ou quando o algoritmo encontrar a quantidade de links desejados, no caso acima (3.1.1) é 10 páginas.

Esta pilha de links, armazena todos os links HTTP encontrados na página inicial e a cada nova página descobre mais links que são adicionados à pilha. Isto faz com que ela aumente o seu tamanho rapidamente.

**Passo 4 – Filtragem de Links**

Foi implementado um filtro que executa a remoção de links "lixo", melhorando bastante a qualidade dos textos extraídos da WEB.

Segue abaixo como funcionam as restrições deste filtro:

```
def setFiltroLinks(self):
     self.filtroLinksAceitos.append(folha.uol)
     self.filtroLinksAceitos.append(.shtml)
     self.filtroLinksRejeitados.append(?)
```

Como pode ser visto acima, foram criadas três regras de filtro: duas regras de aceite e uma regra de negação. O algoritmo irá devolver uma lista de objetos PaginaHTML que contenham em seu endereço as palavras "folha.uol" E ".shtml", negando os endereços que contenham o interrogação "?".

Com estas regras obteve-se páginas com boa qualidade de texto. No site Folha de São Paulo, as notícias terminam com a extensão ".shtml" e contém em seu endereço a expressão "folha.uol".

**Passo 5 – Armazenamento dos textos em Disco Rígido**

Ocorre neste ponto uma iteração da lista de objetos PaginaHTML obtidos e a escrita de cada objeto em um arquivo texto no disco rígido.

**Considerações Gerais**

Existem duas funcionalidades prontas que foram utilizadas e que são bastante relevantes para este módulo. Uma delas é um recurso existente no pacote NLTK e que desempenha o papel de remover as marcações HTML e extrair o texto puro, que é o que realmente interessa para a Extração de Informação. Segue abaixo a utilização deste recurso:

```
nltk.clean_html(textoHtml)
```

O outro recurso utilizado é um Parser de páginas HTML disponibilizado pela linguagem Python. Dado um endereço qualquer, o recurso devolve um objeto Python que já contém as informações de título, uma lista de links existentes e o texto (não limpo, isto é, ainda com marcações HTML). Segue abaixo um exemplo de sua utilização

```
def setLinksTituloPaginaHtml(self, paginaHtml, textoHtml):
        parser = htmllib.HTMLParser(formatter.NullFormatter())
        parser.feed(textoHtml)
        parser.close()
        if parser.title != None:
            paginaHtml.setTitulo(parser.title)
        else:
            paginaHtml.setTitulo(Sem Titulo)
        paginaHtml.setListaLinks(parser.anchorlist)
```

Como pode ser observado acima, o método setLinksTituloPaginaHtml recebe como parâmetros um objeto Python (paginaHtml) para que sejam preenchidas as informações de Título e lista de links e uma string que contém a página (com marcações HTML) da qual se deseja obter as informações.

Nas três primeiras linhas do método é onde ocorre todo o processamento importante. Para se obter o título se utiliza parser.title e para se obter a lista de links devolvida pelo Parser utiliza-se parser.anchorList.

## 3.2 Módulo de Extração de Eventos

### 3.2.1 Introdução

Uma fase inicial deste trabalho foi a escolha do tipo de evento que seria extraído para uso como caso de estudo. Procurava-se por um evento que tivesse um bom número de ocorrências no conjunto de textos utilizado e que ainda tivesse informações relevantes para serem extraídas, foi escolhido o evento "visita", motivado pelo elevado número de ocorrências (659) do verbo "visitar" e (692) do substantivo "visita" e pela relevância no domínio (temas políticos, em período de eleição).

### 3.2.2 Objetivos do módulo de extração de eventos “visita”

Para ilustrar o processo de extração, segue abaixo um trecho contido nos textos utilizados e que contém o evento “visita” a ser extraído. As informações morfossintáticas (representadas por etiquetas) foram obtidas com uso do parser PALAVRAS. (Bick, 2000).

```
</s>
FHC  [FHC] <hum> PROP M/F S @SUBJ>
visitou  [visitar] <fmc> <mv> V PS 3S IND VFIN @FS-STA
ontem    [ontem] ADV @<ADVL
as   [o] <artd> DET F P @>N
cidades  [cidade] <Lciv> N F P @<ACC
de   [de] <NER2> <np-close> PRP @N<
Resende  [Resende] <civ> PROP M/F S @P<
e    [e] <co-prparg> <co-prparg> KC @CO
Barra=Mansa  [Barra=Mansa] <civ> PROP F S @P<
$,
região   [região] <Ltop> <L> <np-close> N F S @N<PRED
sul  [sul] <np-close> ADJ M S @N<PRED
de   [de] <sam-> <np-close> PRP @N<PRED
o    [o] <artd> <-sam> DET M S @>N
Rio=de=Janeiro   [Rio=de=Janeiro] <civ> PROP M S @P<
$.
</s>
```

**Figura 6 - Trecho contendo um evento "visita" a ser extraído**

Este trecho de texto etiquetado representa uma sentença delimitada por </s>. O primeiro termo de cada linha é a palavra original do texto, seguida de sua forma canônica entre colchetes e várias etiquetas de informação morfossintática e semântica. Por exemplo, a palavra “visitou” é derivada de [visitar], um verbo (V) indicativo (IND) de 3ª pessoa singular (3S).

| | |
|---|---|
| Visitante: | FHC |
| Visitado: | cidades de Resende e Barramansa |
| Data da Visita: | ontem |

**Figura 7 - Template contendo o evento “Visita”**

A função do módulo de extração é identificar todas as frases que contenham o verbo "visitar" ou o substantivo "visita" e extrair templates (figura 8) que contenham a informação desejada. A coleção textual utilizada possui um total de 363 arquivos texto do caderno Brasil da Folha de São Paulo do ano de 1994, período em que ocorreram eleições presidenciais. Assim, o problema resolvido por este trabalho é a elaboração de regras de extração para eventos do tipo "visita" com base nas etiquetas existentes na coleção textual utilizada.

#### Etiquetas Utilizadas

Para conseguir extrair as três informações pretendidas (nome do visitante, nome do visitado e a data da visita) foram utilizadas algumas das etiquetas contidas na coleção de textos analisada e preparada pelo módulo PALAVRAS (Bick, 2000). No apêndice, (tabela 8) seguem as etiquetas utilizadas e uma breve descrição sobre sua utilização.

Em uma primeira análise da resolução do problema "extração de eventos do tipo visita", pode parecer que a utilização das etiquetas fornecidas pelo módulo PALAVRAS reduza a complexidade deste trabalho a uma simples busca de strings, ie, para encontrar um evento visita basta buscar pela etiqueta "[VISITAR]", mas, em uma análise mais acurada, percebe-se que as três informações desejadas (nome do visitante, nome do visitado e a data da visita) aparecem nas mais variadas formas possíveis e a dificuldade deste trabalho está na criação de regras que consigam perceber estas "diferentes combinações" e consigam extrair corretamente as três informações desejadas.

### 3.2.3 Diferenciação do Tipo de Extração

Para aumentar o desempenho do algoritmo foi criado um extrator específico para cada tipo de tempo verbal. Segue abaixo uma tabela exemplificando cada tipo de frase e o extrator que foi utilizado:

| Tipo de Frase | Descrição |
|---|---|
| Verbo no Particípio | Frases que tenham o verbo no Particípio seguidos pela palavra por (eg, visitado por) possuem os atributos visitantes e visitados em ordem inversa (frases na voz passiva) |
| Verbo com Ênclise | Frases que tenham o verbo com Ênclise (eg, visitá-lo) devem ser extraídas de um modo diferenciado, isto é, ao pronome "-lo" será aplicado o processo de co-referência de pronomes para identificar o termo antecedente de "-lo". |
| Substantivo "Visita" | Frases que contenham o substantivo "visita" ao invés do verbo apresentam outras características, usam um extrator específico para este tipo de frase |
| Negação | Frases que contenham a negação de visita (eg, Lula não visitou ...), devem ser desconsideradas para a extração. |
| Casos Normais | Frases que não sejam nenhum dos casos anteriores são classificadas nesta categoria. |

**Tabela 2 - Tabela de tipos de extratores criados**

### 3.2.4 Passo a Passo do algoritmo implementado (Módulo de Extração)

Abaixo segue um resumo de quais passos o algoritmo segue para resolver o trabalho de extração de eventos do tipo “visita” :

**Passo 1 – Busca por frases que contenham o evento “visita”**

Procura frases que contenham o verbo “visitar” ou o substantivo “visita” em todos os arquivos textos (ao todo são 363 arquivos). Extrai todas as frases de modo que não precise mais realizar buscas sobre o conjunto total de arquivos. Para os arquivos texto utilizados, o algoritmo extraiu 659 frases contendo o verbo “visitar” e 692 frases contendo o substantivo “visita”.

**Passo 2 – Classificação das frases encontradas**

Cada frase encontrada é submetida ao identificador de tipos de extração (tabela 2) que irá identificar e classificar cada frase em uma categoria para que sejam aplicadas regras específicas de extração para a frase em questão.

**Passo 3 – Aplicação de Regras Específicas**

Neste ponto temos uma frase que contém o evento visita e que já está classificada em alguma das cinco categorias (tabela 2). Abaixo serão explicadas detalhadamente quais as regras que o algoritmo aplica a cada uma

das categorias, utilizando exemplos com frases reais da coleção de textos utilizada.

a. **Ênclise** – As frases com ênclise são aquelas nas quais aparece um hífen (eg, visitá-lo). Segue um exemplo de frase:

*"Jânio=Quadros estava terminando seu governo em São=Paulo e, antes que deixasse o posto, o embaixador britânico resolveu visitá-lo (ref. Jânio=Quadros). "*

Para a frase acima o algoritmo extraiu o seguinte template:

**Visitante:** embaixador

**Visitado:** Jânio=Quadros

**Data da Visita:** vazio

Para esta classificação de frase, o algoritmo com base no pronome "-lo" realizou uma resolução de pronomes (ie, identificou o termo antecedente do pronome no texto). Caso o termo antecedente não aparecesse na mesma frase, o algoritmo recuaria para frases anteriores até identificá-lo. O algoritmo utilizado é o apresentado em Cuevas & Paraboni (2008).

Cada frase extraída possui ponteiros que indicam a frase anterior e isso é utilizado para o recuo no texto à procura do termo antecedente. Nesta categoria, o termo antecedente que foi encontrado (Jânio=Quadros) é classificado como "visitado". Para encontrar o "visitante", o algoritmo recua a partir do verbo buscando pelo sujeito da frase (que contém a etiqueta @SUBJ) e que seja uma entidade (que contém um dos atributos da lista de etiquetas que representam Entidades, vide Tabela 2).

A resolução de pronomes foi a integração realizada com o trabalho anterior (Passavaz, 2008). Ela não é obrigatoriedade neste trabalho uma vez que o algoritmo poderia simplesmente retornar os pronomes "-lo", "-la", "-los" ou "-las" como sendo o "visitado", mas adicionalmente decidiu-se acoplar o trabalho anterior para a resolução destes pronomes, melhorando

desta forma a resposta do algoritmo deste trabalho, retornando como "visitado", por exemplo, a expressão "Lula" ao invés da expressão "-lo". Demonstrando também que os dois trabalhos podem ser integrados facilmente.

b. **Substantivo "visita"** – As frases desta categoria contêm o substantivo "visita" ao invés do verbo. Esta é a categoria onde ocorre a maior dificuldade de extração, pois o termo "visita" aparece nas mais variadas formas possíveis.

*"Em ritmo de campanha, ele (ref. = Britto) tem recebido visitas de prefeitos e vereadores de o interior."*

A frase acima apresenta "visitante" e "visitado" em ordem inversa, ou seja, "visitado" aparece antes do termo "visita" e "visitante" aparece depois do termo "visita". Neste tipo de frase deveria ser utilizado regras parecidas com as do particípio.

Para esta categoria foram criadas duas regras, sendo a primeira regra: [fazer] + "visita" e a segunda regra: [receber] + "visita". Outras regras poderiam ser adicionadas, mas estas duas servem de exemplo como base de extração. Com estas duas regras obtém-se 5% de frases extraídas do total de frases que contém o substantivo visita.

À medida que novas regras vão sendo adicionadas ocorre uma abrangência maior de frases, no exemplo acima com 2 regras foi possível cobrir 5% dos casos e são as frases que apresentam os verbos fazer ou receber antes do substantivo "visita". Esta categoria necessita de muitas regras para que possa ser atingido uma abrangência relevante de extração, as 2 regras servem apenas como exemplo, não é o objetivo deste trabalho ficar criando muitas regras para obter uma alta porcentagem de abrangência, mas sim exemplificar o processo de criação de regras.

c. **Negação –** Esta categoria é constituída por frases que negam a "visita", tais como: "Não vai visitar...", " ...está evitando visitar ..." etc. As frases pertencentes à categoria são eliminadas dos passos seguintes e o processamento com uma nova frase é iniciado.

d. **Particípio –** Pertencem a esta categoria as frases que possuem o verbo visitar no tempo verbal Particípio seguidas pela palavra "por", significando uma frase na voz passiva. Segue um exemplo de frase:

*"Eles estão sendo visitados de casa em=casa por técnicos de a FAC ."*

Para esta frase, o algoritmo extrai como "visitantes" a palavra "técnicos" e como "visitados" o pronome "Eles". É importante ressaltar que a extração se dá no sentido inverso para frases desta categoria, ou seja, o sujeito agente aparece depois do verbo "visitar", complementando, é uma frase na voz passiva.

O algoritmo localiza o "visitante" após o verbo "visitar" com a ajuda das etiquetas "Entidade" e procurando principalmente por palavras que contenham a etiqueta de "Objetos" (Direto e Indireto), pois os objetos complementam o verbo "visitar" e em grande parte das vezes representam os "visitantes".

Para frases deste tipo o sujeito é passivo, ou seja, o "visitado" é o sujeito deste tipo de frase e o algoritmo classifica como "visitado" o sujeito desta frase com a ajuda das etiquetas de sujeito (@SUBJ).

e. **Casos Comuns –** São pertencentes a esta categoria todas as frases que não se enquadram nas outras categorias. São a maioria das frases da coleção textual utilizada. Segue um exemplo de frase:

*"Malu contou que foi visitar a mãe em o sábado passado e ela estava bastante grogue . "*

Para a frase acima, o algoritmo extraiu corretamente "Malu" como "visitante", "mãe" como "visitado" e "sábado" como sendo a "data da visita".

O algoritmo funciona de maneira idêntica ao caso "Particípio", explicado detalhadamente no item d, só que classifica o "visitante" como sujeito e o "visitado" como objeto (ordem inversa do Particípio).

# 4. Resultados

Nesta seção é apresentado o procedimento realizado para o teste da implementação prática deste trabalho bem como os resultados obtidos juntamente com uma análise explicativa ao final da seção.

Para a execução dos testes foram escolhidas algumas das frases que contêm o evento "visita" no Corpus Textual (tabela 3) para compor o que chamaremos aqui de "Conjunto de Testes" que por sua vez não abrange todas as frases da coleção textual por ser necessária uma análise manual dos resultados apresentados pelo algoritmo. Cada frase do conjunto de testes sofreu uma análise manual (acerto ou erro) que foram contabilizados através das tabelas 5 a 8.

Esta seção está dividida em três partes, onde a primeira delas explica quais foram os critérios escolhidos para a montagem do conjunto de testes seguida pela segunda parte que apresenta a contabilização percentual (contagem manual) de acertos e erros do algoritmo para cada uma das frases existentes no conjunto de testes, seguida pela parte final que realiza uma análise das porcentagens obtidas, explicando as mesmas através dos pontos fortes e fracos do algoritmo desenvolvido.

## 1 - Preparação do Conjunto de Testes

O conjunto de testes é composto por frases que contém o evento "visita" e todos os resultados apresentados neste trabalho são baseados nele, pois não haveria tempo suficiente para analisar manualmente todas as frases existentes no Corpus Textual e também não é o propósito deste trabalho analisar tantas frases assim, com o número de frases existentes no conjunto de testes é possível realizar uma avaliação satisfatória do algoritmo desenvolvido.

A tabela 3 abaixo apresenta uma contagem geral de todas as frases que contêm o evento "visita", existentes no Corpus Textual, e mostra quantas destas frases (separadas por categoria) foram utilizadas no Conjunto de Testes. Para a

categoria “Casos Normais”, por exemplo, foi pré-definido testar um total de 100 das 692 frases existentes, mas foi o algoritmo que sorteou aleatoriamente quais seriam estas 100 frases, o mesmo foi feito para a composição do conjunto de testes das outras categorias.

| Tipo de Frase | Número Total de Frases do Conjunto de Teste | Número Total de Frases da Coleção Textual | Percentual de Abrangência dos Casos de Teste |
|---|---|---|---|
| Verbo no Particípio | 32 | 32 | 100.00% |
| Verbo com Ênclise | 3 | 3 | 100.00% |
| Substantivo "Visita" | 32 | 659 | 4.86% |
| Casos Normais | 100 | 692 | 14.45% |
| **Total** | **167** | **1386** | 12.05% |

**Tabela 3 - Casos de Teste que compõem o “Conjunto de Testes”**

Na tabela acima é possível notar que existem apenas 3 casos com ênclise, este foi o número de ocorrências total em todo o conjunto textual, este número não é significativo, mas é possível que em outros Corpus exista um maior número de ocorrências de ênclise, mas apesar de tudo, a extração desta categoria já foi implementada e pode ser utilizada em outras ocasiões.

A tabela 4 abaixo sumariza uma contagem manual do número de vezes que cada atributo (visitante, visitado e data da visita) aparece nas frases do Conjunto de Testes.

| Tipo de Frase | Número Total de "Visitantes" | Número Total de "Visitados" | Número Total de "Data da Visita" |
|---|---|---|---|
| Verbo no Particípio | 31 | 32 | 9 |
| Verbo com Ênclise | 3 | 3 | 2 |
| Substantivo "Visita" | 29 | 30 | 18 |
| Casos Normais | 94 | 99 | 56 |
| **Total** | **157** | **164** | **85** |

**Tabela 4 - Total de atributos por categoria de frase**

## 2 - Avaliação Manual

Esta seção refere-se à análise manual realizada sobre os resultados apresentados pelo algoritmo. A Tabela 4 contém o número de vezes que os atributos “nome do visitante”, “nome do visitado” e “data da visita” aparecem no conjunto de testes e a contagem de erros e acertos para cada um dos atributos estão representados respectivamente nas Tabelas 5, 6 e 7. A Tabela 8 refere-se à integração realizada com o trabalho anterior (Passavaz, 2008) e apresenta quantas vezes os pronomes Ele, Ela, Eles, Elas, aparecem no conjunto de testes juntamente com a contagem de acertos e erros do algoritmo, lembrando que, para estes pronomes o algoritmo realizou uma co referência de pronomes, i.e., recuando no texto e procurando o termo antecedente utilizando para isto o algoritmo Cuevas & Paraboni (2008), abordado detalhadamente no trabalho anterior (Passavaz, 2008).

As Tabela 5, 6 e 7 abaixo, mostram respectivamente o número total de vezes que os atributos “nome do visitante”, “nome do visitado” e “data da visita” aparecem no conjunto de testes e quantas vezes o algoritmo obteve sucesso. A coluna Tipo de Frase refere-se à categoria da frase (Tabela 2).

| Tipo de Frase | Número Total de "Visitante" | Número de Acertos | % de Acertos |
|---|---|---|---|
| Verbo no Particípio | 31 | 24 | 77.42% |
| Verbo com Ênclise | 3 | 3 | 100.00% |
| Substantivo "Visita" | 29 | 27 | 93.10% |
| Casos Normais | 94 | 89 | 94.68% |

**Tabela 5 - Análise Sumarizada do Atributo "Visitante"**

| Tipo de Frase | Número Total de "Visitado" | Número de Acertos | % de Acertos |
|---|---|---|---|
| Verbo no Particípio | 32 | 30 | 93.75% |
| Verbo com Ênclise | 3 | 1 | 33.33% |
| Substantivo "Visita" | 30 | 17 | 56.67% |
| Casos Normais | 99 | 79 | 79.80% |

**Tabela 6 - Análise Sumarizada do Atributo "Visitado"**

| Tipo de Frase | Número Total de "Data da Visita" | Número de Acertos | % de Acertos |
|---|---|---|---|
| Verbo no Particípio | 9 | 8 | 88.89% |
| Verbo com Ênclise | 2 | 2 | 100.00% |
| Substantivo "Visita" | 18 | 15 | 83.33% |
| Casos Normais | 56 | 54 | 96.43% |

**Tabela 7 - Análise Sumarizada do Atributo "Data da Visita"**

A Tabela 8 abaixo apresenta o número de vezes que cada pronome aparece no conjunto de testes (Número de Ocorrências) e quantas vezes o algoritmo acertou (Total de Acertos), lembrando que acerto aqui é se o algoritmo conseguiu localizar o termo antecedente correto, de acordo com o especificado em Cuevas & Paraboni (2008).

| Pronome | Número de Ocorrências | Total de Acertos | Percentual de Acertos |
|---|---|---|---|
| Ele | 51 | 32 | 62.75% |
| Eles | 8 | 4 | 50.00% |
| Ela | 3 | 3 | 100.00% |
| Elas | 0 | 0 | - |
| **Total** | **62** | **39** | **62.90%** |

**Tabela 8 - Taxas de Acerto da Resolução de Pronomes**

## 3 - Discussão dos Resultados

### Dados Coletados

Foi preparado um conjunto de testes (tabela 3) contendo frases divididas nas seguintes categorias: frases com verbo no particípio, frases com ênclise verbal, frases com o substantivo “visita” e por fim a categoria de frases normais. Foram selecionadas aleatoriamente algumas frases para cada categoria. As categorias “particípio” e “ênclise” participam 100% do conjunto de testes, isto é, todas as frases existentes na coleção textual foram escolhidas para o conjunto de testes, por serem em menor número. As outras duas categorias contribuem com cerca de 15% para os casos normais e 5 % para o substantivo “visita”.

Na tabela 4 estão sumarizados os números de ocorrência dos atributos "visitante", "visitado" e "data da visita" existentes no conjunto de testes. Esta tabela também está dividida nas quatro categorias citadas.

As tabelas 5, 6 e 7 exibem as taxas de acerto do algoritmo para os atributos "visitante", "visitado" e "data da visita" respectivamente. Algumas vezes o algoritmo não encontra algum atributo na frase, como "visitante" por exemplo, para estes casos o atributo é marcado como "nulo". Os casos "nulos" também estão sumarizados nestas tabelas ou como acerto (caso o algoritmo tenha classificado como nulo e realmente o atributo é nulo na frase) ou como erro (caso o algoritmo classifique como nulo, sendo que o atributo existe na frase).

A tabela 8 apresenta as taxas de acerto do módulo de resolução de pronomes do trabalho anterior (Passavaz, 2008) e que foi integrado a este trabalho para resolver os casos nos quais algum dos atributos (visitante ou visitado) sejam pronomes em terceira pessoa (Ele, Ela, Eles, Elas). Segue abaixo um dos casos em que o algoritmo resolveu o pronome "Eles" corretamente:

Frase Anterior:

*"Segundo o delegado titular do 30º DP, são gastos R$=15=mil mensais com a alimentação dos presos, que chega a incluir camarão e frutas de sobremesa."*

Frase com o evento "visita":

*" Eles (ref = presos) ficam das 10h às 5h no pátio , todos=os=dias , tomando sol e jogando futebol , além de receberem visitas íntimas , disse ."*

Nos dois exemplos acima, a segunda frase contém o evento "visita", como visitante o algoritmo extraiu o pronome "Eles" e logo em seguida recuou para a frase anterior para tentar resolver o pronome encontrado e neste caso referenciou o pronome "Eles" a "presos" que está totalmente correto.

**Análise dos Resultados Obtidos**

Segue abaixo uma análise detalhada das Tabelas 5 a 8, que têm por objetivo explicar as porcentagens atingidas baseando-se nos pontos fortes e fracos do algoritmo desenvolvido.

**Atributo "Visitante" (Tabela 5) –** As frases na categoria "casos normais" possuem uma maior amostragem no conjunto de testes para o atributo "visitante" e obtiveram uma taxa de acerto de 94,68%. A categoria ênclise obteve 100% de taxa de acerto, porém possui apenas 3 ocorrências no conjunto textual inteiro, portanto não é um valor representativo devido sua baixa amostragem.

A categoria "particípio" obteve uma taxa relativamente baixa de acerto porque o "visitante" para frases desta categoria geralmente é um objeto e encontra-se após o verbo, lembrando que aqui a frase está na voz passiva (visitante e visitado estão em sentido inverso na frase).

O algoritmo demonstrou taxas menores de acerto na extração de objetos diretos e indiretos, pois a qualidade das etiquetas do Corpus Textual para objetos é inferior quando comparada às etiquetas de sujeito.

A categoria "substantivo visita" apresentou boas taxas de acerto para as frases coletadas. As taxas de acerto para esta categoria só foram altas porque foram utilizadas apenas duas regras para extração desta categoria: [receber] + visitar e [fazer] + visitar, sendo por conta disso uma taxa de amostragem pequena (apenas 5 %). Esta categoria apresenta uma variedade muito grande quanto às formas com que aparecem o substantivo visita.

**Atributo "Visitado" (Tabela 6) –** As taxas de acerto nesta categoria seguiram a mesma tendência da Tabela 5. A única categoria que obteve uma taxa de acerto alta foi a categoria "particípio", pois se baseia nas etiquetas de sujeito para localizar o "visitado". A categoria "ênclise" aparece com apenas 33%, mas continua tendo uma amostragem ainda pequena para se obter conclusões. As outras categorias obtiveram uma taxa de acerto pior por se basearem nas etiquetas de objeto para identificar o "visitado".

**Atributo "Data da Visita" (Tabela 7)** – O atributo "data da visita" obteve altos índices de acerto pois as etiquetas referentes à data e hora na coleção textual apresentam alta precisão. Aqui em "data da visita" são consideradas também expressões adverbiais de tempo como "assim que choveu" e outras.

**Resolução de Pronomes (Tabela 8) -** As taxas de acerto para a resolução de pronomes aproximaram-se bastante dos resultados obtidos no trabalho anterior (Passavaz, 2008). Foi utilizado o mesmo código fonte (apenas adaptado a este trabalho) e também a mesma coleção textual, era esperado uma porcentagem de acerto semelhante.

## 5. Conclusões

Este trabalho explorou algumas das técnicas utilizadas pela área Extração de Informação (EI) desenvolvendo para isto uma implementação prática que trata desde a obtenção de textos da WEB (módulo 1), passando pela análise morfossintática e etiquetamento (módulo 2, Bick 2000) e finalmente pela extração de templates relacionados ao evento "visita" (módulo 3). Trabalhando juntos, estes três módulos, são capazes de capturar um texto não estruturado (não tratado) da WEB, realizar o tratamento e capturar a informação desejada em forma de templates, o que torna a informação possível de ser inserida em um sistema de banco de dados relacional (SGBD).

Com o contínuo aumento do volume de informação tanto na Web quanto em uma intranet corporativa, por exemplo, torna-se cada vez mais necessário a existência e aperfeiçoamento de sistemas deste tipo que consigam transformar os dados existentes na rede em informações úteis para a sociedade ou para empresas diversas.

Conforme planejado inicialmente, os três módulos citados acima não estão trabalhando de maneira integrada na implementação desenvolvida, isto foi necessário para que fosse possível a conclusão deste trabalho no prazo determinado. Foram testados cada módulo separadamente e seria interessante em

um trabalho futuro a integração destes módulos para que os mesmos constituam um sistema completo de extração de informação.

Destes três módulos citados, a extração de informação (módulo 3) foi o mais custoso, tanto em tempo como em complexidade, foi realizada a implementação do evento "visita" contemplando os seguintes atributos: nome do visitante, nome do visitado e data da visita. Outros atributos poderiam ser extraídos, tais como: o motivo da visita, dados específicos sobre o visitante e/ou visitado, tais como profissão, cargo, idade e outros atributos que aparecem nos textos, mas por motivo de escopo foram escolhidos apenas os três atributos citados.

Já a obtenção de textos da WEB (módulo 2), possui uma complexidade relativamente alta, porém sua implementação é muito simples pois se utiliza de rotinas prontas sendo algumas delas da própria linguagem Python e outras do Natural Language Toolkit (NLTK). Possui como tarefa a varredura da Web e armazenamento dos textos para serem processados futuramente.

É importante ressaltar que este trabalho teve uma integração com o trabalho anterior (Passavaz, 2008), aplicando os conceitos de resolução de pronomes para identificar os visitantes e visitados nas frases em que estes eram pronomes em terceira pessoa (Eles, Elas, Ele, Ela). Boa parte do código fonte pôde ser aproveitado, dado que foi utilizada a mesma linguagem (Python) e o mesmo conjunto textual.

## 5.3 – Contribuições deste trabalho

Este trabalho contribui com a área de EI ao apresentar uma possível implementação de um módulo de obtenção de textos não estruturados, i.e., não tratados, da Web através da varredura de links entre as páginas HTML iniciando por um endereço raiz (página inicial) contendo um processo de filtragem que permite ao usuário a remoção de páginas indesejadas.

Quanto ao módulo de extração de informação baseado em templates (módulo 3), este trabalho colabora com a apresentação de uma possível implementação

prática de um sistema de extração de templates, utilizando para isso a construção de regras capazes de extrair eventos relacionados ao termo “visita”.

## 5.4 – Experiências Adquiridas

Através da execução deste trabalho foi possível um aprimoramento tanto da linguagem Python quanto do kit NLTK vêm sendo utilizados desde o trabalho anterior (Passavaz, 2008). Através da leitura bibliográfica, foi possível o conhecimento de várias técnicas e metodologias utilizadas na área de Extração de Informação (EI) para se realizar processamento textual, também foi possível o aprendizado de um algoritmo capaz de navegar entre várias páginas Web extraindo o seu conteúdo (texto) e armazenando em arquivos locais.

## 5.5 – Sugestões de Trabalhos Futuros

Seguem algumas sugestões para futuros trabalhos na área de EI.

- ✓ Estudo e implementação de maneiras de se realizar processamento textual sobre coleções de textos que não possuem qualquer anotação morfossintática. Pode ser muito útil o processamento de textos diretamente da Web sem passar por um módulo Part-of-Speech (POS) responsável por anotar o texto (pré-processamento), tal como é o PALAVRAS (Bick, 2000).
- ✓ Aprimoramento do módulo de resoluções de pronomes para aumentar sua precisão ou abrangência.
- ✓ Utilização de técnicas de aprendizagem de máquina para descobrir automaticamente as regras de extração, sem necessidade de criá-las caso a caso como foi feito neste trabalho.
- ✓ Realizar uma integração entre os dois módulos desenvolvidos neste trabalho, o módulo de busca de textos na web com o módulo de extração de eventos, constituindo assim uma aplicação completa de EI.

# Apêndice

| | |
|---|---|
| Etiqueta de Sujeito: | @SUBJ> e @<SUBJ |
| Descrição: | A função desta etiqueta é identificar uma palavra como o sujeito da frase em questão. Esta etiqueta foi utilizada no trabalho para ajudar a identificar o agente da visita (visitante) |
| Características: | A seta > ou < indica a direção do verbo principal com o qual se relaciona a palavra que contém esta etiqueta |

| | |
|---|---|
| Etiqueta de Objeto Direto e Indireto: | @ACC> @<ACC E @ACC>> |
| Descrição: | Esta etiqueta tem por função identificar objetos diretos e indiretos em uma frase. Foi utilizada neste trabalho para ajudar na localização do visitado, que geralmente são objetos e referenciam o verbo visitar. |

| | |
|---|---|
| Etiqueta de Verbo: | V |
| Descrição: | Indica que uma palavra é um Verbo e foi utilizada em diversos pontos do trabalho durante a criação de regras para extração. |

| | |
|---|---|
| Etiqueta Pessoa | 1P 2P 3P |
| Descrição: | Indica em que pessoa um verbo se encontra 1P, 2P ou 3P. Foi utilizada neste trabalho para tratar casos onde o sujeito é composto, ou seja, se o verbo estiver no plural, identificado pela letra P (após os números 1, 2 e 3). |

| | |
|---|---|
| Etiqueta Substantivo | N |
| Descrição: | Indica que uma palavra é um substantivo e foi utilizada em alguns pontos da criação de regras para extração. |

| | |
|---|---|
| Etiqueta Enclise | <hyfen> |
| Descrição: | Indica que um verbo possui uma ênclise (eg, visitá-lo), e foi utilizada no trabalho para identificar este tempo verbal. |

| | |
|---|---|
| Etiqueta Tempo Verbal | PCP |
| Descrição: | Indica que um verbo está no Particípio (eg, visitado) e foi utilizado no trabalho para regras específicas para este tipo de verbo. |

| | |
|---|---|
| Etiqueta Entidade | PROP, <tit>, PERS, <H>, <HH>, <Hattr>, <Hbio>, <Hideo>, <Hmyth>, <Hnat>, <Hprof>, <Hfam>, <sick>, <Hsick>, <Htit>, <B>, <BB>, <Btree>, <Bflo>, <Bbush>, <fruit>, <Bveg>, <A>, <AA>, <Adom>, <AAdom>, <Aich>, <Amyth>, <Azo>, <Aorn>, <Aent>, <Acell>, <hum>, <official>, <member>, <groupind>, <groupofficial>, <grouporg>, <author>, <absname>, <object>, <common>, <mat>, <plant> |
| Descrição: | Estas etiquetas identificam uma Entidade que pode ser uma pessoa, uma planta, um grupo de pessoas e etc ... Estas anotações foram retiradas do manual que acompanha o conjunto de textos. Foram utilizadas neste trabalho para auxiliar na identificação do agente da visita (visitante). |

| | |
|---|---|
| Etiqueta Locais | <L>,<Labs>,<Lciv>,<Lcover>,<Lh>, <Lopening>, <Lpath>, <Lstar>, <Lsurf>, <Ltip>, <Ltop>, <Ltrap>, <Lwater>, <bar>, <build>, <top>, <civ>, <address>, <site>, <virtual>, <astro>, <road>, <school> |
| Descrição: | Estas etiquetas identificam um Local, tal como uma país, uma cidade, estabelecimentos comerciais, e etc ... A utilização destas etiquetas neste trabalho é para identificar o "visitado", que pode ser um Local. |

| | |
|---|---|
| Etiqueta Organização | <org>, <inst>, <suborg>, <admin>, <media>, <party> |
| Descrição: | Estas etiquetas identificam organizações. A utilização destas etiquetas neste trabalho é para identificar o "visitado", que pode ser uma Organização. |

| | |
|---|---|
| Etiqueta Data Hora | <temp>, <date>, <period>, <hour>, <dur>, <cyclic> |
| Descrição: | Estas etiquetas identificam datas e horas e foram utilizadas para auxiliar na extração do atributo "data da visita". |

**Tabela 9 - Etiquetas utilizadas para a extração do evento visita**

# Referências Bibliográficas


BICK, Eckhard. The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Dr.phil. thesis. Aarhus University. Aarhus, Denmark: Aarhus University Press. November 2000.

CORRÊA, Adriana Cristina Giusti (2003) – Recuperação de Documentos baseada em Informação Semântica no Ambiente AMMO – Agosto 2003 – 102 f. – Pós Graduação em Ciência da Computação - Dissertação de Mestrado – Universidade Federal de São Carlos - UFSCAR.

CUEVAS, Ramon Ré Moya e PARABONI, Ivandré (2008) A Machine Learning Approach to Portuguese Pronoun Resolution. 11th Ibero-American Conference on Artificial Intelligence(IBERAMIA-2008). LNAI vol. 5290, pp. 262-271. Springer-Verlag Berlin Heidelberg.

FELDMAN, Ronen; SANGER, James. (2006) The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Primeira Edição. New York. Cambridge University Press.

KONCHADY, Manu. (2006) Text Mining Application Programming: Programming Series. Primeira Edição. Boston. Massachusetts.

LEITÃO, Daniel Almeida; TORRES, Dante Gama (2005) – Resumo do Seminário sobre Métricas de Avaliação em Extração de Informação – Mestrado em Ciência da Computação – Universidade Federal de Pernambuco – UFPE.

LOH, Stanley; WIVES Leandro Krug; OLIVEIRA, José Palazzo. (2000) Descoberta Pró Ativa de Conhecimento em Coleções Textuais: Iniciando Sem Hipóteses – Artigo – 12 f. – Universidade Federal do Rio Grande do Sul – UFRGS.
http://www.leandro.wives.nom.br/publicacoes/OIA2000.pdf
Acesso em: 21 de janeiro de 2009.

LOPES, Maria Célia Santos (2004) – Mineração de Dados Textuais utilizando Técnicas de Clustering para o Idioma Português – Outubro 2004 – 191 f. – Pós Graduação em Engenharia Civil – Tese de Doutorado – Universidade Federal do Rio de Janeiro – UFRJ.

MUC7 – Message Understanding Conference. (1997). Site Mantido pela Instituição NIST, EUA.
http://www.itl.nist.gov/iaui/894.02/related_projects/muc/
Acesso em: 21 de janeiro de 2009.

NLTK - Natural Language Toolkit – Kit de desenvolvimento para aplicações PLN na linguagem Python.
http://nltk.sourceforge.net/index.php/Main_Page
Acesso em: 21 de janeiro de 2009.

PASSAVAZ, Victor Hugo Jabur (2008) – Uso da Ferramenta NLTK e Python para desenvolvimento de aplicações de Processamento de Linguagem Natural – Trabalho de Graduação – Universidade de São Paulo – Escola de Ciências, Artes e Humanidades – USP - EACH

WITTEN, I. H. & E. Frank (2005) Data Mining: Practical machine learning tools and techniques. 2nd edition Morgan Kaufmann, San Francisco.