

WAGNER TOSCANO

**MINERADOR WEB: UM ESTUDO  
SOBRE MECANISMOS DE DESCOBERTA  
DE INFORMAÇÕES NA WEB**

Dissertação apresentada à Escola Politécnica da  
Universidade de São Paulo para a obtenção do  
título de Mestre em Engenharia.

São Paulo  
2003

WAGNER TOSCANO

**MINERADOR WEB: UM ESTUDO  
SOBRE MECANISMOS DE DESCOBERTA  
DE INFORMAÇÕES NA WEB**

Dissertação apresentada à Escola Politécnica da  
Universidade de São Paulo para a obtenção do  
título de Mestre em Engenharia.

Área de Concentração:  
Engenharia Elétrica

Sub-área:  
Sistemas Digitais

Orientador:  
Prof. Dr. Edson Satoshi Gomi

São Paulo  
2003

## FICHA CATALOGRÁFICA

Toscano, Wagner

MINERADOR WEB: um estudo sobre mecanismos de descoberta de informações na Web. São Paulo, 2003.

117p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais

1. Web Mining 2. Representação do Conhecimento 3. Ontologia 4. Descoberta do Conhecimento 5. Extração de Informação 6. Inteligência Artificial I. Universidade de São Paulo. Escola Politécnica. Departamento de Computação e Sistemas Digitais II.t.

À minha família que tanto me apoiou  
em minha busca pelo conhecimento.

## AGRADECIMENTOS

Ao amigo e orientador Professor Doutor Edson Satoshi Gomi que me mostrou uma nova porta para a compreensão do mundo real.

Aos professores e amigos do PCS que sempre estiveram de portas abertas para discutir os assuntos relacionados com o tema deste trabalho.

Ao amigo André Wakamatsu, que me orientou na utilização do  $\text{\LaTeX}$ , com o qual escrevi esta dissertação.

A todos os meus amigos do LTI, que sempre me ajudaram e animaram.

Aos funcionários da biblioteca da Engenharia Elétrica, por me ajudarem quando eu não conseguia o que queria em minhas buscas por livros.

À Elisabeth por me ajudar na formatação das referências bibliográficas.

À minha neta, Luana, que não poderia ter chego em um momento mais oportuno.

À minha esposa, Maria Lucia, pelo carinho, paciência e motivação que me deu para a continuação dos estudos.

A todos vocês, muito obrigado.

## RESUMO

A Web (WWW - World Wide Web) possui uma grande quantidade e variedade de informações. Isso representa um grande atrativo para que as pessoas busquem alguma informação desejada na Web. Por outro lado, dessa grande quantidade de informações resulta o problema fundamental de como descobrir, de uma maneira eficaz, se a informação desejada está presente na Web e como chegar até ela. A existência de um conjunto de informações que não se permitem acessar com facilidade ou que o acesso é desprovido de ferramentas eficazes de busca da informação, inviabiliza sua utilização. Soma-se às dificuldades no processo de pesquisa, a falta de estrutura das informações da Web, que dificulta a aplicação de processos na busca da informação.

Neste trabalho é apresentado um estudo de técnicas alternativas de busca da informação, pela aplicação de diversos conceitos relacionados à recuperação da informação e à representação do conhecimento. Mais especificamente, os objetivos são analisar a eficiência resultante da utilização de técnicas complementares de busca da informação, em particular mecanismos de extração de informações a partir de trechos explícitos nos documentos HTML e o uso do método de Naive Bayes na classificação de *sites*, e analisar a eficácia de um processo de armazenamento de informações extraídas da Web numa base de conhecimento (descrita em lógica de primeira ordem) que, aliada a um conhecimento de fundo, permita responder a consultas mais complexas que as possíveis por meio do uso de expressões baseadas em palavras-chave e conectivos lógicos.

## ABSTRACT

The World Wide Web (Web) has a huge amount and a large diversity of informations. There is a big appeal to people navigate on the Web to search for a desired information. On the other hand, due to this huge amount of data, we are faced with the fundamental problems of how to discover and how to reach the desired information in a efficient way. If there is no efficient mechanisms to find informations, the use of the Web as a useful source of information becomes very restrictive. Another important problem to overcome is the lack of a regular structure of the information in the Web, making difficult the use of usual information search methods.

In this work it is presented a study of alternative techniques for information search. Several concepts of information retrieval and knowledge representation are applied. A primary goal is to analyse the efficiency of information retrieval methods using analysis of extensional information and probabilistic methods like Naive Bayes to classify sites among a pre-defined classes of sites. Another goal is to design a logic based knowledge base, in order to enable a user to apply more complex queries than queries based simply on expressions using keywords and logical connectives.

# Sumário

<b>RESUMO</b>	<b>iv</b>
<b>“ABSTRACT”</b>	<b>v</b>
<b>Lista de Figuras</b>	<b>viii</b>
<b>Lista de Tabelas</b>	<b>x</b>
<b>Lista de Algoritmos</b>	<b>xii</b>
<b>Lista de Abreviaturas e Siglas</b>	<b>xiii</b>
<b>1 INTRODUÇÃO</b>	<b>1</b>
1.1 Processo tradicional de busca da informação na WEB . . . . .	2
1.2 Objetivos . . . . .	8
1.3 Organização . . . . .	8
<b>2 REPRESENTAÇÃO DO CONHECIMENTO</b>	<b>9</b>
2.1 O conhecimento . . . . .	9
2.2 Ontologia . . . . .	14
2.2.1 Propriedades da ontologia . . . . .	14
2.2.2 Parâmetros para construção de uma ontologia . . . . .	16
2.3 Linguagens . . . . .	17
2.3.1 Linguagens de marcação . . . . .	17
2.3.2 Linguagem XML . . . . .	19
2.3.3 Linguagem HTML . . . . .	22
2.3.4 Linguagem lógica . . . . .	23
2.4 Base de conhecimento . . . . .	28
2.5 Estimadores probabilísticos . . . . .	30

2.5.1	Método Naive Bayes . . . . .	30
2.5.2	Informação Mútua Média . . . . .	33
2.5.3	Método Naive Bayes com suavização . . . . .	35
<b>3</b>	<b>MINERADOR WEB</b>	<b>39</b>
3.1	A arquitetura do Minerador Web . . . . .	39
3.2	Processo de geração da meta informação . . . . .	41
3.2.1	Coleta de <i>sites</i> . . . . .	41
3.2.2	Categorização . . . . .	42
3.2.3	Criação da ontologia de empresas . . . . .	49
3.2.4	Criação da DTD . . . . .	53
3.2.5	Criação do conjunto de treinamento . . . . .	55
3.2.6	Treinamento do Classificador . . . . .	66
3.3	Processo de busca da informação . . . . .	83
3.3.1	Extração do elemento “ <i>Site</i> ” . . . . .	85
3.3.2	Extração do elemento “Nome da Empresa” . . . . .	85
3.3.3	Extração do elemento “Endereço” . . . . .	88
3.3.4	Extração do elemento “Comunicação” . . . . .	92
3.3.5	Extração dos elementos de “Produto” . . . . .	93
3.3.6	Extração do elemento “Ramo de Atividade” e “Atividade”	97
3.3.7	Conversor de aplicação XML para linguagem lógica . . . . .	97
<b>4</b>	<b>RESULTADOS EXPERIMENTAIS</b>	<b>103</b>
4.1	Performance do classificador resultante . . . . .	103
4.2	Base de conhecimento resultante da aplicação do Minerador Web .	108
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>110</b>
	<b>Referências</b>	<b>113</b>

## Lista de Figuras

1.1	Exemplo de relação de endereços de <i>sites</i> , fornecida como resultado da aplicação da expressão lógica “ <i>car or tire or sales</i> ” ao mecanismo de busca do portal Yahoo . . . . .	7
2.1	Estrutura de um agente no nível do conhecimento. . . . .	12
2.2	Aplicação de marcação estruturada. . . . .	18
2.3	Exemplo de uma DTD para contatos de empresas . . . . .	21
2.4	Exemplo de uma instância de aplicação XML, com base na DTD para contatos de empresas . . . . .	22
2.5	Código fonte parcial de uma página HTML . . . . .	24
2.6	Documento XML representado em Prolog . . . . .	27
2.7	O problema da classificação de <i>sites</i> . . . . .	32
3.1	Arquitetura do Minerador Web . . . . .	40
3.2	Texto extraído do código do <i>site</i> “ <i>www.stargate.com</i> ”, em formato original . . . . .	43
3.3	Texto extraído do código do <i>site</i> “ <i>www.stargate.com</i> ”, com as marcações removidas . . . . .	43
3.4	Texto extraído do código do <i>site</i> “ <i>www.stargate.com</i> ”, com a simbologia das marcações . . . . .	48
3.5	DTD da ontologia de empresas . . . . .	54
3.6	Informações obtidas, em formato de código HTML, da categoria <i>Shopping and Service</i> . . . . .	57
3.7	Código do método de seleção de números aleatórios. . . . .	58
3.8	Distribuição dos <i>sites</i> obtidos para “Ramo de Atividade” . . . . .	63
3.9	Distribuição das palavras únicas por categoria para “Ramo de Atividade” . . . . .	65
3.10	Percentual de distribuição do total das palavras do conjunto de treinamento . . . . .	66

3.11	Distribuição das palavras no conjunto de treinamento por faixa de Informação Mútua Média (AMI) . . . . .	67
3.12	Distribuição de acertos por classificador <i>Naive Bayes</i> . . . . .	69
3.13	Distribuição de acertos por classificador Minerador Web . . . . .	70
3.14	Distribuição percentual dos <i>sites</i> por categoria para “Atividade” .	76
3.15	Percentual das palavras únicas distribuídas na categoria “Atividade”	77
3.16	Percentual de distribuição da quantidade total das palavras do conjunto de treinamento “Atividade” . . . . .	78
3.17	Distribuição das palavras no conjunto de treinamento por faixa de Informação Mútua Média (AMI) . . . . .	80
3.18	Distribuição das quantidades de palavras no conjunto de treinamento para a categoria “Atividade”, após a aplicação da AMI . .	83
3.19	Código JavaScript em uma página HTML . . . . .	84
3.20	Aplicação XML, resultado da extração de informações HTML . .	86
3.21	Representação de um endereço de uma aplicação XML em um fato na linguagem Prolog . . . . .	98
3.22	Representação de regras em Prolog, relativas ao fato da figura 3.21	99
3.23	Código Prolog contendo exemplo de conhecimento de fundo e sua aplicação . . . . .	99
3.24	Exemplo de conhecimento de fundo, implementado em Prolog . .	102
4.1	Resultados obtidos com a utilização do Minerador Web no primeiro conjunto de teste . . . . .	105
4.2	Resultados obtidos com a utilização do Minerador Web no segundo conjunto de teste . . . . .	106
4.3	Referência ao conteúdo dos fatos da base de conhecimento . . . .	108
4.4	Relação parcial de fatos da base de conhecimento gerada . . . . .	109

## Lista de Tabelas

1.1	Resultados quantitativos referente à busca de informações nos portais	4
2.1	Classificação das primitivas utilizada no formalismo da representação do conhecimento ([29] pag.96)	11
2.2	Conectivos lógicos do Cálculo de Predicados	25
2.3	Quantificadores do cálculo de predicados	26
3.1	Linhas resultante da aplicação do utilitário de extração de informações gerais	45
3.2	Percentual da marcação “title” nos documentos HTML analisados	46
3.3	Percentual da marcação “meta” nos documentos HTML analisados	46
3.4	Percentual de diversas palavras significativas	47
3.5	Resultado, parcial, da etapa de categorização	49
3.6	Exemplo da estrutura de uma referência a endereço de <i>site</i> , obtida no portal Yahoo	57
3.7	Exemplo de linhas irrelevantes, capturadas no processo de coleta de informações no <i>site</i>	58
3.8	Relação de palavras que podem compor um Stop-Word	60
3.9	Resumo dos resultados obtidos na categorização de 10 categorias	61
3.10	Quantidade de <i>sites</i> obtidos por categoria para “Ramo de Atividade”	62
3.11	Quantidade de palavras únicas por categoria para “Ramo de Atividade”	64
3.12	Distribuição da quantidade total das palavras do conjunto de treinamento	65
3.13	Valores de AMI por palavra por categoria	66
3.14	Distribuição das palavras por faixa de valores da AMI	68
3.15	Distribuição dos <i>sites</i> selecionados para o conjunto de teste	68

3.16	Quantidade de acertos do classificador <i>Naive Bayes</i> . . . . .	69
3.17	Quantidade de acertos do classificador Minerador Web . . . . .	70
3.18	Comparação entre as quantidades de palavras no conjunto de treinamento, para a categoria “Ramo de Atividade”, antes e após a seleção da faixa da AMI . . . . .	71
3.19	Quantidade de <i>sites</i> por categoria para “Atividade” . . . . .	76
3.20	Quantidade de palavras únicas por categoria para “Atividade” . . . . .	77
3.21	Distribuição da quantidade total das palavras do conjunto de treinamento “Atividade” . . . . .	78
3.22	Valores de AMI por palavra por categoria . . . . .	78
3.23	Distribuição das palavras por faixa de valores da AMI para o novo agrupamento de “Atividade” . . . . .	79
3.24	Composição quantitativa de “Atividade” com referência ao novo agrupamento de categorias . . . . .	80
3.25	Valores de AMI por palavra por categoria, com o novo agrupamento de “Atividade” . . . . .	81
3.26	Distribuição das quantidades de palavras no conjunto de treinamento, para a categoria “Atividade”, após a aplicação da AMI . . . . .	82
3.27	Conversão de uma aplicação XML em Prolog . . . . .	101
4.1	Distribuição dos <i>sites</i> selecionados para o primeiro conjunto de teste.	103
4.2	Resultados obtidos com o Minerador Web, para informação implícita	104
4.3	Distribuição dos <i>sites</i> por elemento no conjunto de teste . . . . .	105
4.4	Resultados obtidos com o Minerador Web, para informação explícita	105

## Lista de Algoritmos

3.1	Remoção de marcações . . . . .	64
3.2	Classificação de um <i>site</i> em uma classe de uma categoria . . . . .	72
3.3	Carga da “ontologia” de uma categoria - Declarações . . . . .	73
3.4	Carga da “ontologia” de uma categoria . . . . .	74
3.5	Extração do elemento “Nome da Empresa”, por termos . . . . .	87
3.6	Extração do elemento “Nome da Empresa”, pela marcação “< <i>title</i> >” . . . . .	87
3.7	Extração do elemento “Nome da Empresa”, pela quantidade de ocorrência do nome” . . . . .	87
3.8	Extração do elemento “Nome da Empresa”, pela freqüência das palavras . . . . .	89
3.9	Extração do elemento “Endereço”, atributo “UF” . . . . .	89
3.10	Extração do elemento “Endereço”, atributo “ZPT” . . . . .	90
3.11	Extração do elemento “Endereço”, atributo “Cidade” . . . . .	91
3.12	Extração do elemento “Endereço”, atributo “Complemento” . . . . .	91
3.13	Extração do elemento “Endereço”, atributo “Via” . . . . .	92
3.14	Extração do elemento “Comunicação”, atributos “Telefone” e “Fax”	93
3.15	Extração do elemento “Comunicação”, atributo “e-mail” . . . . .	94
3.16	Extração do elemento “Produto” e dos atributos “Nome do Pro- duto” e “Nome da Especialização” . . . . .	95
3.17	Extração do elemento “Produto”, atributo “Característica” . . . . .	96
3.18	Extração do elemento “Produto”, atributo “Preço” . . . . .	97

## Lista de Abreviaturas e Siglas

AMI	<i>Average Mutual Information</i>
CEP	Código de Endereçamento Postal
DTD	<i>Document Type Definition</i>
HTML	<i>HyperText Markup Language</i>
SGML	<i>Standard Generalized Markup Language</i>
URL	<i>Uniform Resource Locators</i>
W3C	<i>World Wide Web Consortium</i>
WWW	<i>World Wide Web</i>
XML	<i>eXtensible Markup Language</i>

# Capítulo 1

## INTRODUÇÃO

A Web (WWW - World Wide Web) possui uma grande quantidade e variedade de informações. Isso representa um grande atrativo para que as pessoas busquem alguma informação desejada na Web. Por outro lado, dessa grande quantidade de informações resulta o problema fundamental de como descobrir, de uma maneira eficiente, se a informação desejada está presente na Web e como chegar até ela.

A existência de um conjunto de informações que não se permitem acessar com facilidade ou que o acesso é desprovido de ferramentas eficazes de busca da informação, inviabiliza sua utilização. Ferramentas de busca necessitam receber parâmetros descritivos da informação desejada. Uma forma usual de se descrever a informação desejada é com a utilização de palavras-chave. A idéia do uso de palavras-chave é tentar caracterizar a informação desejada por meio de termos que tem alta probabilidade de ocorrência nos documentos de interesse. A dificuldade é que o uso de muitas palavras-chave ou que tenham baixa probabilidade de ocorrência no contexto desejado, fará com que a ferramenta de busca devolva poucas ou nenhuma referência a documento. Por outro lado, se os critérios de busca da informação são demais flexíveis, os resultados obtidos são em grande quantidade, necessitando de trabalho redobrado na análise dos resultados, ou não representam a informação desejada. Soma-se a essas particularidades do critério de busca, a falta de estrutura das informações da Web, que dificultam a aplicação de processos na busca da informação.

Por essa visão, este trabalho tem como proposta apresentar um estudo de técnicas alternativas de busca da informação, pela aplicação de diversos conceitos relacionados à recuperação da informação e à representação do conhecimento.

Neste primeiro capítulo, serão apresentados os aspectos reflexivos iniciais que norteiam este trabalho, que tem como tema a recuperação da informação de documentos semi-estruturados, encontrados na Web, e sua transformação em documentos estruturados que permitirão serem representados em uma base lógica de conhecimento.

## 1.1 Processo tradicional de busca da informação na WEB

A Web possui uma grande quantidade de informações distribuídas principalmente na forma de documentos HTML. Esses documentos estão relacionados a um endereço na Web, denominados de *sites*.

Em geral esses documentos possibilitam a aplicação de operações como inclusão, alteração, busca e recuperação de informações.

Todavia, embora seja a Web um elemento facilitador na divulgação da informação, o processo de busca é dificultado pelo crescimento contínuo da quantidade desta informação. Para minimizar essa dificuldade são disponibilizados nos portais de busca como Yahoo [47], Altavista [1] ou Netscape [28], mecanismos que auxiliam na busca da informação.

Esses mecanismos de busca, exigem que o usuário inicie sua pesquisa formulando uma sentença que descreva suas necessidades de informação, como por exemplo na sentença da expressão 1.1.

“Localizar *sites* de revendedores de pneus de carro.” (1.1)

Para os mecanismos de busca atuarem e fornecerem resultados adequados às necessidades de informação do usuário, se faz necessário o fornecimento de uma palavra ou uma expressão com algumas características importantes:

1. A expressão deve conter palavras que caracterizem a informação desejada, tais como “revendedor” e “pneu”;
2. Não é necessário descrever a ação a ser realizada pela ferramenta de busca. Por exemplo, como a ação desejada é localizar *sites*, não há necessidade de constar essa palavra na expressão. O resultado dessa eliminação pode ser visualizado na expressão 1.2.

“de revendedores de pneus de carro.” (1.2)

3. Não é necessário incluir palavras desprovidas de informação semântica significativa, como exemplo as preposições “de”. Sendo assim, as palavras que normalmente compõem a expressão necessária aos mecanismos de busca da

informação são substantivos, ou sejam, termos nominativos. Na expressão 1.3 pode-se visualizar o resultado da eliminação das preposições, sobre a expressão 1.2.

“revendedores pneus carro.” (1.3)

Essas expressões, compostas de termos nominativos, não representam as relações entre as palavras. Porém, esses relacionamentos são possíveis de serem descritos com a inclusão de conectivos lógicos como: “*and*”, “*or*”, “*not*” e “*near*”, e suas combinações como “... *and* ... *or* ... *and* ...”. Vale ressaltar que, a inclusão desses conectivos é permitida nos mecanismos de busca da informação e que os portais de busca disponibilizam auxílio de como utilizar esses conectivos. O resultado da aplicação de alguns conectivos na expressão 1.3, pode ser visualizado na expressão 1.4.

“revendedores *and* pneus *and* carro”. (1.4)

As expressões compostas desses conectivos são denominadas expressões lógicas.

A construção de expressões lógicas com palavras de significados afins aos substantivos, com o objetivo de ampliar o sentido da expressão (sinonímia), auxiliam e enriquecem os resultados da busca. Sinonímia refere-se a palavras que fornecem uma noção mais ampla em relação a um termo. Por exemplo, aplicando o conceito de sinonímia ao termo substantivo “revendedores”, obtém-se palavras como: lojas, vendas, automóveis.

Na tabela 1.1, podem ser visualizadas a relação de quantidades de endereços de *sites* resultantes da utilização dos mecanismos de busca da informação nos portais Yahoo, Altavista e Netscape.

A construção das expressões lógicas utilizando os conectivos lógicos, fornecem orientação aos mecanismos de busca da informação para selecionarem os *sites*. Essa orientação segue critérios comuns, independente do portal utilizado.

Com referência à tabela 1.1, vale observar que os critérios de busca associados aos mecanismos de busca com relação às expressões lógicas fornecidas são:

- Utilização do conectivo lógico “*and*” para relacionamento dos termos nominativos: orientam os mecanismos de busca da informação a selecionarem os

Linha	Expressões lógicas — Portais	Yahoo	Altavista	Netscape
1	<i>car and tire and sales</i>	171.000	53.161	151.003
2	<i>car or tire or sales</i>	4.790.000	40.971.648	3.330.003
3	<i>car tire sales</i>	171.000	53.161	151.003
4	<i>“car tire sales”</i>	10	9	19
5	<i>car and tire and store</i>	247.000	70.688	204.002
6	<i>car or tire or store</i>	4.810.000	3.342.438	3.240.002
7	<i>car tire store</i>	247.000	70.688	205.002
8	<i>“car tire store”</i>	4	4	8
9	<i>venda and pneu and carro</i>	2.410	289	1.580
10	<i>venda or pneu or carro</i>	360.000	722.582	193.000
11	<i>venda pneu carro</i>	2.410	289	1.170
12	<i>“venda pneu carro”</i>	0	0	0
13	<i>loja and pneu and carro</i>	2.080	1.810	1.230
14	<i>loja or pneu or carro</i>	339.000	2.080	179.000
15	<i>loja pneu carro</i>	2.080	1.810	903
16	<i>“loja pneu carro”</i>	0	0	0
17	<i>car and tire and shop</i>	71.312	235.000	219.003
18	<i>car or tire or shop</i>	49.624.832	270.000	3.320.003
19	<i>car tire shop</i>	71.312	235.000	162.003
20	<i>“car tire shop”</i>	5	10	14

Tabela 1.1 – Resultados quantitativos referente à busca de informações nos portais

*sites* que contenham todos os termos nominativos fornecidos. A existência desses termos nominativos independe da ordem ou localização;

- Utilização do conectivo lógico “*or*” para relacionamento dos termos nominativos: orientam os mecanismos de busca da informação a selecionarem os *sites* que contenham pelo menos um dos termos nominativo fornecidos;
- Não utilização de conectivos lógicos: orientam os mecanismos de busca da informação a utilizarem os mesmos critérios associados quando da utilização do conectivo lógico “*and*”;
- Encapsulamento da expressão por aspas (“”): orientam os mecanismos de busca da informação a selecionarem os *sites* que contenham a seqüência exata dos termos nominativos, fornecidos entre as aspas.

Porém, apesar do auxílio à busca de informações na Web fornecido pelos portais por meio dos mecanismos de busca da informação, os mesmos criam três

problemas. Esses problemas estão relacionados à abrangência dos resultados, aos critérios da busca aplicados pelos portais e ao desestímulo à análise dos resultados.

O primeiro problema, relacionado à abrangência dos resultados, pode ser visualizado na tabela 1.1, comparando os diferentes resultados quantitativos obtidos de cada portal para uma mesma expressão lógica. Essa diferença nos resultados pode ser explicada com base em informações obtidas no *site* “A Promotion Guide” [34].

Esse *site* descreve como divulgar um *site* utilizando os critérios de busca dos mecanismos de busca da informação e os critérios de classificação dos *sites* adotados pelos portais. Inicialmente os portais distinguem os *sites* em duas categorias, os comerciais e os não comerciais. Para os *sites* comerciais, isto é, os *sites* que tem como propósito primário gerar renda ou promover a venda de bens ou serviços, é cobrada uma taxa de divulgação. Como exemplo de tratamento diferenciado na divulgação desses *sites* comerciais, tem-se a inclusão dos *sites* em listas de diretórios, associados a palavras-chave, e que terão prioridade de divulgação quando os termos nominativos fornecidos aos mecanismos de busca da informação coincidirem com palavras-chave do *site*. Porém, isso não inibe a presença de *sites* comerciais que não pagam a taxa de divulgação. O diferencial entre pagar ou não a taxa de divulgação, além da presença na lista de diretórios do portal, é a utilização de alguns truques ou regras na construção dos *sites*. Essas heurísticas incluem regras que estimulam a inserção e utilização adequada de marcações no texto. Essas marcações relacionam o *site* a categorias de negócios pré-definidas pelos portais, otimizando o processo de busca da informação pelo mecanismo de busca do portal.

O texto a seguir é uma marcação especificada pelo portal:

```
<meta name="keywords" content="cooper tire and rubber company, ctb,
tire, tires, commercial truck tire, commercial truck, medium truck tire,
medium truck, all position, drive axle, cooper, cooper tire, cooper tires">
```

Esse texto está contido na página localizada no endereço

```
“www.coopertires.com\tire_cooper\commercial.html”
```

do *site* da empresa “*Cooper Tire & Rubber Company*”.

A empresa é fornecedora de produtos de borracha para automóveis, abrangendo controle de vibração e sistemas de mangueira. A primeira palavra à esquerda, “meta”, é uma marcação, a palavra “*name*” é um atributo de valor “*keywords*” e identifica que o valor do atributo “*content*”, que vem a seguir, são palavras-chave. Essas palavras-chave, são utilizadas na comparação com as palavras fornecidas pelo usuário, por meio do mecanismo de busca.

O segundo problema, relacionado à semântica dos critérios de busca aplicados pelos portais, pode ser observado na tabela 1.1 comparando os resultados da linha 5 com a linha 7, em que a expressão de ambas as linhas possuem o mesmo significado para os portais Yahoo e Altavista, fornecendo assim os mesmos resultados. Porém, apesar das expressões serem as mesmas, também, para o mecanismo de busca fornecido pelo portal Netscape, os valores são diferentes. Ora, supondo que o conectivo “*and*” fosse considerado um termo nominativo, a quantidade de endereços de *sites* retornada está aquém das expectativas, tendo em vista que a palavra “*and*” é de uso comum em qualquer texto em inglês. Por outro lado, se for considerado, “*and*”, como um conectivo, os resultados diferenciados nas linhas negam essa afirmação.

Assim, conclui-se que o mecanismo de busca fornecido pelo portal Netscape, interpreta a expressão que contém o conectivo “*and*” diferentemente da expressão que não contém. Esses resultados diferentes para uma mesma expressão podem ser visualizados, também, nos seguintes pares de linhas da tabela 1.1: linha 9 com a linha 11, linha 13 com a linha 15 e linha 17 com a linha 19.

O terceiro problema, relacionado ao desestímulo à análise dos resultados, tem como origem a quantidade elevada de endereços de *sites* retornados, apesar de os mecanismos de busca tenderem a delimitar um contexto. Por exemplo, na figura 1.1 pode ser visualizado, parcialmente, um resultado da aplicação de uma expressão lógica a um mecanismo de busca da informação de um portal, que resultou em uma lista contendo 20 endereços de *sites* (de um total de 28.922), os quais deverão ser analisados em busca de solução às necessidades de informação.

Um outro exemplo pode ser observado na tabela 1.1, na coluna Altavista linha 2 (40.971.648) e na coluna Yahoo linha 18 (49.624.832), em que os resultados quantitativos são mais volumosos em relação às outras expressões lógicas.

Apenas como observação, vale alertar, que essa grande quantidade de endereços de *sites* não é proporcional a qualidade da informação, isto é, muitas informações

Web Site Matches 1 - 20 of 28922 | [Next 20 >](#)

1. [BFGoodrich Tires](#) - manufacturing and sales of automotive aftermarket replacement tires.  
<http://www.bfgoodrichtires.com>  
 More sites about: [Auto Tires > Brands](#)
  
2. [Olson Tire Total Car Care](#) - wholesale and retail sales of commercial and passenger car tires, as well as providing general repair services.  
<http://www.olsontire.com/>  
 More sites about: [Automotive Parts > Tires](#)
  
3. [Timmins Tire Sales LTD.](#) - retailer of car and truck tires, and cellular communications products.  
<http://www.timminstire.com/>  
 More sites about: [Canada > Ontario > Cochrane > Timmins > Business to Business](#)
  
4. [Discount Tire Centers](#) - tire sales in California.  
<http://www.discounttires.com/>  
 More sites about: [California > Automotive > Tires](#)
  
5. [Goss Tire Company](#) - complete automotive service and repair company specializing in retail and wholesale tire sales. Locations throughout New York and Vermont.  
<http://www.gosstire.com>  
 More sites about: [Vermont > Burlington Metro > Automotive > General Repair](#)

---

13. [Pacific Tire Sales, Inc.](#) - offering various brands of tires, custom wheels, alignments, brakes, shocks, and more.  
<http://www.pacifictiresales.bizonthe.net/>  
 More sites about: [California > Midway City > Automotive > Tires](#)
  
14. [Rolls Royce and Bentley Motor Cars](#) - everything you ever wanted to know about the cars, including technical information, every model ever made, cars for sale and wanted, etc.  
<http://www.darkforce.com/royce/>  
 More sites about: [Automotive > Rolls-Royce](#)

---

17. [Pirelli Tire Italy](#)  
<http://www.pirelli.com/>  
 More sites about: [Tires > Pirelli](#)
  
18. [Discount Tire Direct](#) - mailorder source for tires and wheels.  
<http://www.discounttiredirect.com>  
 More sites about: [Automotive Parts > Tires](#)
  
19. [Tire Rack, The](#) - high performance tire, wheel, and suspension products distributor.  
<http://www.tirerack.com/>  
 More sites about: [Automotive Parts > Tires](#)  
 Yahoo! Shopping: [Shop at Tire Rack, The](#)
  
20. [Hertz Car Sales](#) - offers a selection of vehicles nationwide.  
<http://www.hertzcarsales.com/>  
 More sites about: [Automotive Dealers > Used](#)

1-20 of 28922 | [Next 20 >](#)

Figura 1.1 – Exemplo de relação de endereços de sites, fornecida como resultado da aplicação da expressão lógica “car or tire or sales” ao mecanismo de busca do portal Yahoo

não atendem às necessidades de informação do usuário. Isso pode ser notado em uma análise superficial do resultado fornecido por um mecanismo de busca. Nesses resultados é possível detectar endereços de *sites* que, apesar de possuírem relação com a expressão lógica fornecida, não traduzem corretamente as necessidades de informação.

## 1.2 Objetivos

Uma forma de melhorar a eficiência do processo de busca por uma informação é acrescentar técnicas complementares aos mecanismos sintáticos existentes.

Assim, este trabalho tem os seguintes objetivos:

- Analisar a eficiência resultante da utilização de técnicas complementares de busca da informação, em particular mecanismos de extração de informações a partir de trechos explícitos nos documentos HTML, e
- Analisar a eficácia de um processo de armazenamento de informações extraídas da Web numa base de conhecimento.

## 1.3 Organização

Esta dissertação está organizada em 5 capítulos. No capítulo 2, é feita uma revisão bibliográfica sobre os conceitos básicos de representação do conhecimento, iniciando-se com o conceito do conhecimento, abordando ontologia, linguagens de representação do conhecimento, bases de conhecimento e estimadores probabilísticos. No capítulo 3, está descrito a proposta de solução, com detalhes do processo de desenvolvimento. O capítulo 4, apresenta os resultados experimentais com a utilização do Minerador Web. Por fim, no capítulo 5 são descritas as considerações finais.

# Capítulo 2

## REPRESENTAÇÃO DO CONHECIMENTO

Neste capítulo são apresentados conceitos sobre conhecimento e representação do conhecimento. A seção 2.1, é dedicada à apresentação da relação entre o conhecimento e sua representação. Na seção 2.2 é abordado o conceito de ontologia. A seção 2.3 contém conceitos das linguagens XML, HTML e Prolog. Na seção 2.4 é apresentado o conceito de base de conhecimento, e por fim na seção 2.5 é apresentado o conceito das equações utilizadas para melhorar os estimadores probabilísticos, a serem aplicados no classificador Minerador Web.

### 2.1 O conhecimento

A resolução de um problema exige o uso de conhecimento, tanto específico, relacionado com os elementos pertinentes ao problema, como genérico, sobre o mundo onde o problema se insere. O processo prático de captura do conhecimento necessário à resolução de um problema inicia-se por uma atividade denominada abstração. A abstração é a origem de todo o processo cognitivo, no qual o que é escolhido como objeto de reflexão é isolado de uma série de características a ele relacionado na realidade concreta, com o intuito de considerá-lo apenas em seu aspecto mensurável e quantitativo. Em resumo, a abstração consiste em desenvolver um modelo simplificado do mundo real, em que são descritos apenas os objetos e relações relevantes para a resolução do problema. Esse modelo abstrato resultante é chamado de conceituação.

A conceituação, segundo Gamper [12], é definida pela composição de duas partes:

1. **Identificação dos conceitos:** objetos, eventos, estados de negócios, crenças, etc;
2. **Relações entre conceitos.**

Já para Genesereth ([13] pág. 9) a conceituação é definida por meio de uma seqüência formada por objetos pertencentes ao domínio de conhecimento, um conjunto de funções e um conjunto de relações sobre o domínio considerado.

É apresentado, a seguir, um exemplo do processo de conceituação no domínio empresas. Inicialmente o domínio considerado é descrito em linguagem natural:

“Uma empresa possui um nome que a distingue de outras empresas, possui um ramo de atividade de negócio que caracteriza seu produto, tem um endereço do local onde ela está instalada e é possível contatá-la por meio de diversos meios de comunicação.”

Uma possível conceituação poderia incluir os seguintes objetos:

- Nome da empresa;
- Ramo de atividade do negócio;
- Produto;
- Endereço;
- Meios de contato.

A conceituação resultante também poderia incluir as seguintes relações entre os objetos que foram definidos:

- Nome da empresa **está associado** a um ramo de atividade do negócio;
- Produto **é consequência do** ramo de atividade do negócio;
- Nome da empresa **está associada** a um endereço;
- Nome da empresa **está associada** a meios de comunicação.

Porém, para a descrição formal de uma conceituação é necessário utilizar uma linguagem simbólica, que deve ser expressiva o suficiente para representar de maneira clara os elementos da abstração, ou seja, a linguagem deve prover elementos simbólicos que, ligados aos objetos e relações da abstração, descrevam corretamente a semântica da conceituação. As ligações entre os elementos da conceituação e os termos simbólicos da linguagem formam um conjunto denominado de interpretação.

<b>Nível</b>	<b>Primitiva</b>
<i>Nível do Conhecimento</i>	possui uma interpretação com relação à aquisição do conhecimento.
Nível dos símbolos	corresponde aos objetos.
Nível Lógico	onde se encontram as proposições, predicados lógicos, funções e operadores que determinam uma semântica.
Nível de Implementação	corresponde ao nível primitivo que permite a construção de estruturas de dados sem se preocupar com a semântica

Tabela 2.1 – Classificação das primitivas utilizada no formalismo da representação do conhecimento ([29] pag.96)

A descrição do conhecimento (sendo que em termos práticos o conhecimento a ser considerado é aquele definido pelo processo de abstração) e a representação simbólica estão intimamente relacionados.

Para tentar esclarecer as relações entre conhecimento e representação, Newell [29] propôs o modelo de um sistema de níveis de representação, em que um nível consiste em um meio no qual os componentes fornecem processos primitivos que permitem instanciar componentes do nível imediatamente superior. Newell exemplifica sua proposta, que pode ser visualizada na tabela 2.1, utilizando um sistema computacional. Cada nível é definido de maneira independente, mas passível de ser representado pelo nível abaixo.

O nível do conhecimento, proposto por Newell, define um sistema denominado agente, composto de objetivos, ações e corpo. Na figura 2.1 pode ser visualizada a estrutura de um agente.

O agente interage com o ambiente por meio de seu corpo por meio de processos de aquisição de informações e de inferência. O resultado desses processos é conhecido como conhecimento adquirido, que o agente processa para determinar as ações a serem executadas. As ações que farão atingir os objetivos do agente são selecionadas por meio da aplicação de regras de comportamento. Essas regras seguem o que é conhecido como princípio da racionalidade:

*“Se um agente tem conhecimento que uma de suas ações irá conduzir a um de seus objetivos, então o agente selecionará essa ação.”*

Com base na definição de um agente no nível do conhecimento e do princípio da

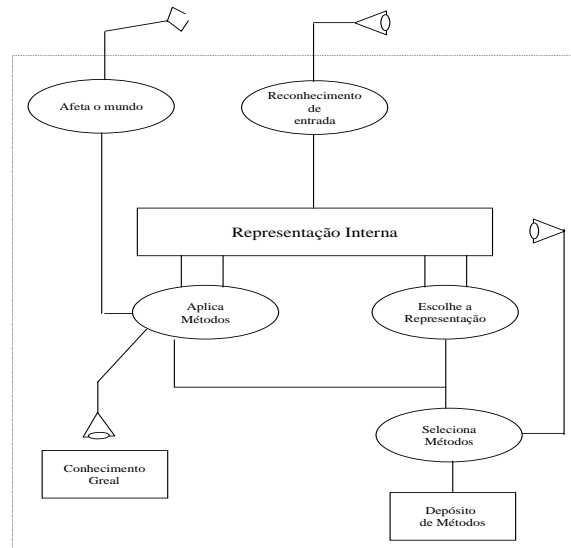


Figura 2.1 – Estrutura de um agente no nível do conhecimento.

racionalidade, conhecimento é definido como sendo:

*“Tudo que pode ser atribuído a um agente, de modo que seu comportamento possa ser computado pelo princípio da racionalidade.”*

Para representar o conhecimento em um formato que não esteja relacionado somente com a abstração de um observador individual ou a um determinado contexto, Parsaye [31] identificou cinco componentes que tornam a descrição de objetos independente do observador:

1. **Nome:** identificador único dentro de um domínio;
2. **Descrição:** cada objeto é composto por um conjunto de atributos que possuem valores, onde esses atributos descrevem as propriedades do conhecimento ou seus relacionamentos com outros objetos;
3. **Organização:** cada objeto faz parte de uma hierarquia de herança de atributos;
4. **Relações:** os valores dos atributos podem representar relações com outros objetos;

5. **Restrições:** Os atributos podem sofrer restrições de valores admissíveis.

Kelsey [19] propõe a representação do conhecimento com três componentes: a classe, o domínio e o evento. A classe refere-se a um objeto ou uma coleção de objetos. Uma vez que uma classe é definida, torna-se um recurso disponível para criar um domínio do conhecimento.

O primeiro componente proposto para a representação do conhecimento é a classe, constituída por 3 partes:

1. **Atributo:** é uma característica que auxilia na descrição de um objeto de uma classe. O atributo possui um nome e um valor ou uma lista de valores;
2. **Método:** determina o comportamento de um objeto de uma classe. Um método é composto de uma assinatura (nome unívoco do método) e um corpo contendo um conjunto de operações que definirão seu comportamento. A assinatura é composta de duas partes. A primeira parte, que é obrigatória, é um nome. A segunda parte, que é opcional, contém uma lista de argumentos. Vale ressaltar os seguintes pontos:
  - Os objetos se relacionam por meio de mensagens;
  - A descrição sobre a operação a ser realizada por um método pode ser descrita de várias formas: regras, pseudo-códigos ou programas;
  - Um método não está disponível para ser utilizado até que um objeto seja instanciado de uma classe.
3. **Perspectiva:** define como um objeto de uma classe será visto, isto é, alguns atributos serão importantes para uma determinada perspectiva e não serão relevantes em outras.

O segundo componente proposto para a representação do conhecimento é o domínio, composto pelos objetos das classes instanciadas para uso. No processo de instanciação, os objetos recebem nomes e a seus atributos são atribuídos valores. Somente o conhecimento relevante a uma determinada aplicação necessita ser definido no domínio.

O terceiro componente proposto para a representação do conhecimento é o evento. O evento refere-se à interação existente entre dois objetos, e consiste de quatro partes: um estado anterior, um método, um estado posterior e um agente. Um

estado é uma fotografia de um objeto, com os valores de seus atributos. Cada evento representa dois objetos, o objeto agente e o objeto estimulado pelo agente. Tendo identificado a forma com que o conhecimento será representado, e também, identificado os objetos e seus relacionamentos, o próximo passo consiste em descrever os objetos e seus inter-relacionamentos. Para fazer essa descrição são utilizados os conceitos de ontologia, que serão descritos na seção a seguir.

## 2.2 Ontologia

Ontologia é definida por Gruber [15] como sendo:

“Uma especificação explícita de uma conceituação.”

Quando o domínio de um conhecimento é representado em um formalismo declarativo, o conjunto de objetos a ser representado é chamado de universo de discurso. O conjunto de objetos e as relações entre eles constituem um vocabulário. A ontologia pode, então ser descrita pela definição de um conjunto de termos formados por esse vocabulário.

### 2.2.1 Propriedades da ontologia

Segundo Guarino [16], uma propriedade inerente às ontologias é a eliminação da ambigüidade. Como exemplo, tem-se a palavra “cabo”, que é um termo equívoco pois isolado constrói diversas idéias como de “cabo marítimo”, “cabo do exército”, “cabo de vassoura”, etc.

Quando delimitado por um domínio, a possibilidade de ter-se ambigüidade é pequena, mas a ambigüidade sempre existirá. Na área lingüística, a ambigüidade é denominada de polissemia, a qual é definida como a propriedade do signo lingüístico que possui vários sentidos.

A ambigüidade gera mal-entendidos na comunicação, que quando interpretada resulta em raciocínios errados. Esses problemas de ambigüidade ocorrem mesmo com a utilização de termos simples (aparentemente não ambíguos) em um domínio específico, como exemplo:

O fim de uma coisa é sua perfeição. A morte é o fim da vida. Portanto a morte é a perfeição da vida.

Observe que o termo fim é utilizado em dois sentidos diversos. Na primeira sentença o significado da palavra é um objetivo e na segunda sentença é um acontecimento.

Uma das formas de reduzir a ambigüidade é a definição formal da semântica das palavras, estabelecendo um vocabulário comum com a possibilidade de utilizá-lo com segurança dentro de vários domínios. Para isso utiliza-se ontologias, cuja função é prover o entendimento comum de um termo dentro de um domínio. A seguir uma frase com a palavra comum “vermelho”, em que se aplica a ontologia para eliminar um equívoco:

1. “Ele é um dissidente vermelho”

Aplicando ontologia em um contexto político, tem-se:

**vermelho:** pessoas com ideologia comunista.

2. “Ele ficou todo vermelho”

Aplicando ontologia em um contexto sobre a natureza humana, tem-se:

**vermelho:** enrubescer, envergonhado.

3. “Este tomate quando amadurecer ficará todo vermelho”

Aplicando ontologia em um contexto agrícola, tem-se:

**vermelho:** tonalidade de cor na qual alguns frutos alcançam quando estão maduros.

Apesar do segundo e do terceiro item referir-se à tonalidade de cor, o termo “vermelho” possui interpretações diferentes dependendo do contexto.

A descrição de ontologias por meio de uma linguagem formal facilita a definição de conceitos por meio da composição de conceitos primitivos. Os conceitos primitivos nem sempre são formalmente descritos na ontologia que está sendo criada.

Para exemplificar a noção de conceito primitivo (Vet [44]), tome-se como exemplo o conceito de um átomo de hélio. Esse conceito pode ser interpretado como um conjunto de objetos primitivos, tais como prótons, nêutrons e elétrons, partes essas que interagem para formar um átomo. Por sua vez esses objetos primitivos nem sempre precisam ser formalmente descritos.

A composição de conceitos complexos por conceitos primitivos permitem que, sem nenhuma instância conhecida, seja possível construir algo inexistente no mundo conhecido, oferecendo assim um vasto campo de pesquisa para identificar novos conceitos.

Exemplos de criação de algo inexistente podem ser encontrados em desenhos cinematográficos, onde objetos inanimados ganham membros e o dom da fala.

### 2.2.2 Parâmetros para construção de uma ontologia

A principal característica de uma ontologia é que conceitos devem ser definidos sem ambigüidade. Segundo Vet [44], a construção de uma ontologia é direcionada por dois parâmetros, sendo que o primeiro parâmetro é a escolha de objetos indivisíveis e o segundo a escolha do nível de detalhe na especificação das interações entre os objetos.

A identificação de objetos indivisíveis consiste em definir os conceitos primitivos. Por exemplo, em muitas aplicações na área biológica, a descrição do conceito no nível de uma célula é uma escolha adequada. Em outras aplicações, é desejável a definição dos componentes da célula. Na Física, para se definir o átomo de Hélio-4, poderia ser utilizado o conceito dos *quarks* como objetos indivisíveis, porém no domínio da Química o átomo de Hélio-4 seria o objeto indivisível.

O nível de detalhe na especificação das interações entre os objetos, deve conter detalhes que não dêem margem à ambigüidade, mas que sejam limitados à descrição específica das interações entre os objetos no contexto descrito. Como exemplo, ao especificar uma boa ou má troca de um pneu furado em um carro, pode ser necessário especificar além da seqüência dos parafusos a serem atarraxados, o torque a ser aplicado à chave no momento do atarraxamento. Entretanto, tal detalhamento nem sempre é interessante. Se o detalhamento for específico a uma determinada aplicação, o resultado poderá ser uma ontologia que não poderá ser compartilhada ou nem reutilizada.

## 2.3 Linguagens

Linguagens são descritas sob dois aspectos: a sintaxe e a semântica. A sintaxe representa as regras de formação das palavras e sentenças a partir dos elementos simbólicos primordiais. A semântica representa o significado dos termos simbólicos da linguagem (palavras e sentenças).

### 2.3.1 Linguagens de marcação

A marcação (*markup*) é o elemento básico das linguagens de marcação. A semântica das linguagens de marcação deve definir:

- qual marcação é permitida inserir em um documento;
- qual marcação é requerida;
- como uma marcação é identificada em um texto;
- o que a marcação significa.

O princípio básico de uma marcação é fazer uma distinção do conteúdo de um documento da sua visualização. Essa distinção permite que o autor concentre-se no conteúdo do documento e não em qual meio deverá ser publicado, se na Web, em uma carta, em uma parte de um jornal, em Braille, em áudio etc. Donovan [9], explicando sobre a origem do termo marcação, afirma que a marcação originou-se nos dias em que um editor marcava sobre um manuscrito as instruções para o tipógrafo. Essas instruções continham informações que orientavam o tipógrafo sobre a formatação do documento (parágrafo, cabeçalho, etc).

Caso as marcações sejam descritas utilizando-se o mesmo conjunto de caracteres utilizados pelo texto, é preciso utilizar um mecanismo que permita distingui-las das palavras do texto. Para diferenciá-las, as marcações são delimitadas por símbolos que identificam seu início e término, como, por exemplo, os símbolos “<” e “>”. Para exemplificar sua utilização vamos supor que a letra “b” identifique a marcação negrito e a palavra “em” identifique a marcação itálico. Aplicando-as à sentença

“Relação de <em> pontos de venda </em> do produto <b>pneu</b>.”

as palavras “pontos de venda”, deverão ser apresentadas em formato itálico e a palavra “pneu” realçada com negrito, conforme pode ser visualizado a seguir:

“Relação de *pontos de venda* do produto **pneu**.”

As marcações podem ser divididas nos seguintes tipos, conforme cita Maler [22]:

- Marcação estruturada, declarativa ou descritiva: divide o documento em partes, tais como capítulo, seção, subseção, parágrafos e tópicos, formando a estrutura do documento, conforme pode ser visualizado na figura 2.2.

Linha	Código HTML
1	... the services:
2	<ul>
3	<li> Oil change
4	<li> Tire repair
5	</ul>
6	are executed ...

Figura 2.2 – Aplicação de marcação estruturada.

As linhas 1 e 6 foram acrescentadas para elucidar o exemplo, indicando que as informações estão em uma formação diferente antes da aplicação da marcação, e que ao final retornaram a essa mesma formação. Na linha 2 está sendo declarado o início de uma lista de itens. Na linha 3 está sendo iniciado um item de uma lista e as palavras “Oil change” devem estar contidas nesse item. O mesmo ocorre na linha 4, com as palavras “Tire repair”. Na linha 5 está sendo finalizada uma lista de itens anteriormente aberta.

O texto a ser visualizado num navegador deverá ser algo do tipo:

```
... the service
    • Oil change
    • Tire repair
are executed ...
```

- Marcação procedural ou semântica: identifica conjuntos de dados, criando facilidades para que outras aplicações possam compilar, recuperar ou manipular os documentos por meio de seus atributos, em especial programas

de apresentação como, por exemplo, um navegador. Utilizando o exemplo da figura 2.2 tem-se as seguintes interpretações:

Na linha 2 o navegador recebe uma instrução de que a partir da próxima instrução, deverá inserir um *bullet* (●). As linhas 3 e 4 instruem o navegador a distanciar o texto da margem esquerda em 24 pontos (ponto neste contexto é uma medida) e qualquer “quebra” dessa linha deverá ser indentada na mesma posição vertical. A linha 5 instrui o navegador à restaurar as propriedades originais da linha.

- Marcação estruturada e procedural: possui os dois significados, como marcação estruturada organiza o documento de forma estruturada e como marcação procedural contém informação para o analisador executar ações. Como pode ser visualizada na figura 2.2 as marcações estruturadas “<ul>” e “<li>”, organizam o texto em formato de lista, sendo que, essas marcações também possuem a característica de uma marcação procedural pelo motivo de fornecerem informações de como o navegador deverá proceder.

### 2.3.2 Linguagem XML

A *eXtensible Markup Language* (XML) é uma meta-linguagem que permite definir linguagens de marcação, que por sua vez são denominadas de aplicações XML. A linguagem XML, que é uma versão simplificada da *Standard Generalized Markup Language* (SGML) [17]), foi projetada para melhorar a funcionalidade da Web, provendo identificação da informação sem a complexidade da SGML [43].

Assim, no decorrer do desenvolvimento da linguagem XML, além de serem aplicadas restrições que visavam diminuir a complexidade da SGML procurou-se manter a compatibilidade entre essas meta-linguagens. Nesse processo, a XML incorporou diversas características, como as listadas a seguir [6]:

- Um documento *eXtensible Markup Language* (XML) possui uma estrutura física e lógica, em que a estrutura física é descrita também pelos mesmos construtores da *Standard Generalized Markup Language* (SGML);
- Os nomes associados aos construtores iniciam com uma letra ou um dos caracteres: ‘.’ (ponto), ‘-’ (sinal unário de negação) e ‘\_’ (subscrito), e continuam com qualquer caracter Unicode;

- A exceção se faz para os nomes associados aos construtores que iniciam com a seqüência de caracteres “X”, “M” e “L”, em qualquer combinação de minúscula ou maiúscula. Palavras que iniciam com essa seqüência são consideradas palavras reservadas;
- As marcações finais não podem ser omitidas;
- Elementos como “br” e “img”, que não possuem conteúdo, são terminadas com uma barra (/) antes do sinal maior que (>);
- As mesmas palavras escritas com caracteres minúsculos possuem significados diferentes quando escritas com caracteres maiúsculos, o mesmo ocorrendo quando houver uma mescla entre caracteres minúsculos e maiúsculos;
- Todos os valores dos atributos devem estar entre aspas (”).

A possibilidade de criação de novas marcações (sendo essa uma das características da XML) pode gerar ambigüidade entre os nomes de marcações. Esse problema pode ser solucionado por meio da criação de um espaço de nomes (*namespace*) de forma que marcações com o mesmo nome, mas com semânticas distintas, podem existir simultaneamente, desde que existam em espaços de nomes diferentes.

A gramática de uma aplicação XML pode ser descrita formalmente por meio de uma *Document Type Definition* (DTD). A utilização de uma DTD é opcional, porém, a ausência de referência a uma DTD dificulta a portabilidade do documento gerado pela aplicação XML.

Na figura 2.3 pode ser visualizado um exemplo de uma DTD contendo a estrutura de um documento que poderá ser utilizado para descrever um contato de empresa.

Em relação a esse exemplo, figura 2.3, alguns comentários se fazem relevantes:

- A primeira linha iniciada por “<!--” e finalizada por “-->” encapsula um comentário, isto é, o programa analisador da DTD não considerará esta linha como informação a ser tratada;
- Todos os objetos obtidos no processo de abstração, quando utilizados em uma DTD, são precedidos pelos símbolos “<!” e pela palavra-chave (construtor) “ELEMENT”;

```

<!-- DTD para contato de empresa: Contato.dtd -->
<!ELEMENT empresa (nomeEmpresa & contato)>
<!ELEMENT contato (endereco & comunicacao)>
<!ELEMENT endereco (logradouro & cidade & estado? & cep?)>
<!ELEMENT comunicacao (telefone? & fax? & email?)>
<!ELEMENT nomeEmpresa (#PCDATA)>
<!ELEMENT logradouro (#PCDATA)>
<!ELEMENT cidade (#PCDATA)>
<!ELEMENT estado (#PCDATA)>
<!ELEMENT cep (#PCDATA)>
<!ELEMENT telefone (#PCDATA)>
<!ELEMENT fax (#PCDATA)>
<!ELEMENT email (#PCDATA) >

```

Figura 2.3 – Exemplo de uma DTD para contatos de empresas

- Após o nome dos objetos, existe uma relação, entre parênteses, de outros objetos que o compõem. Esses objetos de composição podem ser outros objetos que deverão ser declarados na DTD, ou palavras-chave da linguagem de marcação, como por exemplo “PCDATA” ou “EMPTY”;
- O conectivo ‘&’, indica ao analisador que os objetos podem ser utilizados em ordens diversas;
- Existem três símbolos que podem ser colocados logo após os objetos da composição ou logo após o fechamento do parênteses da composição, que indicam a quantidade de instâncias que o analisador deve encontrar. Esses símbolos possuem os seguintes significados:

’?’ , indica que se o objeto for utilizado deve ser único;

’\*’ , indica que se o objeto for utilizado pode possuir diversas instâncias;

’+’ , indica que o objeto deve ser utilizado no mínimo uma vez;

Vale salientar que a ausência de sinais nos objetos de composição indica ao analisador que o objeto deverá ser utilizado e uma única vez;

- Uma das palavras-chave utilizada, “#PCDATA”, indica ao analisador para aceitar qualquer conjunto de caracteres, mas que não faça referência aos objetos declarados.

Na figura 2.4, pode ser visualizada uma instância de aplicação XML com base na *Document Type Definition* (DTD) da figura 2.3, para contato de empresa.

```

<?xml version="1.0" ?>
<!DOCTYPE empresa SYSTEM "Contato.dtd" >
<empresa>
<nomeEmpresa>burdge, inc.</nomeEmpresa>
<contato>
<endereco>
<estado>CA</estado>
<cep>90040-1900</cep>
<cidade>Los Angeles</cidade>
<logradouro>2151 Yates Avenue</logradouro>
</endereco>
<comunicacao>
<telefone>323.722.2011 or 800.962.2486</telefone>
<fax>323.724.7901</fax>
<email>Info@Burdge.Com</email>
</comunicacao>
</contato>
</empresa>

```

Figura 2.4 – Exemplo de uma instância de aplicação XML, com base na DTD para contatos de empresas

Com relação ao exemplo da figura 2.4, a primeira linha representa o cabeçalho da aplicação e a segunda linha identifica qual DTD será referenciada para reger a seqüência, a lógica e as regras das marcações. Note que as palavras que compõem as marcações (a palavra encapsulada entre os símbolos ‘<’ e ‘>’, com início na terceira linha) possuem correspondência direta com os elementos (<!ELEMENT) declarados na DTD. As palavras entre as marcações, de início (< ... >) e fim (< /... >), correspondem às informações.

### 2.3.3 Linguagem HTML

A maioria dos documentos existentes na Web são descritos em HTML. A linguagem HTML permite a criação de documentos que serão interpretados por navegadores (*browsers*), que fazem o processamento do texto, *hyperlinks* e multimídia.

A criação de instâncias HTML deve seguir as regras contidas em uma DTD HTML, porém não há necessidade de fazer uma referência explícita à essa DTD HTML [26].

A HTML possibilita a inclusão de *scripts* escritos em “javascript”, “applet”, “vbs-

cript”, entre outros. Esses *scripts* possuem diversos propósitos como: identificação de *links* para outras páginas, informações relevantes sobre o conteúdo das páginas as quais devem ser compartilhadas com o usuário mas que não possuem um significado intrínseco sobre o conteúdo do *site* etc.

A linguagem HTML fornece um conjunto de marcações de “meta-dados” contidas na seção “< head >”, formada por “< script >”, “< style >”, “< link >”, “< title >”, “< meta >”, “< isindex >” e “< base >”; e que podem ser interpretados por diferentes aplicações. Apesar da marcação “<title>” ser sugestivo para a colocação do título da página, não existe regras que impõem ao usuário o que deverá ser escrito nas marcações.

Na figura 2.5 pode ser visualizado um exemplo de parte do código (apontado pelas setas) de uma página HTML, com sua respectiva aparência dada pelo navegador.

As informações contidas nos documentos de marcação, especificamente representadas por meio das linguagens HTML ou aplicação XML, podem ser convertidas em uma linguagem lógica para serem inseridas em uma base de conhecimento. Dentro desse contexto é feita uma introdução à linguagem Prolog na próxima seção.

### 2.3.4 Linguagem lógica

Linguagens lógicas permitem descrever objetos e seus relacionamentos, utilizando Lógica de Predicados ou Cálculo de Predicados. O Cálculo de Predicados permite representar, por meio de símbolos, objetos, propriedades e relações dos objetos de um domínio. A seguir pode ser visualizado dois exemplos de representação de objetos.

1. livro(matemática,renato).  
livro é uma relação entre os objetos “matemática” e renato.
2. madura(pera)  
madura é uma propriedade de pera.

Um termo pode ser:

- Constante (no sentido de representar um único objeto), por exemplo o fato que uma onça é um animal pode ser representado por “animal(onça)”;

The figure illustrates the mapping between the visual layout of a website and its HTML source code. It is divided into three main sections:

- Header Section:** Shows the company name and address: "Pacific Tire Sales, Inc.", "8249 Bolsa Ave", and "Midway City, Ca. 92655". Below this is a navigation bar with buttons for "Contact Info", "Directions", "Tire Specials", "Specials", and "Service Specials". At the bottom of the header are links for "E-mail", "E-mail Reminder", and "feedback". A red oval highlights the company name and address, with an arrow pointing to the corresponding HTML code snippet.
- Navigation Menu Section:** A vertical list of menu items: "TIRES", "BRAKES", "ALIGNMENTS", "SHOCKS / STRUTS", "OIL CHANGE SERVICE", and "CUSTOM WHEELS". The "BRAKES" and "OIL CHANGE SERVICE" items are highlighted in yellow. Arrows point from these items to their respective HTML code snippets.
- HTML Code Snippets:**
  - The first snippet shows the header content:

```
<TD align=middle background="pacifictiresales-- Home_ervivos/space.gif" vAlign=center width="100%">
<FONT color=black face=Tahoma size=5>Pacific Tire Sales, Inc.</FONT>
<BR>
<FONT color=black face=Tahoma size=3>
  <B>8249 Bolsa Ave </B>
</FONT>
<BR>
<FONT color=black face=Tahoma size=3>
  <B>Midway City, Ca. 92655</B>
</FONT>
</TD>
```
  - The second snippet shows the "BRAKES" menu item:

```
<TABLE border=0 cellPadding=0 width="100%">
<TBODY>
<TR>
<TD align=left vAlign=top>
  <FONT face=Verdana,Arial,Helvetica,sans-serif size=1>
    <P>BRAKES</P>
  </FONT>
</TD>
</TR>
</TBODY>
</TABLE>
```
  - The third snippet shows the "OIL CHANGE SERVICE" menu item:

```
<TABLE border=0 cellPadding=0 width="100%">
<TBODY>
<TR>
<TD align=left vAlign=top>
  <FONT face=Verdana,Arial,Helvetica,sans-serif size=1>
    <P>OIL CHANGE SERVICE</P>
  </FONT>
</TD>
</TR>
</TBODY>
</TABLE>
```

Figura 2.5 – Código fonte parcial de uma página HTML

Denominação	Símbolo
negação	$\neg$
conjunção	$\wedge$
disjunção	$\vee$
implicação	$\leftarrow$ ou $\rightarrow$
bi-implicação	$\leftrightarrow$

Tabela 2.2 – Conectivos lógicos do Cálculo de Predicados

- Variável (no sentido de representar diferentes objetos em diferentes instâncias) como exemplo, qualquer coisa é um animal, representado por  $\text{animal}(X)$ ;
- Função (mapeamento de um objeto em outro) como por exemplo em “ $\text{animal}(\text{mamífero}(\text{gato}))$ ”, em que “gato” é um objeto e está sendo uma constante para “mamífero”. O novo objeto “ $\text{mamífero}(\text{gato})$ ” está sendo uma constante para “animal”;

Um predicado é formado por um símbolo que representa uma relação, acrescido dos termos que compõem a relação, como exemplo:

$\text{homem}(\text{socrates})$

em que:

**predicado** : homem;

**termo** : socrates.

O termo “socrates” é uma constante, pois representa um objeto.

A Lógica de Predicados permite o uso de diversos conectivos, cuja a relação pode ser visualizada na tabela 2.2.

Um exemplo de utilização dos conectivos pode ser:

$$\forall X(c(X) \rightarrow o(X))$$

que esta representando a sentença: “Para todo  $X$ , se  $X$  é  $c$ , então  $X$  é  $o$ ”.

Predicados podem conter variáveis quantificadas por meio de quantificadores universais ou existenciais. Na tabela 2.3, podem ser visualizados esses símbolos.

Um exemplo de utilização dos quantificadores, pode ser:

Denominação	Símbolo
universal	$\forall$
existencial	$\exists$

Tabela 2.3 – Quantificadores do cálculo de predicados

- 1:  $\forall X(mamifero(X) \rightarrow animal(X))$
- 2:  $\exists X(ave(X) \rightarrow voa(X) \wedge animal(X))$

que estão representando as seguintes sentenças:

- 1: Para todo  $X$ , se  $X$  é mamífero isso implica que  $X$  é um animal;
- 2: Existe  $X$ , se  $X$  tem a propriedade de ser uma ave, implica que  $X$  voa e também é um animal.

Segundo Emden e Kowalski [10], uma disjunção ( $\vee$ ) de fórmulas atômicas, ou literais ( $l_n$ ), negativos ou não, do tipo:

$$l_1 \vee l_2 \vee l_3 \vee \dots \vee l_n$$

é uma cláusula e quando essa disjunção de literais possuir no máximo um literal positivo é denominada de “Cláusula de Horn”.

Uma notação alternativa de uma cláusula é apresentada a seguir:

$$A_1 \wedge A_2 \wedge \dots \wedge A_m \rightarrow B_1 \vee B_2 \vee \dots \vee B_n$$

O lado esquerdo é denominado de “antecedente” e o lado direito “conseqüente”.

A resolução foi idealizada para ser aplicada a sentenças na forma clausal. Um exemplo de resolução pode ser visualizada a seguir([38] pag. 576):

$$\begin{aligned} \text{mais\_velho(joanne, jake)} &\rightarrow \text{mãe(joanne, jake)} \\ \text{mais\_sábio(joanne, jake)} &\rightarrow \text{mais\_velho(joanne, jake)} \end{aligned}$$

o resultado da aplicação da regra da resolução é:

$$\text{mais\_sábio(joanne, jake)} \rightarrow \text{mãe(joanne, jake)}$$

```

1  % Formato dos termos
2  % site( nome da empresa, endereço do site).
3  % ramo( nome da empresa, nome do ramo).
4  % atividade( nome da empresa, atividade).
5  % endereco( nome da empresa, estado, zona postal, cidade, complemen-
   to, via).
6  % meiosCOM( nome da empresa, telefone voz, telefone fax, e-mail).
7  % produto( nome da empresa, produto, especialização, características,
   preço).
8  %
9  % Regras
10 estado(E,U) :- endereco(E, U, -, -, -, -).
11 zpostal(E,Z) :- endereco(E, -, Z, -, -, -).
12 cidade(E,C) :- endereco(E, -, -, C, -, -).
13 via(E,V,C) :- endereco(E, -, -, -, C, V).
14 ...
15 correioEletronico(E,Email) :- meiosCOM(E, -, -, Email).
16 %
17 % Fatos
18 site( 'pacific tire sales, inc', 'pacific.cpct' ).
19 ramo( 'pacific tire sales, inc', 'Automóvel' ).
20 atividade( 'pacific tire sales, inc', 'Fabricante' ).
21 endereco( 'pacific tire sales, inc', 'Ca', '92655', 'Midway City', -, '8249
   Bolsa Ave').
22 meiosCOM( 'pacific tire sales, inc', '714.892.2093', -, -).
23 produto( 'pacific tire sales, inc', tire, '155R12', 'PASSENGER / PER-
   FORMANCE TIRES METRIC P-METRIC 155R', 18.00).
24 ...

```

Figura 2.6 – Documento XML representado em Prolog

A linguagem Prolog também pode ser utilizada para representar aplicações XML. Na figura 2.6, pode ser visualizado um exemplo de um documento XML escrito em Prolog.

Com relação à figura 2.6, as linhas correspondem à:

- Linhas 1 a 8: são comentários sobre a estrutura da informação, e são opcionais. O símbolo ‘%’, identifica que os símbolos à direita são comentários;
- Linhas 10 a 15: são regras disponíveis ao usuário para busca das informações relacionadas;
- Linhas 18 a 24, estão representados os fatos.

## 2.4 Base de conhecimento

Russel [36] apresenta o conceito de base de conhecimento como sendo um conjunto de representações de fatos do mundo.

O’Leary [20] recomenda que a base de conhecimento deve ser construída de tal forma que se tenha qualidade no conhecimento adquirido, isto é, que não haja informações conflitantes ou desnecessárias.

Um dos processos utilizados para aumentar a qualidade do conhecimento adquirido é delimitar o domínio. Porém, mesmo delimitando um domínio existem dificuldades em se desenvolver uma base de conhecimento. Algumas dessas dificuldades foram observadas por O’Leary e estão descritas a seguir:

- As informações podem estar armazenadas em mídia que não possibilitam a aquisição imediata por mecanismos computacionais, como por exemplo documentos em formato de papel;
- Muitas bases de conhecimento utilizam uma única fonte de informação e normalmente são limitadas a um único tipo de informação;
- A base de conhecimento normalmente é composta de informações provenientes de fontes diversas que podem estar estruturadas ou não. As informações provenientes de fontes não estruturadas, não permitem a aplicação de um mesmo conjunto de métodos de recuperação da informação em diversos documentos;

- Algumas bases de conhecimento utilizam todas as informações disponíveis em um documento fonte, em outras a informação deve ser abstraída, sintetizada e/ou complementada com outras fontes de informação.

A complexidade de desenvolvimento de uma base de conhecimento pode ser caracterizada pelos seguintes aspectos:

- Grau de manutenibilidade: se a informação sofre constantes alterações;
- Cardinalidade: se a informação possui uma formação simples, com um único valor (univalorada) como por exemplo, “nome do pai”, ou se a informação é multivalorada, como exemplo, “nome dos filhos”;
- Existência explícita: a menção da própria informação por si só é suficiente, como por exemplo: “Rua Cardeal Arco Verde, 7684”. Essa informação refere-se explicitamente a um endereço;
- Existência implícita: a informação não é expressa formalmente, por exemplo, “A redação foi devolvida”, em que está implícito que a redação foi entregue, ou que a redação sofreu algum tipo de avaliação.

Sob esses aspectos, pode-se concluir que a identificação da informação exige o desenvolvimento de estratégias específicas para capturá-las e armazená-las.

Deve-se notar que, uma base de conhecimento pode conter um conhecimento pré-existente. A esse conhecimento pré-existente dá-se o nome de “Conhecimento de Fundo” (*Background Knowledge*). O conhecimento de fundo é formado de fatos relacionados ao domínio representado na base de conhecimento. Como exemplo de uma base de conhecimento com conhecimento de fundo, seria uma relação de fatos sobre a malha rodoviária, ferroviária e aeroviárias de uma região, contendo distância e tempo de trajeto entre as cidades utilizando os vários meios de locomoção. Essas informações de distância e tempo são inseridas diretamente na base de conhecimento, formando o conhecimento de fundo.

Uma aplicação para esse conhecimento de fundo seria a delimitação de fornecedores dentro de um raio de ação (distância), ou a delimitação de fornecedores por prazo de entrega (tempo de trajeto entre as cidades). A seguir é apresentado um exemplo de consulta que pode ser feita à essa base de conhecimento, a qual necessita, para ser respondida, fatos relativos a distância entre as cidades:

“Relacione as empresas que possam fornecer pneus pelo preço máximo de R\$70,00, com frete incluso, tomando como referência de entrega a cidade de São Carlos, constando na relação o nome da empresa, valor e a cidade”.

O resultado esperado é uma relação dos itens solicitados, com a abrangência a todas as cidades que cobram, incluso o frete, o limite estipulado, considerando como referência a cidade de São Carlos.

Os conceitos descritos neste capítulo serão aplicados no desenvolvimento do Minerador Web e que será iniciado no capítulo 3.

## 2.5 Estimadores probabilísticos

O Minerador Web necessita da identificação de informações implícitas em um documento. Essa identificação é feita por meio de um classificador de *sites*, que tem como base o método Naive Bayes, descrito na subseção 2.5.1. O método Naive Bayes não considera a relação entre palavras em um texto, além de atribuir valores nulos para as palavras que não ocorreram no conjunto de treinamento. Isso prejudica os resultados de classificação, pelo motivo de apresentar valores extremos e às vezes indistintos entre duas classes. Para minimizar esse problema é aplicado a técnica de suavização de Witten-Bell, descrita na subseção 2.5.3. Porém a aplicação de um volume grande de palavras no classificador não interfere nos resultados, apenas torna o processo de classificação mais demorado. Visando agilizar o processo e identificar as possíveis palavras que contribuem para a identificação de uma classe é aplicada uma medida da informação denominada de Informação Mútua Média (*Average Mutual Information (AMI)*), que está descrita na subseção 2.5.2.

### 2.5.1 Método Naive Bayes

Existem informações que não são possíveis de serem visualizadas em um texto. Essas informações são denominadas de informações implícitas. A abordagem utilizada pelo Minerador Web é determinar essas informações por meio de classificação do *site* sobre classes de uma categoria. Considerando as páginas de um *site* como sendo um único documento HTML, o problema é equivalente ao de

classificação de textos.

Formalmente, a tarefa de classificação de textos consiste em atribuir um valor booleano para cada par  $\langle d_j, c_i \rangle \in \mathcal{D} \times \mathcal{C}$ , onde  $\mathcal{D}$  é o conjunto dos documentos e  $\mathcal{C}$  é o conjunto de classes de documentos, por meio de uma função

$$\Phi : \mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$$

Assim, se o documento  $d_j$  pertencer à classe  $c_i$  a função  $\Phi$  irá atribuir o valor  $T$  ao documento  $d_j$ , caso contrário irá atribuir o valor  $F$  ([37]).

Na década de 80 o que mais se aproximava ao processo de classificação automática de textos eram técnicas que consistiam em definir manualmente um conjunto de regras para uma área específica do conhecimento, que serviam de base para o processo de classificação de documentos. Posteriormente, principalmente durante a década de 90, iniciou-se a utilização de técnicas de aprendizagem de máquina para a construção de classificadores de texto. O objetivo dessas técnicas é encontrar uma função  $\Psi$ , denominada de classificador, que seja uma aproximação da função  $\Phi$  desconhecida e que permita realizar a tarefa de classificação com um índice de erro menor que um certo patamar estabelecido.

O Minerador Web utiliza uma abordagem probabilística para a classificação dos *sites* em termos de ramos de atividades. O método utilizado é uma variante do classificador *Naive Bayes*.

A abordagem bayesiana para classificação de textos é escolher a classe mais provável ( $c^*$ ) dado os atributos  $a_1, a_2, \dots, a_n$  que descrevem cada documento. Como pode ser visualizado na figura 2.7, um *site* é composto por um conjunto de páginas HTML, que formam um documento  $d$ . O documento pode ser caracterizado por atributos que correspondem as palavras  $\{w_1, w_2, \dots, w_n\}$  do seu conteúdo útil (texto). Cabe ao Minerador Web classificar o documento em uma das classes de uma categoria

Assim, um método bayesiano classifica um documento utilizando a seguinte equação:

$$\begin{aligned} c^* &= \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} [p(c_i | a_1, a_2, \dots, a_n)] = \\ &= \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} [p(c_i | w_1, w_2, \dots, w_n)] \end{aligned} \quad (2.1)$$

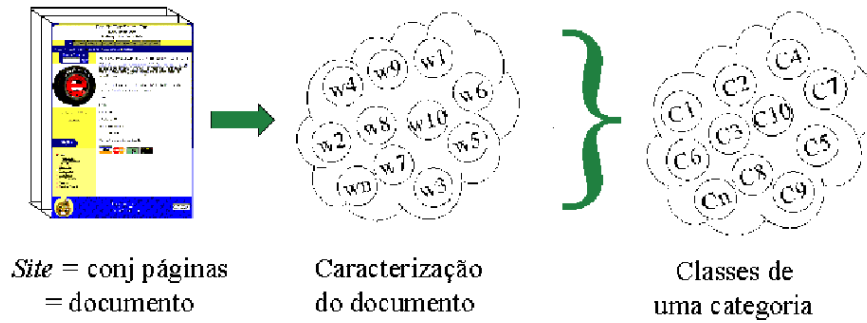


Figura 2.7 – O problema da classificação de *sites*.

Usando o teorema de Bayes, a equação 2.1 toma a seguinte forma:

$$c^* = \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} \frac{p(w_1, w_2, \dots, w_n | c_i) p(c_i)}{p(w_1, w_2, \dots, w_n)} \quad (2.2)$$

Eliminando o denominador, uma vez que é um termo constante em relação às classes de documentos, tem-se que:

$$c^* = \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} p(w_1, w_2, \dots, w_n | c_i) p(c_i) \quad (2.3)$$

A questão que surge agora é como determinar as duas probabilidades da equação 2.3. O valor do termo  $p(c_i)$  pode ser facilmente estimado por meio da frequência com que cada classe  $c_i$  ocorre no conjunto de treinamento. Entretanto, calcular o valor do termo  $p(w_1, w_2, \dots, w_n | c_i)$  depende da disponibilidade de um grande conjunto de treinamento para se ter estimativas confiáveis, uma vez que o tamanho do espaço amostral é igual ao número de possíveis seqüências de palavras vezes o número de classes consideradas. O classificador *Naive Bayes* assume, como estratégia simplificadora, que as ocorrências das palavras são eventos independentes, dado uma classe de documentos ([24]). Dessa forma, a probabilidade de ocorrência da seqüência pode ser calculada pelo produto das probabilidades de ocorrência das palavras, consideradas individualmente:

$$c^* = \underset{c_i \in \mathcal{C}}{\operatorname{argmax}} p(c_i) \prod_{j=1}^n p(w_j | c_i) \quad (2.4)$$

A equação 2.4 é utilizada no classificador Naive Bayes.

### 2.5.2 Informação Mútua Média

Considerando um cenário simplificado em que as classes de documentos são equiprováveis, a equação 2.4 mostra que a classe escolhida será aquela que apresentar maiores valores de probabilidade para as palavras que representam um documento. Isso significa que palavras que aparecem com a mesma frequência em todas as classes de documentos não contribuem para o processo de classificação, uma vez que os seus valores de probabilidade serão os mesmos no cálculo do produto da equação 2.4. Sendo assim, a remoção dessas palavras simplifica os cálculos a serem realizados pelo classificador *Naive Bayes*.

Analisando as palavras e suas associações com as classes, o conjunto de palavras que são candidatas a serem removidas inclui:

- remoção de palavras neutras como artigos, preposições, conjunções, etc;
- remoção de palavras que compartilham as mesmas raízes (monemas);

Outra hipótese simplificadora é que as palavras não devem ser consideradas sensíveis ao tamanho da letra, isto é, os termos são analisados na sua forma em letras minúsculas (*lower case*).

Em termos quantitativos, uma técnica utilizada para a redução do vocabulário a ser usado é ponderar as palavras de acordo o ganho de informação. Esse ganho de informação é dado pela informação mútua média (*Average Mutual Information - AMI*, [8]). Seja  $W_i$  uma variável aleatória que indica se a palavra  $w_i$  está presente ou ausente num documento e  $\mathcal{C}$  o conjunto de classes de documentos. A informação mútua média é calculada da seguinte maneira:

$$\begin{aligned}
I(C; W_i) &= H(C) - H(C | W_i) = \\
&= - \sum_{c \in C} p(c) \log_2(p(c)) \\
&\quad - - \sum_{v_i \in \{w_i, \neg w_i\}} p(v_i) \sum_{c \in C} p(c | v_i) \log_2(p(c | v_i)) = \\
&= \sum_{v_i \in \{w_i, \neg w_i\}} \sum_{c \in C} p(c, v_i) \log_2 \left( \frac{p(c, v_i)}{p(c) p(v_i)} \right) \tag{2.5}
\end{aligned}$$

em que  $H(C)$  é a entropia de  $C$ .

Na implementação do termo  $\frac{p(c, v_i)}{p(c) p(v_i)}$ , da equação 2.1, foi aplicado o seguinte procedimento:

- $p(c)$  é a divisão da quantidade de palavras que ocorreram nos documento rotulados como classe  $c$  ( $|c|$ ), pela quantidade total de palavras ( $|C|$ );

$$p(c) = \frac{|c|}{|C|}$$

- $p(v_i)$  é a divisão da quantidade de ocorrências da palavra  $w_i$  ( $N(w_i, c_j)$ ), pela quantidade total de ocorrências das palavras ( $N(w_k, C_y)$ );

$$p(v_i) = \frac{\sum_{j=1}^C N(w_i, c_j)}{\sum_{\substack{k=1 \\ y=1}} N(w_k, c_y)}$$

- $p(c, f_i)$  é a divisão da quantidade de ocorrência da palavra  $w_i$  nos documentos rotulados como classe  $c$  ( $N(w_i, c)$ ), pela quantidade de ocorrência das palavras ( $N(w_k, c_y)$ );

$$p(c, v_i) = \frac{N(w_i, c)}{\sum_{\substack{k=1 \\ y=1}}^{T,C} N(w_k, c_y)}$$

### 2.5.3 Método Naive Bayes com suavização

Craven et al ([8]) mostraram que é possível melhorar a performance do classificador *Naive Bayes* utilizando estimadores mais complexos para as probabilidades de ocorrência das palavras. Um dos problemas notados é que a premissa de independência entre as palavras cria distorções significativas nos resultados fornecidos pelo classificador *Naive Bayes*. Para contornar esse problema, foi desenvolvida uma variante do classificador *Naive Bayes*, conforme é mostrado no desenvolvimento a seguir.

Transformando a equação 2.4 da forma de produto de probabilidade de ocorrência de palavras para uma forma de média geométrica das probabilidades, tem-se a equação 2.6.

$$\sqrt[n]{p(c) \prod_{j=1}^n p(w_j | c)} \quad (2.6)$$

A Média Geométrica de um conjunto de ‘ $n$ ’ valores é a raiz  $n$ -ésima do produto desses ‘ $n$ ’ valores. Quando o número de eventos é muito grande, é aconselhável o emprego de logaritmos, neste caso na base 2, resultando na equação 2.7

$$\begin{aligned} & \frac{1}{n} \cdot \log_2 \left( p(c) \prod_{j=1}^n p(w_j | c) \right) = \\ & = \frac{\log_2(p(c))}{n} + \frac{\sum_{j=1}^n \log_2(p(w_j | c))}{n} \end{aligned} \quad (2.7)$$

Aplicando o conceito de perplexity, conceito este utilizado na área de reconheci-

mento de voz, com o objetivo de normalizar os resultados na largura dos dados, tem-se a equação 2.8

$$\propto \frac{\log_2(p(c))}{n} + \frac{\sum_{i=1}^T \log_2(p(w_i | c))^{N(w_i, d)}}{n} \quad (2.8)$$

onde  $T$  representa a quantidade de palavras únicas do vocabulário e  $N(w_i, d)$  representa a quantidade de palavras  $w_i$  no documento  $d$ .

Aplicando a propriedade de logaritmo de potência na equação 2.8, tem-se a equação 2.9.

$$\begin{aligned} &= \frac{\log_2(p(c))}{n} + \frac{\sum_{i=1}^T N(w_i, d) \log_2(p(w_i | c))}{n} = \\ &= \frac{\log_2(p(c))}{n} + \sum_{i=1}^T \frac{N(w_i, d)}{n} \log_2(p(w_i | c)) \end{aligned} \quad (2.9)$$

Se o termo  $\frac{N(w_i, d)}{n}$ , da equação 2.9, for interpretado como  $p(w_i | d)$ , então a expressão resulta em:

$$\propto \frac{\log_2(p(c))}{n} + \sum_{i=1}^T p(w_i | d) \log_2(p(w_i | c)) \quad (2.10)$$

O segundo termo da equação 2.10 pode ser interpretado como a “cross-entropia” negativa entre a distribuição de palavras induzida pelo documento  $d$  e a distribuição de palavras induzida pela classe  $c$ . A “cross-entropia” pode ser interpretada como o número médio de bits necessários para codificar uma palavra do documento  $d$  usando uma codificação otimizada para a distribuição de probabilidade das palavras da classe  $c$ .

Documentos mais complexos exigem mais bits em média para serem codificados. Com o intuito de tornar o classificador mais sensível a essas diferenças entre

diversos documentos, é utilizado o conceito da “Divergência Kulback-Leibler”, que subtrai o número médio de bits necessário para codificar cada documento por meio da sua codificação ótima:

$$\propto \frac{\log_2(p(c))}{n} + \sum_{i=1}^T p(w_i | d) \log_2 \left( \frac{p(w_i | c)}{p(w_i | d)} \right) \quad (2.11)$$

Até o momento não houve alteração nos cálculos resultantes do método Naive Bayes e sim uma melhor distribuição dos resultados.

Um dos problemas no método Naive Bayes é que as palavras que não ocorrem no conjunto de treinamento são ignoradas. Neste caso o termo  $p(w_i | c)$  pode assumir valor zero. Para solucionar esse problema são aplicadas técnicas de suavização.

Suavização é uma técnica utilizada para melhorar a estimativa das probabilidades, quando há insuficiência de dados. A palavra *smoothing* vem do fato destas técnicas tenderem a aumentar os valores baixos e diminuir os valores altos de probabilidade, sem fornecer valores extremos como “0” e “1”. A técnica de suavização mais simples é de Laplace [8, 5, 4], que considera a quantidade de ocorrências de uma palavra mais um. Porém será utilizada a técnica de suavização (*smoothing*) de Witten-Bell ([2]), utilizada por Craven [8]. Essa técnica de suavização resulta em atribuir peso maior para a ocorrência das palavras únicas e peso menor quando houver repetição de palavras.

A aplicação da técnica suavização de Witten-Bell requer que seja feita uma pré-análise sobre a palavra  $w_i$  a ser considerada. Se a palavra  $w_i$ , do documento, ocorreu no conjunto de treinamento ( $N(w_i, c) \neq 0$ ) é aplicada a equação 2.12:

$$p(w_i | c) = \frac{N(w_i, c)}{T_c + \sum_{j=1}^{T_c} N(w_j, c)} \quad (2.12)$$

Se, por outro lado, a palavra  $w_i$ , do documento não ocorreu no conjunto de treinamento ( $N(w_i, c) = 0$ ) é aplicada a equação 2.13:

$$p(w_i|c) = \frac{T}{T_c + \sum_{j=1}^{T_c} N(w_j, c)} \frac{1}{T - T_c} \quad (2.13)$$

Assim, a aplicação da técnica de suavização de Witten-Bell, considerando as equações 2.12 e 2.13 , resulta na equação 2.14 para classificação de documentos ( $d$ ):

$$p(c|d) = \frac{\log_2 p(c)}{n} + \sum_{i=1}^T p(w_i|d) \cdot \log_2 \left( \frac{p(w_i|c)}{p(w_i|d)} \right) \quad (2.14)$$

em que,

$d$  é o documento a ser analisado =  $(w_{i_1}, w_{i_2}, \dots w_{i_n})$ ;

$c$  é uma classe de  $C$ ;

$p(c|d)$  é a probabilidade de ocorrer a classe  $c$ , dado o documento  $d$ ;

$p(c)$  é probabilidade de ocorrência da classe  $c$ ;

$n$  é quantidade total de palavras no documento;

$w_i$   $i$ -ésima palavra do conjunto de treinamento a ser analisada;

$N(w_i, d)$  é a quantidade de ocorrências da palavra  $w_i$  no documento  $d$ ;

$p(w_i|d)$  é a probabilidade da palavra  $w_i$ , dado o documento  $d$ ,  $\left( \frac{N(w_i, d)}{n} \right)$ ;

$p(w_i|c)$  é a probabilidade da palavra  $w_i$ , dado o conjunto de treinamento de dados da classe ( $c$ ) em análise, resultado da aplicação da equação 2.12 ou da equação 2.13;

$N(w_i, c)$  é a quantidade de ocorrências da palavra  $w_i$  no conjunto de treinamento para a classe  $c$ ;

$T_c$  é a quantidade de palavras únicas existentes no conjunto de treinamento da classe  $c$ ;

$T$  é a quantidade total de palavras únicas existentes no conjunto de treinamento.

# Capítulo 3

## MINERADOR WEB

Neste capítulo é apresentada a descrição da arquitetura e do funcionamento do Minerador Web e o processo de busca da informação. Na seção 3.1 é apresentada a arquitetura do Minerador Web. Na seção 3.2 é descrito o processo de geração da meta informação. Na seção 3.3 é descrito o processo de busca e armazenamento da informação.

### 3.1 A arquitetura do Minerador Web

O processo usual de pesquisa de informações executada por um usuário utilizando uma relação de endereços de *sites* fornecida por um portal de busca, consiste em “carregar” um *site* e analisá-lo visualmente. Caso a informação procurada não esteja disponível nesse *site*, repete-se a operação sobre o próximo *site* da relação e assim por diante. Esse processo manual de busca pela informação desejada acaba sendo demorado e cansativo, dependendo da quantidade de *sites* que devem ser analisados. Uma das funções do Minerador Web é agilizar esse processo de análise visual de *sites* na Web, utilizando técnicas de extração e classificação de informações contidas nas páginas HTML. As informações extraídas são inseridas numa base de conhecimento que, juntamente com um conhecimento de fundo disponível previamente, permite ao usuário realizar consultas dentro do contexto de um domínio escolhido.

O funcionamento do sistema divide-se em dois processos denominados de “geração de meta-informações” e de “busca da informação”, os quais podem ser visualizados na figura 3.1.

O processo de geração de meta-informações tem como objetivo definir uma ontologia para um domínio de conhecimento. A ontologia é definida por meio de um processo de análise do conteúdo de *sites* típicos do domínio. Em linhas gerais, o processo consiste em selecionar *sites* para análise (coleta de *sites*), identificação dos conceitos típicos (categorização), definição da ontologia e descrição formal da ontologia por meio de uma DTD.

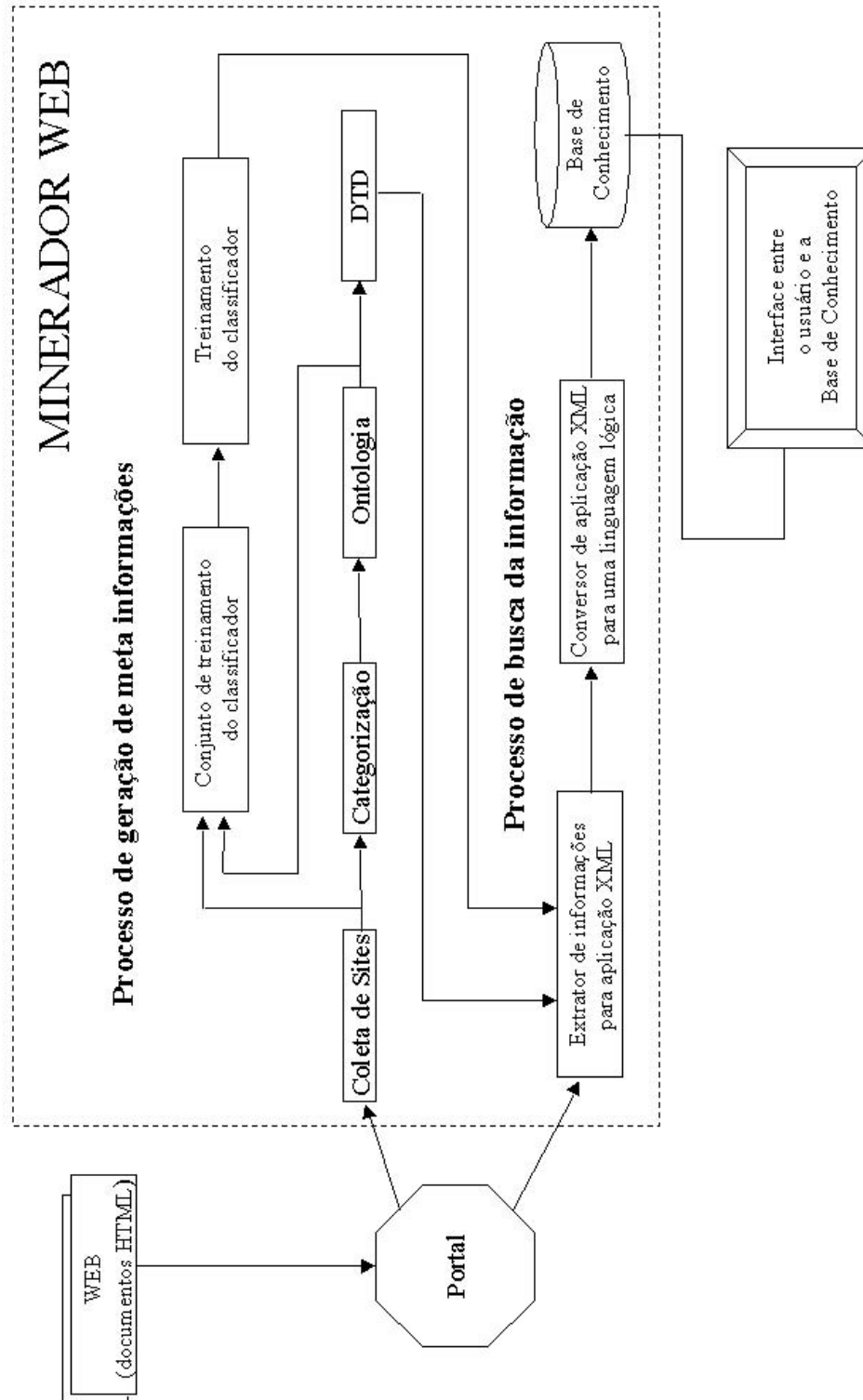


Figura 3.1 – Arquitetura do Minerador Web

Com as informações coletadas, a próxima etapa consiste em categorizá-las. A categorização consiste em agrupar as informações que possuem um relacionamento conceitual, como por exemplo: o número “7334” não possui significado quando tratado isoladamente. Porém, quando analisado junto com o nome de uma via, por exemplo “Long Island”, o conjunto fornece uma informação sobre o conceito de endereço. Mesmo a via “Long Island” tendo significado próprio, as informações “Long Island” e “7334” devem ser agrupadas, formando uma categoria que pode ser denominada de “endereço”.

## 3.2 Processo de geração da meta informação

O produto final deste processo é uma ontologia de empresa, que após ser escrita formalmente será utilizada no Minerador Web, como parâmetro para busca de informações.

O processo de criação de uma ontologia consiste de duas etapas, a abstração dos elementos de um domínio do conhecimento e a descrição desses elementos. A primeira etapa, abstração dos elementos de um domínio do conhecimento, consiste em identificar objetos e relações entre esses objetos, formando a conceituação do domínio. A segunda etapa, descrição desses elementos, consiste em especificar a conceituação por meio de uma linguagem informal, que descreva informações pertinentes ao contexto e ao domínio. A utilização em programas, da ontologia criada, requer a descrição formal dos objetos e suas inter-relações, e isto é feito descrevendo a ontologia em uma *Document Type Definition* (DTD).

A abstração dos elementos de um domínio tem início com a identificação de um domínio do conhecimento e a coleta de *sites* relacionados à esse domínio. O domínio escolhido foi o domínio do conhecimento de empresas. Tendo sido definido o domínio o processo tem início com a etapa de coleta de sites.

### 3.2.1 Coleta de *sites*

A coleta de sites constitui-se na busca de 3.210 endereços de *sites* diversos, coletados no portal “Yahoo”, na categoria “*Business to Business*” no endereço:

[http://dir.yahoo.com/Business and Economy/Business to Business/](http://dir.yahoo.com/Business_and_Economy/Business_to_Business/)

focando apenas *sites* referentes a empresas distribuídos em 66 categorias.

Desses 3.210 endereços de *sites* foram selecionados, aleatoriamente, 50 endereços e que se distribuía em 16 categorias. Essa quantia de 50 endereços de *sites* resultou em 1.316 páginas de documentos *HyperText Markup Language* (HTML). Essa quantidade de páginas foram consideradas razoáveis para início do processo de abstração. As relações, dos 3.210 endereços de *sites* e dos sites selecionados, se encontram no cdrom anexo com os nomes “PrimeiraFase\_3120Sites.tex” e “PrimeiraFaseSelecionados.tex”, respectivamente.

Tendo sido definido um domínio e coletado os *sites* a próxima etapa consiste em abstrair os elementos do domínio delimitado (domínio do conhecimento empresa). Essa etapa é denominada de “Categorização”.

### 3.2.2 Categorização

A categorização constitui-se na análise visual das 1.316 páginas coletadas, sendo utilizado para análise navegadores e editores de texto, como Word e TexPad, para leitura do código fonte das páginas. Durante esse processo de leitura do código fonte, foram identificadas páginas de *sites* compostas somente de *links*, que não permitem, em uma primeira vista, a identificação de seu conteúdo. Sendo assim, foram necessários a busca de páginas complementares por meio de seus *links*.

A análise visual das páginas oferecem dificuldades em se identificar informações úteis, relevantes a este trabalho. Essa dificuldade é notada quando na leitura do código de um documento são encontradas marcações de estilo como pode ser visualizada na figura 3.2.

Como primeiro passo para minimizar essa dificuldade, optou-se pela eliminação dessas marcações de estilo, a qual resultou em um documento com informações desconexas, confusas e sem sentido, prevalecendo a dificuldade no entendimento. O resultado obtido pode ser visualizado na figura 3.3.

Através de nova análise visual sobre os documentos pode ser constatado que algumas marcações eram necessárias para identificar uma informação. Dessa forma, o programa foi implementado para produzir 4 sub-conjuntos dos conteúdos dos *sites* analisados, relacionados e exemplificados a seguir:

1. sub-conjunto de *Links*:

```

<P ALIGN="CENTER"> <B> Extreme Careers <BR> </B>
Stargate.net, Inc. <BR> 40 24 <SUP> th </SUP> Street
<BR> Crane Building, Suite 300 <BR> Pittsburgh,PA 15222
</P> <P ALIGN="CENTER"> EEO/AA </FONT> </TD>
</TR> </TABLE> </TD> </TR> </TABLE> </TD> <TD
WIDTH=210 BGCOLOR="#CCCCCC"> <TABLE BORDER=0
CELLSPACING=0 CELLPADDING=5 WIDTH=210> <TR>
<TD> <P ALIGN="CENTER"> <B> <FONT SIZE=-1"FA-
CE="Arial,Helvetica,Univers,Zurich BT"> Stargate Consulting Careers
</P> <P ALIGN="CENTER"> </FONT> </B> <FONT SIZE=-
1"FACE="Arial,Helvetica,Univers,Zurich BT"> Stargate's Professional
Services Group currently has open <BR> IT consulting positions. These
technical positions will be delivered primarily on-site at our client's
facilities. Click on <BR> the <B> (PSG) </B> positions above for more
details. </P>

```

Figura 3.2 – Texto extraído do código do *site* “www.stargate.com”, em formato original

```

Extreme Careers Stargate.net, Inc. 40 24th Street Crane Building, Suite
300 Pittsburgh, PA 15222
Stargate Consulting Careers Stargate's Professional Services Group cur-
rently has open IT consulting positions. These technical positions will be
delivered primarily on-site at our client's facilities. Click on the (PSG)
positions above for more details.

```

Figura 3.3 – Texto extraído do código do *site* “www.stargate.com”, com as marcações removidas

Contém informações que permitem acessar outros sites, como por exemplo:

```

default.htm,
nav.js,
http://www.aerojetfinechemicals.com/,
http://www.gencorp.com/vs.html

```

## 2. sub-conjunto de **Códigos não HTML**:

Contém marcações que se referenciam a outros aplicativos como “OBJECT”, “APPLET” e “SCRIPT”, exemplo:

```

<script: – function MM_reloadPage(init)
{ //reloads the window if Nav4 resized
if (init==true) with (navigator) ...; // -}

```

### 3. sub-conjunto **Imagens:**

Contém marcações que se referenciam a imagens como:

```
<IMG alt=
src=" pacifictiresales – Home_arquivos/000_YES_Row1_Left_Selected.gif"
border=0>
```

### 4. sub-conjunto **Informações:**

Contém marcações que não são chamadas a outros endereços de *sites* (*links*), chamadas a imagens ou chamadas a aplicativos diversos. Exemplo de um documento resultante:

```
<title>AEROSPACE Composite Products Catalog Home Page
<meta>name= "CreatedBy" content= "Deltronix Enterprises"
<META>Name= "description" Content= "Advanced Composite ...
This Site is being Slowly Remodeled ! Please excuse our dust!
<TABLE>Carbon Laminates
<TR><TD>UNIDIRECTIONAL
<TD>GRAPHITE PLATES
<TD>TAPERED
<TD>WOVEN Fabricated Products
<TABLE><TR>
<TD>COMPOSITE PANELS
<TD>FIBERGLASS LAMINATES
...
```

Analisando visualmente esses sub-conjuntos, em especial o último da lista anterior, constatou-se que algumas informações se agrupam entre determinadas marcações. Para identificar essas informações que estavam agrupadas entre marcações, foram desenvolvidos programas que auxiliaram em uma pesquisa mais apurada dos resultados. Um dos programas tinha como função organizar as marcações dos *site* em estrutura de modelo de árvore, facilitando dessa forma a identificação do conteúdo das marcações, e sendo encontradas muitas marcações de estilo sem informações úteis. Com base nesses resultados foi desenvolvido um outro programa que relacionava as marcações e seus conteúdos. O resultado desse último programa pode ser visualizado na tabela 3.1

Com relação à tabela 3.1, a coluna “Qtd.”, refere-se à quantidade de ocorrência da coluna “Informação”, contida no *site* cujo endereço consta na coluna “*Site*”. A

Qtd	Site	Categoria	Marcação	Informação
1	www.byreferralonly.com	realstate	<title>	By Referral Only
1	194.206.158.178	personalcare	<td>	contact information Info-sikosoftcom
1	www.aaicorp.com	transportation	<a>	Who We Are Guest Book Products Search Contact Us News UAV Defense Engineering Support Fluid TestACL Transportation Manufacturing AAI's Parent Company UIC Working in Maryland

Tabela 3.1 – Linhas resultante da aplicação do utilitário de extração de informações gerais

coluna “Categoria”, representa a categoria do *site*, classificada pelo portal Yahoo, e a coluna “Marcação” identifica a marcação anterior à informação. Vale ressaltar que a coluna “Informação” pode estar vazia, pois existem muitas marcações que são utilizadas apenas para estruturar o texto, não contendo entre o início e fim da marcação alguma informação útil.

O resultado dessa análise identificou a localização de informações relevantes das páginas que pudessem contribuir no processo de abstração. Algumas dessas informações se concentram entre marcações específicas. Abaixo estão relacionadas as marcações identificadas como relevantes no processo de abstração, juntamente com sua incidência nos documentos e o resultado da análise:

- <title> e </title>:  
O conteúdo destas marcações é apresentado no cabeçalho dos browsers. Existe somente um conjunto dessas marcações por página e foram encontradas em 94% dos sites analisados, distribuídos conforme a tabela 3.2.
- <meta>:  
O uso desta marcação nas páginas é opcional. Na análise foram encontradas, em média, 3 marcações “meta” por página. Dos sites analisados 78% possuíam a marcação distribuídas conforme a tabela 3.3.
- <td> e </td>:  
Essas marcações fornecem um visual estruturado da página e apresentam

perc.	Comentários
82%	dos conteúdos possuíam o nome da empresa;
26%	faziam referência à atividade da empresa;
22%	possuíam palavras chave sobre a atividade da empresa;
36%	possuíam o nome da página, que não faziam referência alguma ao seu conteúdo e
62%	possuíam o nome da página, com referência a seu conteúdo.

Tabela 3.2 – Percentual da marcação “title” nos documentos HTML analisados

perc.	Comentários
02%	possuíam a descrição das atividades da empresa;
54%	possuíam palavras chaves referente às atividades da empresa;
54%	repetiam o conteúdo das marcações <title> e </title> e
72%	possuíam outras informações não relevantes a este trabalho

Tabela 3.3 – Percentual da marcação “meta” nos documentos HTML analisados

um agrupamento de informações relacionadas como textos, *links* e imagens. foram encontradas em 74% das páginas, sendo que 46%, desse total, não possuem informações relevantes com referência ao contexto estudado, pois ou não contém conteúdo ou quando possuem fazem referências a atributos de estilos.

- <p> e </p>:  
Essas marcações constam em 82% das páginas e identificam sub-conjuntos de um texto.
- <blockquote> e </blockquote>:  
Essas marcações foram encontradas em 36% das páginas e possuem a mesma funcionalidade das marcações <p> e </p>
- <a e </a>:  
O conteúdo dessas marcações agrupa um conjunto de informações sobre multimídia, normalmente identificada por um texto compreensível ao ser humano. Informações multimídia, por que faz referência a *links* para imagens e outras páginas. Não foi identificado nenhum áudio na análise, mas é nesta marcação que devem ser encontrados. Essas marcações constam em 86% dos sites.
- <pre> e </pre>:

Essas marcações delimitam um texto que não será alterado quando visualizado por algum navegador, foram identificadas em 16% dos sites

- $\langle Hn \rangle$  e  $\langle /Hn \rangle$

Onde  $n$  representa os números na faixa de 1 a 6 (inclusivos). As informações contidas entre as marcações  $\langle Hn \rangle$  e  $\langle /Hn \rangle$  formam o título sobre o conjunto de frases que se segue. Foram encontrados em 38% dos sites.

Outros levantamentos foram efetuados com relação as palavras significativas, isto é, que claramente informam o significado do conteúdo, o resultado pode ser visualizado na tabela 3.4.

Palavras	Incidência	Observações
Order	48%	Normalmente antecedem a chamada a um formulário
Mailto	78%	12% se referem ao webmaster, o restante a contatos da empresa
Http	78%	Possuem sites relacionados ao site principal
Contact	52%	Quando não encontradas em menu, agregam e-mail/fone/fax
Product	72%	Estão contidos em frases que descrevem sobre as características de produtos
About	62%	Normalmente são palavras que antecedem <i>links</i> para páginas que descrevem as atividades ou apresentam o histórico das empresas
Call	62%	Antecedem números de telefones
\$	42%	Antecedem uma seqüência numérica. Sua proximidade a outro símbolo identifica uma faixa de valores numéricos (preços).
Phone	72%	Normalmente antecede um número telefônico
Fax	70%	Encontrado próximo da palavra Phone, seguida de uma seqüência de número telefônico
$\langle form$	54%	Identificam uma área de entrada de dados
News	46%	Novidades/notícias/acontecimentos recentes sobre a empresa
Copyright	22%	Identificação de conteúdo restrito a direitos autorais
Locations	28%	Normalmente seguidos de endereços
Franchise	16%	Grande incidência nas empresas com atividades de franchise

Tabela 3.4 – Percentual de diversas palavras significativas

Essa análise permitiu identificar a estrutura de algumas informações (que serão detalhadas nas próximas seções) como a estrutura de um endereço, a identificação do nome da empresa, o formato de um número de telefone, e a localização do nome do produto em um *site*.

Extreme Careers <>< / > Stargate.net, Inc. <> 40 24 <> th < / >  
 Street <> Crane Building, Suite 300 <> Pittsburgh, PA 15222 < / ><>  
 EEO/AA < / ><> Stargate Consulting Careers < / ><> Stargate's  
 Professional Services Group currently has open <> IT consulting positions.  
 These technical positions will be delivered primarily on-site at our client's  
 facilities. Click on <> the <> (PSG) < / > positions above for more  
 details.

Figura 3.4 – Texto extraído do código do *site* “www.stargate.com”, com a simbologia das marcações

Como já mencionado, nem todas as informações se encontram entre marcações específicas. Sendo assim, os *sites* foram novamente analisados de uma forma menos rígida, isto é, a análise foi feita somente sobre as informações úteis, independente do conjunto de marcações.

Para isso, o programa que antes eliminava as marcações passou a retirar apenas o conteúdo das marcações (o nome e os atributos), preservando a informação da presença e posicionamento, mas excluindo o conteúdo das marcações, obtendo a seqüência apresentada na figura 3.4.

Isso permite a criação de melhores heurísticas para a identificação de grupos de informações como, por exemplo a informação de nome da empresa e endereço, tendo com base a figura 3.4:

- **Nome da empresa:** < / >Stargate.net, Inc.<>;
- **Endereço:** <>40 24<>th< / >Street<>Crane Building, Suite 300 <> Pittsburgh, PA 15222< / >.

As marcações < *title* > e < /*title* >, foram preservadas por dois motivos. O primeiro motivo é o fato de ser uma informação visível, mesmo não existindo no corpo do *site* apresentado pelo navegador, e o segundo motivo é o fato que muitos *sites* colocam informações úteis entre essas marcações.

Com essa nova estratégia foi possível obter com mais precisão a relação de objetos e o formato com que aparecem nas páginas dos *sites*. Essa relação pode ser visualizada parcialmente na tabela 3.5.

Sobre essa nova visão de análise, a categorização identificou os seguintes elementos:

- Endereço de acesso ao *site* da empresa;

Elemento	Objetos	Condições
Nome da empresa	Pacific Tires Sales, Inc.	<> PACIFIC TIRE SALES, INC. <>
Logradouro	8294 Bolsa Ave	<>8249 BOLSA AVE <>
Cidade	Midway City	<>MIDWAY CITY, CA. 92655 <>
Estado	CA.	<>MIDWAY CITY, CA. 92655 <>
Zona Postal	92655	<>MIDWAY CITY, CA. 92655 <>

Tabela 3.5 – Resultado, parcial, da etapa de categorização

- Nome da Empresa;
- Endereço de localização da empresa, compreendendo: nome da via, complemento, nome da cidade, código postal, unidade da federação;
- Meios de comunicação da empresa, compreendendo: números dos telefones, fax e endereços de *e-mail*;
- Nome dos Produtos, especialização, características e preço do produto;
- Ramo de atividade;
- Atividade.

Tendo sido completada a etapa de categorização, identificando a conceituação do domínio do conhecimento de empresas, o próximo passo é a especificação dessa conceituação, denominada de ontologia.

### 3.2.3 Criação da ontologia de empresas

A ontologia, como já descrito anteriormente, compreende a descrição informal dos elementos da conceituação. Essa descrição deverá conter informações relevantes ao contexto. Sendo utilizado como parâmetro, uma descrição conceitual do elemento e a forma com que esse elemento se encontra nas páginas dos *sites*.

A seguir serão especificados os elementos da conceituação:

- **Site:**  
Endereço de acesso à empresa por meio da Web. Uma empresa pode possuir diversos endereços de acessos (*Uniform Resource Locators* (URL)), porém para o Minerador Web a URL capturada para identificação dos elementos é a que será identificada como *Site*;

- **Nome da Empresa:**

Identificação unívoca da empresa. Normalmente a seqüência de caracteres que a identifica. Termina com um termo que se referencia ao tipo de sociedade comercial, como por exemplo: “com”, “inc.”, “company”.

A seqüência pode ser encontrada no corpo do *site* em grande quantidade ou entre as marcações “<title>” e </title>. O formato no qual as seqüências podem ser encontradas, variam entre abreviadas, completas ou não, como por exemplo:

- Entre as marcações de “title”

```
<title>Hammer Lithograph Corporation</title>
```

- Com outras informações

```
<title>Mimigraphics, Inc. - Flash Animation< //title>
```

- No corpo do *site*

```
Burst of Growth For Stargate Data Center Business ...
```

- Na forma abreviada

```
C & K Industries</title>
```

Em que “C & K” correspondem à abreviatura de “CandeKent”.

- **Endereço:**

Local (físico) onde se encontra a empresa. Possui os seguintes sub-elementos:

- **UF**, sigla da unidade da federação. É formada por duas letras, sendo a primeira maiúscula, isoladas à direita por um espaço e uma seqüência de dígitos (referente à Zona Postal), no formato “UF nnnnn” ;
- **Zona Postal**, código postal equivalente ao Código de Endereçamento Postal (CEP). Pode ser identificado logo após o sub-elemento “UF”. Possui de 5 a 9 dígitos. Sendo que quando contém 9 dígitos, existe o símbolo separador ‘-’, entre os 5 primeiros dígitos e os 4 últimos dígitos;

- **Cidade**, inicia e termina com caracteres alfabéticos. Antecede o sub-elemento “UF”;
- **Complemento**, pode não ocorrer. Caso contrário ocorre antes do sub-elemento “Cidade”, iniciando com caracteres alfabéticos, podendo terminar com dígitos;
- **Via**, inicia com dígitos e termina com caracteres alfabéticos. Ocorre antes de um dos sub-elementos “Cidade” ou “Complemento”.

Exemplo da ocorrência do elemento “Endereço” no *site*:

Sem o sub-elemento “Complemento”,

```
<>8249 Bolsa< / >Avenue.< / ><><>< / ><>MidWay
City,<><>Ca 92655-1244< / >
```

Com o sub-elemento “Complemento”,

```
<>8249 Bolsa< / >Avenue.< / ><>Unit J. Regatta Plaza<><
/ ><>MidWay City,<><>Ca 92655-1244< / >
```

- **Comunicação:**

Meios eletrônicos de contato de uma empresa, contendo os seguintes sub-elementos:

- **Telefone/Fax**, foi constatado, por processo empírico, na etapa de categorização, a presença de termos que antecedem os números de telefone e fax, como por exemplo: *call, phone, telephone, tel, voice, toll free* e *tell me*. Podem ser formados de 2 a 4 conjuntos de caracteres alfanuméricos, tendo como separador dos grupos o sinal de menor ‘-’, espaços, um par de parênteses ou pontos. Exemplos dos formatos do sub-elemento “Telefone/Fax”:

4 conjuntos numéricos	1-800-969-7829
4 conjuntos alfanuméricos	1-800-969-STAX
2 conjuntos numéricos	773-8839
3 conjuntos numéricos com parênteses	(323) 889-3343

Exemplos da ocorrência do sub-elemento “Telefone/Fax” em um *site*:

```
<>Phone: 860-684-4281<>Fax: 860-684-5913<>
```

- **Email**, possui a seguinte formação: contém um símbolo ‘@’. À esquerda e à direita do símbolo contém caracteres alfanuméricos e opcionalmente os símbolos “- . \_”. Sendo que à direita do símbolo ‘@’ deverá conter no mínimo um símbolo ‘.’. Qualquer seqüência com as características anteriormente descritas e que possuam a palavra “webmaster”, não corresponde a um *e-mail* válido. Exemplo de um endereço de *e-mail*:

<>robbos@senet.com.au< / >

- **Produto:**

Tudo que a empresa produz e/ou oferece para venda no mercado. Possui os seguintes sub-elementos:

- **Nome do Produto**, identifica o produto oferecido. É formado por uma seqüência de caracteres alfanuméricos;
- **Especialização**, modelo, uma peça específica ou ainda uma diferenciação na prestação do serviço. Como exemplo, tem-se:

<b>Produto</b>	<b>Especialização</b>
“tire”	“195/75r14”
“tire”	“Brakes”
“plan of health”	“Master”

- **Características**, detalhes ou comentários sobre a especialização. Exemplo de uma característica para o produto “tire”, com especialização “195/75r14”, característica “P195/75R14 92S SP 20 A/S TACOMA 4x4 96-99”. As características contém o nome da especialização, podendo, as características, estarem no conjunto de caracteres antes e após a especialização. Caso o conjunto de caracteres identificado possuir no mínimo 3 conjuntos de no máximo 2 palavras e uma vírgula, o conjunto será desconsiderado por ter as características de uma lista de palavras ou no caso uma lista de produtos.
- **Preço**, valor da especialização, pode ser encontrado próximo à especialização ou contido no conjunto de caracteres que formam as características. Inicia com o símbolo “\$” seguido de dígitos.

- **Ramo de Atividade:**

Representa a área de atuação de uma empresa, em linhas gerais identifica

o relacionamento entre a empresa e o produto fornecido. É uma informação implícita, não possui uma estrutura definida, é uma categoria composta por um conjunto de classes. Uma empresa está associada a uma única classe. Essa classe refere-se à área de atuação da empresa, como por exemplo:

- Uma fábrica de auto peças, será classificada como “Ramo de Atividade”: Automóvel;
- Uma revenda de calças, será classificada como “Ramo de Atividade”: Vestuário.

- **Atividade:**

Em linhas gerais identifica a relação entre o tipo de produto oferecido pela empresa e o cliente. É uma informação implícita, não possui uma estrutura definida, sendo composta por um conjunto de classes. Uma empresa está associada a pelo menos uma classe. A classe caracteriza a empresa em relação ao tipo de serviço realizado, podendo ser:

- Fabricante, se a empresa fabrica seus produtos;
- Revendedora, se a empresa revende os produtos, próprios ou de fabricantes;
- Prestadora de serviço, o produto é o resultado da aplicação de mão de obra, para a execução de uma tarefa.

Para utilização desta ontologia, se faz necessário descrevê-la em linguagem formal, formando uma DTD.

### 3.2.4 Criação da DTD

Na figura 3.5, pode ser visualizada a DTD originada da descrição formal da ontologia.

Como pode ser identificado na criação da ontologia para o domínio empresa, as informações explícitas podem ser identificadas visualmente nos *sites*, pela forma com que estão estruturadas. Sendo assim atendidas pelas etapas de coleta, categorização, criação de uma ontologia e criação de uma DTD. Porém as informações implícitas, como “Ramo de Atividade” e “Atividade” não aparecem explicitamente no texto das páginas de um *site*. As informações implícitas, neste

```

<!-- DTD para criação de uma base de conhecimento em Prolog baseado
em paginas HTML de Empresas -->
<!ELEMENT empresa (site, nomeEmpresa, ramoAtividade, contato*,
produto+)>
<!ELEMENT site (#PCDATA)>
<!ELEMENT nomeEmpresa (#PCDATA)>
<!ELEMENT ramoAtividade (nomeRamo, nomeAtividade+)>
<!ELEMENT nomeRamo (#PCDATA)>
<!ELEMENT nomeAtividade (#PCDATA)>
<!ELEMENT contato (endereco*, comunicacao*)>
<!ELEMENT endereco (logradouro, cidade, estado?, zpt?)>
<!ELEMENT logradouro (#PCDATA)>
<!ELEMENT cidade (#PCDATA)>
<!ELEMENT estado (#PCDATA)>
<!ELEMENT zpt (#PCDATA)>
<!ELEMENT comunicacao (telefone?, fax?, email*)>
<!ELEMENT telefone (#PCDATA)>
<!ELEMENT fax (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT produto (nome, especializacao*)>
<!ELEMENT especializacao (nome, caracteristicas*, preco*)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT caracteristicas (#PCDATA)>
<!ELEMENT preco (#PCDATA)>

```

Figura 3.5 – DTD da ontologia de empresas

contexto, são categorias compostas de classes, como exemplo para a categoria “Ramo de Atividade” composta pelas classes “Automóvel”, “Saúde”, Vestuário etc.

A abordagem utilizada pelo Minerador Web visa determinar essas informações por meio de classificação do *site* sobre conjuntos de ramos de atividade e de atividades previamente determinados. Considerando as páginas de um *site* como sendo um único documento HTML, o problema se torna equivalente ao de classificação de textos.

Para essa abordagem foram identificadas mais duas etapas no processo de geração da meta informação, a criação do conjunto de treinamento e o treinamento desse conjunto.

### 3.2.5 Criação do conjunto de treinamento

A busca da informação implícita será feita com a utilização de classificadores, desenvolvidos com base no método Naive Bayes. O método Naive Bayes é um método probabilístico, tendo como evento as palavras que caracterizam um documento. O processo de busca divide-se em duas etapas. A primeira etapa, criação de um conjunto de treinamento, que deve atender às necessidades do classificador, sendo que um dos parâmetros a serem obtidos na categorização são as quantidades de ocorrência das palavras.

Para atender essa nova abordagem, o programa desenvolvido para retirar o conteúdo das marcações foi incrementado por uma nova função, a de gerar uma lista de palavras únicas (sem considerar a quantidade de ocorrência das palavras) e a frequência de ocorrência de cada palavra no *site* analisado.

Vale ressaltar que, existem programas prontos que extraem as marcações. Porém uma rápida análise sobre esses programas identificou a impossibilidade de, a curto prazo, adaptá-los aos processo do Minerador Web. Sendo assim, o desenvolvimento de um novo programa de extração de marcações foi mais viável.

Sobre esse enfoque, de se utilizar um classificador, os sites foram analisados sobre os critérios do elemento “Ramo de Atividade” sendo agrupados em classes. Essa primeira análise mostrou que a tarefa de identificação das classes é complexa, pois na Web existe uma grande variedade de tipos de *sites*, como *sites* comerciais, filantrópicos, educacionais, pessoais, que possuem pouca informação em comum

entre eles, mesmo sendo considerados *sites* do domínio do conhecimento empresa. Além do que, os 50 *sites* distribuídos em 16 categorias não apresentam resultados satisfatórios com relação a identificação dos elementos implícitos.

Visando buscar uma melhor conceituação sobre esses elementos implícitos, uma nova a estratégia foi adotada, sendo selecionado um volume maior de *sites*, distribuídos em poucas categorias e que se restringissem apenas a *sites* de empresas relacionadas ao comércio de seus produtos. Para isso foi aplicada a técnica de escolha aleatória sobre todos os endereços de *sites*.

A Web é composta de centenas de milhares de páginas, distribuídas em diversos endereços virtuais, sendo contraproducente a coleta desses endereços. Como proposta inicial para essa nova estratégia, optou-se por acessar o portal Yahoo [47]. Esse portal oferece uma relação de 95 categorias de *sites* relativos ao comércio em geral, enumerados na categoria “*Business and Economy*”, sub-categoria “*Shopping and Services*”.

De cada um dos *sites* da categoria “*Shopping and Services*”, foram obtidas informações de referências que constam no Portal Yahoo. A figura 3.6 apresenta o conjunto de informações de *sites*, no formato em que são encontradas quando visualizado seu código, e que podem ser obtidas para cada categoria. No caso em particular, na figura 3.6, podem ser visualizadas as informações para o *site* “Autoanything.com”, com os seguintes campos:

1. Número seqüencial, para acesso futuro às informações;
2. A categoria do *site* no Portal Yahoo;
3. A atividade, ou o nome popular (“nome fantasia”) da empresa, ou ainda, uma referência curta relativo ao *site*;
4. Uma descrição resumida sobre o que pode ser encontrado no endereço, referências a produtos, “slogan”, entre outras informações. Esta posição é opcional;
5. O endereço de acesso ao *site* (URL).

Essas referências foram organizadas em uma base de dados, no formato apresentado na tabela 3.6

```

<LI><A href="http://srd.yahoo.com/S
=7664877:D0/R=1/CS=7664877/SS=30955060/*http://www.
autoanything.com/">Autoanything.com</A> -
features name brand accessories such as car and seat
covers, wood dash trim, grill guards, and floor mats.
</LI>

```

Figura 3.6 – Informações obtidas, em formato de código HTML, da categoria *Shopping and Service*

Posição	Conteúdo
1	21111
2	yahoo! business and economy>shopping and services>automotive>accessories
3	autoanything.com
4	features name brand accessories such as car and seat covers, wood dash trim, grill guards, and floor mats.
5	http://www.autoanything.com/

Tabela 3.6 – Exemplo da estrutura de uma referência a endereço de *site*, obtida no portal Yahoo

Foram coletadas 63.805 informações de referências dos *sites* das categorias, que podem ser visualizadas no arquivo denominado “ReferenciasEnderecos.txt”, no cdrom anexo a esta dissertação.

Sobre essa relação de endereços foi aplicado um processo randômico de seleção, para obter, inicialmente, 100 endereços de *sites*, e acumulativamente 200, 300, 400 e 500, tendo sido abrangido inicialmente 30 categorias. O utilitário utilizado no processo de seleção randômica, foi desenvolvido em linguagem Java, e seu código pode ser visualizado na figura 3.7.

Porém ainda que a razão entre a quantidade de *sites*/categoria, tenha sido aumentada de 3,125 para 3,33, existiam poucas informações que pudessem caracterizar uma classe, existindo classes que possuíam um único *site* relacionado, em relação a outras classes que possuíam 5 *sites*. Sendo assim, foi aplicada uma seleção mais ampla, isto é, foram selecionadas aleatoriamente 10 categorias e sobre essas dez categorias foram selecionados 10 endereços. A relação desta última seleção pode ser encontrada no arquivo “ListaEnderecos\_10Cat.tex”, no cdrom anexo.

Nesses 100 *sites*, foram coletadas 6.922 páginas, totalizando 25.342 palavras únicas e 1.037.492 ocorrências das palavras únicas. Esse volume de informação a ser

```

public EscolherCategoria(int Maximo, int Qtd) {
    // Qtd, quantidade máxima de valores a serem selecionados
    // Maximo, valor máximo da faixa de valores a ser selecionado
    Maximo--;
    // vSelecao, contém a relação dos números selecionados
    Vector vSelecao= new Vector();
    String sApoio="";

    while (Qtd > 0)
        if (vSelecao.indexOf((sApoio= " " + (int)
            (java.lang.Math.random() * Maximo ))) == -1)
            vSelecao.addElement(sApoio); Qtd--;

    for(;Qtd < vSelecao.size(); Qtd++)
        System.out.print(vSelecao.elementAt(Qtd)+" ");
}

```

Figura 3.7 – Código do método de seleção de números aleatórios.

analisado é muito grande, onerando o tempo de processamento. Sendo assim foi implementada, inicialmente, a redução das linhas que não possuíssem informação útil, como linhas que continham somente marcações de estilo. Essa redução foi aplicada sobre a relação de marcações com informação.

O resultado foi a identificação de muitas linhas irrelevantes. Um exemplo do resultado obtido com a identificação de linhas, pode ser visto na tabela 3.7.

Linha	Qtd.	Site	Categoria	marcação	Informação
1	6	www.aramark.com	hospitality	< <i>td</i> >	
2	1	www.aramark.com	hospitality	< <i>td</i> >	SearchTips
3	1	www.crystallex.com	mining	< <i>td</i> >	January 29 2003

Tabela 3.7 – Exemplo de linhas irrelevantes, capturadas no processo de coleta de informações no *site*

Com relação à tabela 3.7, tem-se:

- Linha 1, não existe informação. Existem dois motivos para esse resultado: O primeiro motivo é com relação à página de origem, em que poderia haver marcações de estilo e/ou estrutura, as quais não eram as procuradas pelo utilitário. O segundo motivo, refere-se ao código da página não ter sido implementado corretamente pelo desenvolvedor, com por exemplo o desenvolvedor não ter colocado marcações de término (“< /... >”), ou ter

preenchido com caracteres que não são “imprimíveis”, o que acontece com alguns caracteres de controle e alguns caracteres que correspondem a valores superiores a 127, com referência à tabela *American Standard Code for Information Interchange* (ASCII). Vale ressaltar que, um navegador ignora tais erros, ignorando também a função que deveria aplicar.

- Linha 2, a informação identifica um processo comum à maioria dos *sites* (dicas de pesquisa). A informação “*SearchTips*” por estar agrupando duas palavras, que quando forem analisadas por um sistema, deverá ser tratada de forma diferenciada para poder ser identificado seu significado, exigindo o desenvolvimento de funções, que devem identificar tais palavras e separá-las, onerando o tempo do processo.
- Na linha 3, a informação refere-se a uma data. Caso essa data pertença a um relatório, as informações referente ao relatório serão visualizadas nas próximas linhas.

Para suprimir essas linhas, irrelevantes, foi desenvolvido um programa que preservava as linhas que continham pelo menos uma palavra na lista de palavras únicas.

Porém, nem todas as palavras capturas possuem significado, como por exemplo as seguintes palavras: *zccm*, *zdnnet*, *ultragator*, *udlp*, *udm*, *udmercy*, *aaa*, *äää*, *ääääíñôâê* e *aideentertainment*. Essas palavras são decorrentes de erros de digitação, ou códigos inseridos pelos desenvolvedores.

Além de que, nem todas as palavras da lista de palavras únicas tem importância para a categorização.

Sendo assim, da lista de palavras únicas foram retiradas todas as “palavras” que não possuíam significado, isto é, desconhecidas ou com grafia errada. Para essa tarefa, foi utilizado o processo de comparação das palavras únicas com um dicionário eletrônico inglês/português, as palavras que não possuíam significado eram eliminadas. Para essa tarefa de comparação foram utilizados dois aplicativos, a planilha “Excel” da Microsoft e o tradutor “L & H Power Translator Pro”.

O resultado desse processo foi uma relação de palavras que possuíam um significado e com grafia correta, totalizando 10.169 palavras únicas, concentradas em substantivos, substantivos-adjetivos e verbos, como por exemplo: *fills*, *abdomen*, *bee*, *bees*, *fir*, *pumpkin*, *squash*, *aborted*, *abrasion*, *apricot*, *haunt*, *shed*, *sheds*.

Dessa lista, foram retiradas as palavras como artigos, preposições, pronomes, numerais e conectivos, que aparecem em diversos *sites*, porém não contribuem significativamente em um processo de classificação, nem auxiliam na criação de heurísticas a serem utilizadas na extração de informações dos *sites*. Essas palavras, denominadas *stop-words* ([37], [21]), podem ser removidas da seqüência de palavras a fim de diminuir a complexidade dos algoritmos de análise. Um exemplo de palavras que compõem um conjunto de “Stop-Words” pode ser visualizado na tabela 3.8

Pronomes	<i>any, some, noone, neither, nobody, all, everybody, everything, anything, nothing, I, you, he, she, it, we, they, me, him, her, us, them, my, your his, its, our, their, mine yours, hers, ours, that, towards, upward, downward, theirs</i>
Preposições	<i>of, to, from, than, with, on, for, inat, by, about, into</i>
Conectivos	<i>and, or, no, not</i>
Artigos	<i>a, an, the</i>
Numerais	<i>one, two, hundred, million</i>

Tabela 3.8 – Relação de palavras que podem compor um Stop-Word

Com essa relação de palavras depuradas, foi aplicado novamente o programa de redução das linhas. O resultado foi uma relação de sentenças que possuem algum relacionamento com o assunto que o *site* trata. Essa relação de sentenças podem ser visualizadas no arquivo “Linhas.zip”, que contém todas as linhas por *página* e no Arquivo “LinhasSemVazios”, onde estão relacionadas somente as linhas que possuem alguma informação. Sendo incrementado, no programa de redução de linhas, a função de totalização das palavras por categoria, obtendo uma relação em que pode ser verificada a percentagem de incidência das palavras nas categorias, essa relação pode ser visualizada no arquivo ‘PalavrasUnicas.2.0’.

O resumo dos resultados obtidos são apresentados na tabela 3.9. Na coluna “Palavras 100%”, contém a quantidade de palavras que tiveram 100% de incidência na categoria, e na coluna “Quant Palav”, representa a quantidade total de palavras que ocorreram na categoria.

Sobre os arquivos por categoria, que possuem apenas informações sobre o *site*, foi feita uma análise sobre o significado das palavras, procurando identificar alguma informação comum entre os *sites* e a categoria “Ramo de Atividade”. Essa análise constitui-se no processo visual de comparação das palavras do arquivo

<b>Categorias</b>	<b>Palavras 100%</b>	<b>Quant Palav</b>
Security	325	3298
Hospitality	460	3477
Internet WWW	340	3436
Jewelry	99	1030
Mning	629	3189
Personal Care	586	3330
Real State	301	2707
Gambling	164	1355
Information	507	5684
Transportation	357	2471

Tabela 3.9 – Resumo dos resultados obtidos na categorização de 10 categorias

com a classe a que pertencia o *site* analisado, consolidando assim, o conjunto de treinamento.

Com a construção do conjunto de treinamento, o próximo passo foi parametrizar o conjunto de treinamento.

Como o processo de classificação é feito com base no método Naive Bayes, um método probabilístico cujo o evento é a palavra e o resultado é a classificação do *site* em uma classe, parametrizar consiste em calcular as probabilidades das palavras nas classes. Sendo essa etapa denominada de “Treinamento do Classificador”

Nesta segunda etapa, “Treinamento do Classificador”, o grande volume de palavras comuns às categorias, ainda existentes, não permitiam uma boa definição nos resultados da classificação, sendo então aplicada a técnica de extração das palavras que não contribuem no processo de classificação, denominada de *Average Mutual Information* (AMI).

A aplicação da técnica de redução com AMI e a eliminação de palavras com valores baixos, resultou em classes que não possuíam palavras com valores significativos em relação a mesma palavra em outras classes, resultando assim, em não haver palavras que contribuíssem para a classificação dessas classes.

O motivo encontrado para esse resultado foi a baixa quantidade de palavras únicas por categoria. Para solucionar esse problema, foi feita uma nova coleta de *sites*, aumentando a razão da quantidade de *sites* por categoria e diminuindo a quantidade de categoria de 10 para 4.

Os endereços dos *sites* utilizados para construção deste novo conjunto de treina-

mento foram obtidos da base estruturada sobre as referências. Um exemplo pode ser visualizado na tabela 3.6.

Das 63.805 referências a *sites*, foram selecionados 7.393 *sites* de quatro subcategorias (*Automotive, Computers, Health, Apparel*) para compor o conjunto de treinamento do classificador do elemento “Ramo de Atividade”. Essas quatro subcategorias correspondem às classes Automóvel, Computador, Saúde e Vestuário. Entretanto, só foi possível obter as páginas HTML de 945 (13%) dos 7.393 *sites*. Os principais problemas ocorridos estão relacionados a seguir:

1. Os endereços apontam para *sites* desativados;
2. Um grande volume de acessos ao *site* ou ao provedor desses *site*, podem ter provocado um “congestionamento”, no momento em que o aplicativo extrator de *site*, Wget, acessou;
3. O *site* possuir mecanismos que bloqueiam a cópia de seu conteúdo;

Na tabela 3.10 é apresentada a quantidade de *sites* efetivamente obtidos, por categoria e o nome dos arquivos que contém os endereços dos *sites* acessados e efetivamente obtidos (vide cdrom anexo). Apesar da quantidade de *sites* obtidos ser menor que a quantidade de *sites* acessados, a quantidade de *sites* obtidos por categoria pode ser considerada representativa (conforme pode ser visualizado na figura 3.8), ou seja, não há uma discrepância muito grande (mais especificamente, não há uma categoria com 1% dos *sites* e outra com 99%) entre as quantidades de *sites* obtidas por categoria.

Categoria	Quantidades		Arquivos de Sites	
	Seleção	Acesso	Acessados	Obtidos
<i>Automóvel</i>	1.495	333	WGETAutomovel	SitesAutomóvel.txt
<i>Computador</i>	2.287	178	WGETComputador	SitesComputador.txt
<i>Saúde</i>	1.976	179	WGETSaude	SitesSaude.txt
<i>Vestuário</i>	1.635	255	WGETVestuario	SitesVestuario.txt
Total	7.393	945		

Tabela 3.10 – Quantidade de *sites* obtidos por categoria para “Ramo de Atividade”

Aplicando o programa de remoção do conteúdo das marcações, em sua versão final representado pelo algoritmo 3.1, em que:

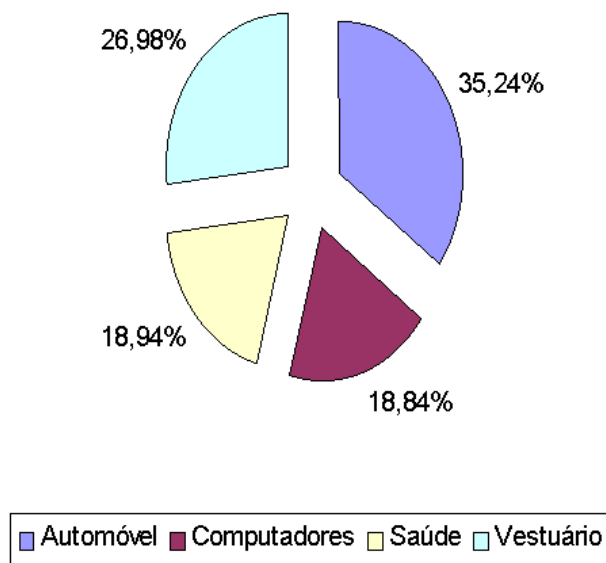


Figura 3.8 – Distribuição dos *sites* obtidos para “Ramo de Atividade”

- Nas linhas 1 a 6: estão relacionadas as declarações;
- Na linha “Retornar”: está descrita a informação de retorno;
- Nas linhas 7 a 9: são extraídos os códigos não HTML;
- Nas linhas 10 a 14: são extraídos os nomes das marcações, permanecendo os símbolos `<>` e `< / >`, com exceção às marcações “title”;
- Nas linhas 15 a 17: as letras codificadas (delimitadas por `&` e `;`), são traduzidas pelo seu equivalente;
- Nas linhas 18 a 27: são extraídas as palavras e contabilizadas. Vale ressaltar que, as palavras consideradas são armazenadas por *site* e categoria. As palavras a serem consideradas não constam na relação de StopWords (linha 19) e possuem tamanho maior que 1.

Aplicando um processo de contagem de palavras, sobre os 945 *sites* obtidos, foram identificadas 26.433 palavras únicas (diferentes), distribuídas de uma maneira equitativa por categoria, conforme pode ser visualizado na figura 3.9. Na tabela 3.11 é apresentada a quantidade de palavras únicas por categoria.

A quantidade total de palavras (incluindo as repetições) totalizou 147.106 palavras, cuja quantidade por categoria é apresentada na tabela 3.12. Uma análise

---

**Algoritmo 3.1** Remoção de marcações
 

---

**Declaração:** :

- 1:  $S$  {um conjunto de *site*}
  - 2:  $s$  {elemento de  $S$ }
  - 3:  $M$  {marcação}
  - 4:  $p$  {Palavra, qualquer conjunto de caracteres não delimitados pelo símbolo de marcação}
  - 5: *Sequencia* {Vetor= (nome da categoria, nome do site, palavra e quantidade de ocorrência)}
  - 6: *ArqSite* {Vetor de *Sequencia*}
- Retornar:** Uma imagem do *site* sem o nome das marcações e uma lista de quantidade de ocorrência das palavras
- 7: **Para todo**  $s$  de  $S$  **Faça**
  - 8:   Identificar e extrair todos os símbolos entre as marcações “<style”, “<object”, “<script” e “<applet”, inclusive as marcações.
  - 9: **Fim Para**
  - 10: **Para todo**  $M$  **Faça**
  - 11:   **Se**  $M_i \neq \text{“<title>”} \vee M_i \neq \text{“</title>”}$  **então**
  - 12:     retirar o nome da marcação, tendo como resultado <> ou < / >
  - 13:   **Fim Se**
  - 14: **Fim Para**
  - 15: **Para todo** conjunto de símbolos delimitados por & e ; **Faça**
  - 16:   Substituir o conjunto pelo caracter equivalente.
  - 17: **Fim Para**{Palavras associadas por – e ' devem ser consideradas como uma única palavra}
  - 18: **Para todo**  $p$  **Faça**
  - 19:   **Se** (o tamanho de  $p > 1$ )  $\wedge$  (StopWords não contém  $p$ ) **então**
  - 20:     Atribuir a *Sequencia*[1][2][3] a localização da palavra (categoria, site, palavra)
  - 21:     **Se** *ArqSite* $p$  não contém *Sequencia* **então**
  - 22:       *Sequencia*[4]  $\leftarrow$  1
  - 23:       *ArqSite*  $\leftarrow$  *Sequencia*
  - 24:     **senão**
  - 25:       Incrementar de 1 a *Sequencia*[1][2][3] em *ArqSite*
  - 26:     **Fim Se**
  - 27:   **Fim Se**
  - 28: **Fim Para**
- 

Categoria	Qtd. Palavras
Automóvel	7.468
Computador	5.887
Saúde	6.666
Vestuário	6.412
Total	26.433

Tabela 3.11 – Quantidade de palavras únicas por categoria para “Ramo de Atividade

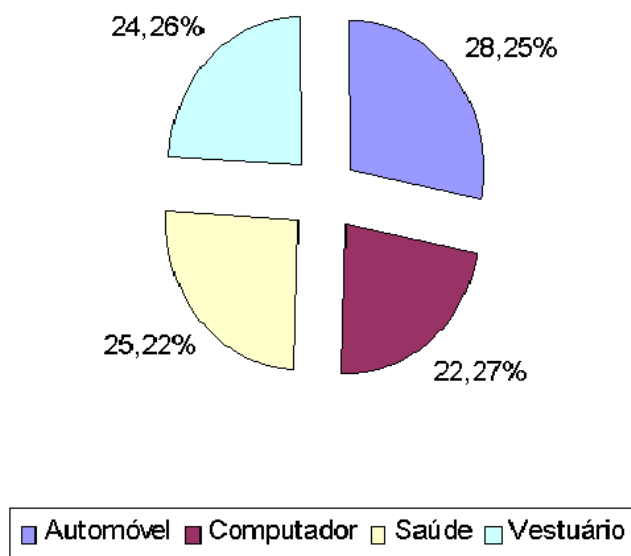


Figura 3.9 – Distribuição das palavras únicas por categoria para “Ramo de Atividade”

da figura 3.10 mostra que, também nesse caso, não há grandes discrepâncias na quantidade de distribuição entre as categorias das palavras obtidas.

<b>Categoria</b>	<b>Qtd. Palavras</b>
Automóvel	51.711
Computador	31.036
Saúde	34.687
Vestuário	29.672
<b>Total</b>	<b>147.106</b>

Tabela 3.12 – Distribuição da quantidade total das palavras do conjunto de treinamento

Os dados apresentados mostram que, apesar do conjunto de treinamento ter sido composto por uma fração reduzida do conjunto de *sites* enumerados no Portal Yahoo, foi possível obter quantidades de dados representativas e, de certa forma, eqüitativas para as sub-categorias escolhidas (automóvel, computador, saúde e vestuário).

Tendo sido definido o conjunto de treinamento, o próximo passo constitui-se na parametrização desse conjunto, por meio da etapa de “Treinamento do Classificador”.

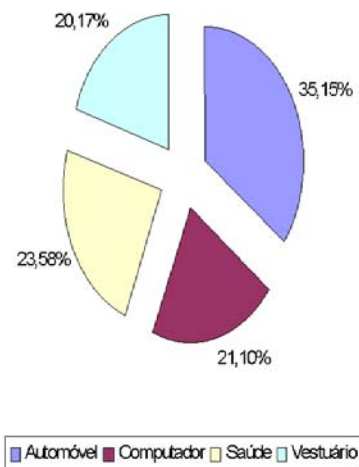


Figura 3.10 – Percentual de distribuição do total das palavras do conjunto de treinamento

### 3.2.6 Treinamento do Classificador

O objetivo nesta etapa é caracterizar as classes da categoria por meio de palavras. Com o objetivo de selecionar as palavras com maior capacidade de contribuir para a discriminação das classes, foi calculada a Informação Mútua Média (*Average Mutual Information - AMI*) de cada palavra (vide equação 2.5). Na tabela 3.13 são apresentados alguns valores de AMI que foram calculados para as 26.433 palavras únicas nas quatro classes, em que a coluna de cada categoria contém o valor de AMI calculado e entre parênteses a quantidade de ocorrência da palavra na categoria.

Uma relação completa das palavras por categoria e os valores de AMI atribuídos, podem ser visualizados no arquivo “ValoresAMIRamo.pl”, no cdrom anexo.

Palav.	Automóvel	Computador	Saúde	Vestuário
car	-0,0115000 ( 1046)	-	-0,0012900 ( 31)	
auto	-0,0056700 ( 511)	-0,0003370 ( 7)	-0,0003440 ( 7)	-0,000334 ( 7)
buttons	-0,0001860 ( 9)	-0,0001340 ( 8)	-0,0000383 ( 1)	-0,000124 ( 7)
protect	-0,0002880 ( 15)	-0,0001250 ( 5)	-0,0001720 ( 8)	-0,000160 ( 8)
server	-0,0002250 ( 4)	-0,0025000 (366)	-0,0011700 ( 39)	-0,000243 ( 5)
mind	-0,0001740 ( 8)	-0,0000793 ( 3)	-0,0001240 ( 6)	-0,000131 ( 8)

Tabela 3.13 – Valores de AMI por palavra por categoria

Na tabela 3.14 pode ser visualizada a quantidade de palavras por valores de AMI. Por exemplo, existem 883 palavras com valor de Informação Mútua Média na categoria “Automóvel” ( $-0,000075 < \text{valorAMI} \leq -0,0001$ ). Não há palavras com valor de AMI superior a  $-0,0075$  na categoria “Vestuário”. Quanto maior o valor da informação mútua média (em módulo), mais discriminante é a palavra. Por exemplo, a única palavra com valor AMI superior a  $-0,01$  é “car”. As palavras com valores próximos de  $-0,000075$  não contribuem efetivamente para o processo de discriminação das classes.

Na figura 3.11 é apresentada uma visão gráfica dos valores da tabela 3.14. A curva “Todas as Categorias” representa a quantidade total de palavras, sem considerar a separação por classes.

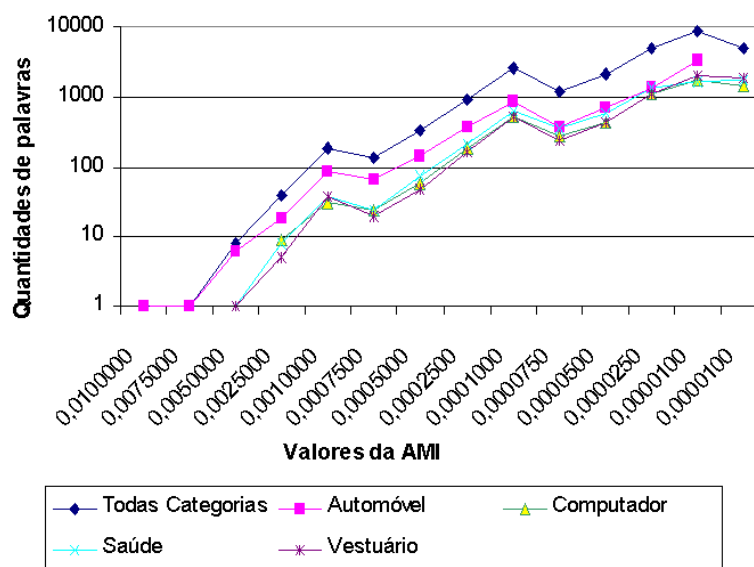


Figura 3.11 – Distribuição das palavras no conjunto de treinamento por faixa de Informação Mútua Média (AMI)

Um conjunto de teste foi selecionado, utilizando o mesmo procedimento na seleção do conjunto de treinamento, porém formado com os *sites* diferentes do conjunto de treinamento. A distribuição dos endereços de *sites* por categoria, pode ser visualizada na tabela 3.15. Vale ressaltar que, na nova seleção de *sites* não foi possível obter mais do que 12 endereços de *sites* para a categoria “Saúde”.

Esses 72 *sites* foram aplicados a 7 classificadores que foram construídos de acordo com os valores de AMI apresentados na tabela 3.14. O classificador 4 foi construído utilizando-se palavras cujos valores de AMI são maiores ou iguais (em módulo) que  $-0,0025$  e o classificador 10 foi construído utilizando-se palavras

Faixa da AMI	Automóvel	Computador	Saúde	Vestuário
-0,01	1	0	0	0
-0,0075	1	0	0	0
-0,0050	6	0	1	1
-0,0025	18	9	8	5
-0,001	31	37	36	36
-0,00075	7	23	19	19
-0,000500	145	57	74	49
-0,000250	366	181	215	161
-0,000100	883	519	629	525
-0,000075	377	268	344	239

Tabela 3.14 – Distribuição das palavras por faixa de valores da AMI

Categoria	Quant. sites
Automóvel	20
Computador	20
Saúde	12
Vestuário	20
Total	72

Tabela 3.15 – Distribuição dos *sites* selecionados para o conjunto de teste

com valores de AMI maiores ou iguais que  $-0,000075$ . Seqüências de palavras com valores exclusivamente acima de  $-0,0025$  não foram consideradas pois, de acordo com a tabela 3.14, não haveria palavras para discriminar as classes “Computador”, “Saúde” e “Vestuário”.

Na tabela 3.16 e, graficamente na figura 3.12, podem ser visualizados os resultados obtidos com a aplicação do classificador *Naive Bayes*. Com relação aos resultados, Vale ressaltar que, os valores se encontram em uma faixa estreita (30%) de percentual de acertos, que tende a diminuir quando são acrescentadas mais palavras (vide tabela 3.14). Exceção se faz ao valor de 100% para a classe Automóvel, porque o classificador sempre fornecerá um resultado, sendo assim, quando não possui informações suficientes para identificar uma classe, o classificador elege a classe mais provável.

Na tabela 3.17 pode ser visualizado o resultado do processo de classificação dos 72 *sites* utilizando o classificador do Minerador Web.

Analisando a figura 3.13, nota-se que há uma estabilização do índice de acertos a partir do classificador 4 (treinado com palavras com valores de AMI acima de

Classificador	Faixa da AMI	Automóvel	Computador	Saúde	Vestuário
1	-0,002500	20	2	1	2
2	-0,001000	7	9	2	6
3	-0,000750	6	6	3	2
4	-0,000500	6	1	0	3
5	-0,000250	6	1	0	1
6	-0,000100	3	0	1	1
7	-0,000075	4	2	1	1
8	-0,000050	4	2	0	1
9	-0,000025	5	3	1	1
10	-0,000010	5	1	1	0

Tabela 3.16 – Quantidade de acertos do classificador *Naive Bayes*

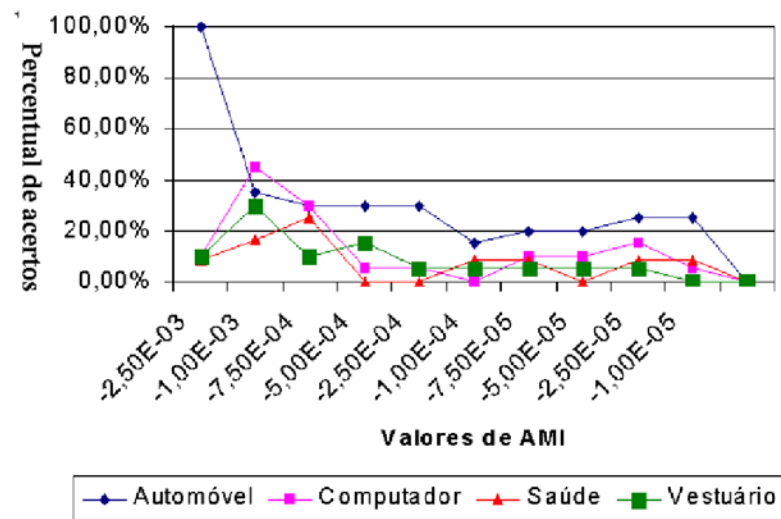


Figura 3.12 – Distribuição de acertos por classificador *Naive Bayes*

Classificador	Faixa da AMI	Automóvel	Computador	Saúde	Vestuário
1	-0,002500	15	4	4	8
2	-0,001000	17	11	4	15
3	-0,000750	15	12	4	16
4	-0,000500	15	11	6	18
5	-0,000250	15	14	6	17
6	-0,000100	14	15	5	18
7	-0,000075	14	16	5	17
8	-0,000050	14	16	7	17
9	-0,000025	14	16	7	17
10	-0,000010	13	16	7	17

Tabela 3.17 – Quantidade de acertos do classificador Minerador Web

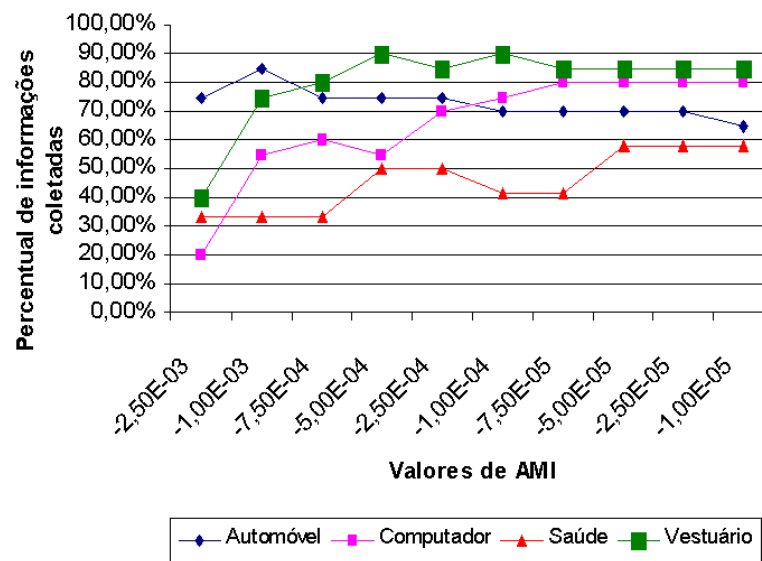


Figura 3.13 – Distribuição de acertos por classificador Minerador Web

-0,0005). Assim, foi selecionado o classificador 4 para os demais testes realizados para a classificação dos *sites* em relação a “Ramo de Atividade”. A escolha desse valor, como limite inferior da faixa de AMI, reduz consideravelmente a quantidade de palavras no conjunto de treinamento, como pode ser visto na tabela 3.18. A relação dessas palavras pode ser visualizada no arquivo “CategoriaRamo.pl”.

<b>Categoria</b>	<b>Qtd. palavras antes da seleção</b>	<b>Qtd. palavras após a seleção</b>
Automóvel	7.468	2.680
Computador	5.887	1.524
Saúde	6.666	1.912
Vestuário	6.412	1.466

Tabela 3.18 – Comparação entre as quantidades de palavras no conjunto de treinamento, para a categoria “Ramo de Atividade”, antes e após a seleção da faixa da AMI

O processo de classificação de elementos implícitos podem ser visualizados por meio de 3 algoritmos. O algoritmo 3.3 contém a descrição da estrutura de dados utilizada no algoritmo 3.4 que efetua a “carga da ontologia” no Minerador Web, que precede o algoritmo 3.2 no qual está descrito o processo do classificador Minerador Web para identificação de uma classe em uma categoria.

A extração da “Atividade”, também uma categoria implícita, foi tratada de forma semelhante ao processo de extração da categoria “Ramo de Atividade”, ou seja, para a identificação do elemento “Atividade” foi utilizado o produto resultante da aplicação do mesmo conjunto de algoritmos de “Busca de Palavras” que podem ser visualizados na algoritmo 3.1. Porém, os *sites* do conjunto de treinamento foram reagrupados em categorias de “Atividades”.

O agrupamento dos *sites* na categoria “Ramo de Atividade” foi feita com base na divisão de categorias fornecida pelo portal Yahoo. Mas não existe uma divisão de classes, explícita, para a categoria “Atividade”, sendo necessário analisar os *sites* e classificá-los na categoria “Atividade”.

O conjunto de informações a serem analisadas para obter as classes e os agrupamentos foram obtidos na base de dados estruturada de referências dos *sites* (arquivo “ReferenciasEnderecos”).

Em um primeiro momento, foram selecionadas as mesmas referências utilizadas (7.393 referências a *sites*) para construir o classificador da categoria “Ramo de

---

**Algoritmo 3.2** Classificação de um *site* em uma classe de uma categoria
 

---

**Declaração:** :

*vPalavra* {vetor tridimensional preenchido em algoritmo 3.4}

*vDoc* {vetor tridimensional, contendo:

categoria.
palavra.
quantidade de ocorrência da palavra.

}

*PosDoc* {ponteiro que aponta uma posição de *vDoc*.}

*PrWiD* {probabilidade da i-ésima palavra no documento.}

*PrWiC* {probabilidade da i-ésima palavra no conjunto de treinamento.}

*n* {quantidade de palavras no documento.}

*QtdCat* {quantidade de categorias, atribuída no algoritmo 3.4}

*T* {variável do tipo inteiro, atribuído em algoritmo 3.4.}

*vCT* {vetor tridimensional, contendo a palavras do conjunto de treinamento, por categoria.

Dim	Conteúdo
1	Nome da Categoria
2	Palavra
3	Frequência de ocorrência da palavra na categoria

}

**Para**  $i = 0$  a quantidade de elementos em *vPalavra* **Faça**
**Se** *vPalavra*[1], ocorreu em *vDoc* **então**
*PosDoc*  $\leftarrow$  posição de *vPalavra*[1] em *vDoc*
*PrWiD*  $\leftarrow \frac{vDoc[2]}{n}$ 
**Para**  $k$ , de 0 a *QtdCat* **Faça**
**Se** *vPalavra*[ $i$ ] ocorreu em *vCT* **então**
*iPosPalavra*  $\leftarrow$  posição em *vCT* de *vPalavra*
**Se** *vCT*[*iPosPalavra*] = *vClasse*[ $k$ ] **então**
*PrWiC*  $\leftarrow \text{frac}vCT[iPosPalavra + 1]vClasse[k + 1] + vClasse[k + 2]$ .

**senão**
*PrWiC*  $\leftarrow \text{frac}TvClasse[k + 1] + vClasse[k + 2] * \frac{1}{T - vClasse[k + 1]}$ .

**Fim Se**
*vClasse*[ $k + 3$ ]  $\leftarrow vClasse[k + 3] + PrWiD * \log_2 \left( \frac{PrWiC}{PrWiD} \right)$ .

**Fim Se**
 $k++$ 
**Fim Para**
**Fim Se**
**Fim Para**


---

---

**Algoritmo 3.3** Carga da “ontologia” de uma categoria - Declarações
 

---

**Declaração:** :

*ArqCat* {descriptor do arquivo de categoria, em que os registros possuem o seguinte formato:

```

data(td(categoria, palavra, quantidade)).
qtdSiteCat(categoria, quantidade).
qtdCat(quantidade).

```

}

*sApoio* {variável do tipo string para auxiliar na carga dos registros de *ArqCat*}

*sCategoria* {variável do tipo string para auxiliar na identificação da categoria}

*sPalavra* {variável do tipo string para auxiliar na identificação de palavra}

*iQtd* {variável do tipo string, para auxiliar na identificação da frequência de ocorrência de *sPalavra* em *sCategoria*.}

*vClasse*[1..n][1-5] {vetor pentadimensional, sendo:

Dim	Conteúdo
1	Nome da categoria
2	quantidade de site
3	quantidade de palavras na categoria (Tc)
4	Somatória das quantidades de todas as palavras na categoria
5	Valor totalizado para a classe <i>r</i>

}

*T* {variável do tipo inteiro, correspondendo à quantidade de palavras únicas do conjunto de treinamento.}

*vData* {vetor tridimensional, contendo a palavras do conjunto de treinamento, por categoria.

Dim	Conteúdo
1	Nome da Categoria
2	Palavra
3	Frequência de ocorrência da palavra na categoria

}

---

**Algoritmo 3.4** Carga da “ontologia” de uma categoria
 

---

$sApoio \leftarrow$  registro de  $ArqCat$ .

**Enquanto** houver registros em  $ArqCcat$  **Faça**

**Se** Se o tamanho de  $sApoio > 0$  e  $sApoio$  não for comentário (‘%’) **então**

**Se** Se existe o trecho “data(” em  $sApoio$ ) **então**

$sCategoria \leftarrow$  trecho de  $sApoio$  que represente a categoria

$sPalavra \leftarrow$  trecho de  $sApoio$  que represente a palavra.

$iQtd \leftarrow$  trecho de  $sApoio$  que contenha a frequência de ocorrência de  $sPalavra$

$PosClasse \leftarrow$  posição em  $vClasse[x, 1] = sCategoria$ .

Incrementar  $vClasse[x, 3]$ .

Incrementar  $vClasse[x, 4]$  de  $iQtd$

**Se**  $sPalavra$ , não existe em  $vData$  **então**

$vData[1] \leftarrow sCategoria$

$vData[2] \leftarrow sPalavra$

$vData[3] \leftarrow iQtd$

**Fim Se**

**senão** **Se** existe o trecho “qtdSiteCat” em  $sApoio$  **então**

$vClasse[1] \leftarrow$  trecho de  $sApoio$  que represente a categoria

$vClasse[2] \leftarrow$  trecho de  $sApoio$  que represente a quantidade de sites

$vClasse[3] \leftarrow 0$

$vClasse[4] \leftarrow 0$

$vClasse[5] \leftarrow 0$

Incrementar  $QtdSites$  de 1.

**senão** **Se** existe o trecho “qtdCat(” em  $sApoio$ ) **então**

$QtdCat \leftarrow$  trecho de  $sApoio$  que represente a quantidade de categorias.

**Fim Se**

**Fim Se**

$sApoio \leftarrow$  registro de  $ArqCat$ .

**Fim Enquanto**

**Para**  $i$  de 0 a quantidade de classes  $r$  **Faça**

$vClasse[5 + i] \leftarrow \frac{vClasses[2+i]}{QtdSites}$

**Fim Para**

---

Atividade”. Sobre essa relação de referências, foi feita uma análise visual no campo 3 da estrutura de referência (apresentada na tabela 3.6), com o objetivo de identificar substantivos que fizessem uma analogia com o significado de “Atividade” (sinonímia), resultando em 1.717 referências e 4 categorias: “Fabricante”, “Prestadora de Serviços”, “Vendas no Atacado” e “Vendas no Varejo”. Esses endereços constam no arquivo “EnderecosAtividades.txt”.

O critério de classificação dos *sites*, aplicado sobre essas categorias utilizou as seguintes regras:

- As empresas que efetuam vendas e as que estão explicitamente identificadas como venda no atacado, foram classificadas na atividade “Vendas no Atacado”;
- As empresas que efetuam compra e venda, facilitam pagamentos e/ou possuem uma diversidade de produtos, ou possuem mensagens direcionadas ao consumidor varejista, como por exemplo “desconto em compras acima de X unidades”, ou ainda, explicitar a forma de pagamento como, “compre pelo cartão”, foram classificadas na atividade “Vendas no Varejo”;
- As empresas que prestam consultoria, reparos ou manutenção, entregas, entre outras palavras relacionadas a serviços, além de explicitamente o conjunto de palavras “nossos serviços”, foram classificadas como “Prestadoras de Serviços”;
- As empresas que continham explicitamente a palavra fabricação, desenvolvimento, produto personalizado, foram classificadas na atividade “Fabricante”.

A distribuição em quantidades de *sites* por categoria, pode ser visualizada na tabela 3.19, e pode ser notado um valor discrepante para a categoria “Venda no Atacado”, que como mostra o gráfico de distribuição percentual é de 2%, aproximadamente 40 vezes menor que a categoria “Venda no Varejo”, resultando em uma distribuição não eqüitativa dos valores.

Aplicando o processo de contagem de palavras sobre esses 1.717 *sites*, foram identificadas 21.354 palavras únicas, distribuídas no conjunto de categorias “Atividade”, e essa distribuição pode ser visualizada na tabela 3.20 e no arquivo “PalavrasAtividades.pl”.

<b>Categoria</b>	<b>Qtd. identificada</b>
Fabricante	127
Serviços	286
Venda no atacado	34
Venda no varejo	1.270
Total	1.717

Tabela 3.19 – Quantidade de *sites* por categoria para “Atividade

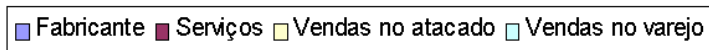
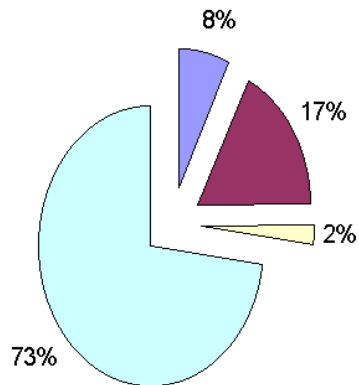


Figura 3.14 – Distribuição percentual dos *sites* por categoria para “Atividade”

Na figura 3.15 pode ser visto que na distribuição percentual das palavras únicas, a categoria “Venda no Atacado” é menor pelo motivo de ter sido coletada uma amostra com poucos *sites*.

<b>Categoria</b>	<b>Qtd. Palavras</b>
Fabricante	3.322
Serviços	7.317
Venda no atacado	1.311
Venda no varejo	18.238

Tabela 3.20 – Quantidade de palavras únicas por categoria para “Atividade”

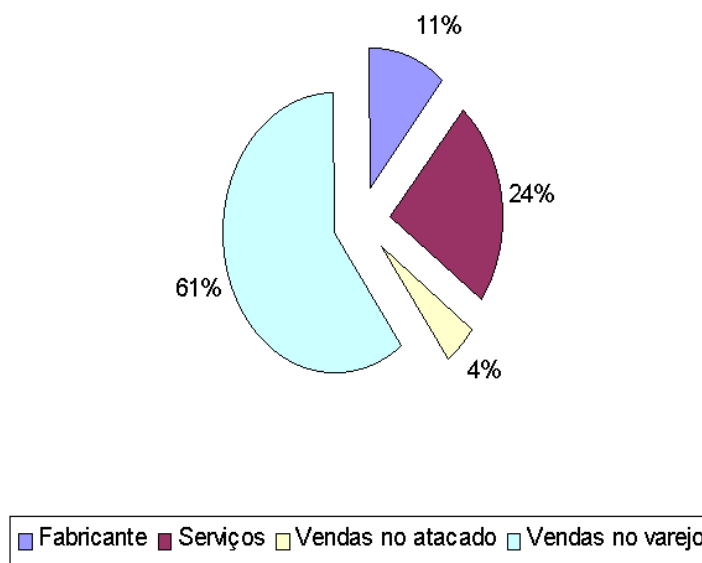


Figura 3.15 – Percentual das palavras únicas distribuídas na categoria “Atividade”

A quantidade total de palavras (incluindo as repetições) totalizaram 197.263 palavras, cuja quantidade por categoria é apresentada na tabela 3.21. Uma análise da figura 3.16 mostra que a categoria “Vendas no Varejo” continua inexpressiva.

Com o objetivo de selecionar as palavras com maior capacidade de contribuir para a discriminação das classes, foi calculada a Informação Mútua Média (Average Mutual Information - AMI) de cada palavra (equação 2.5). Na tabela 3.22 são apresentados alguns valores AMI que foram calculados para as 21.354 palavras únicas, nas quatro classes, em que a coluna de cada categoria contém o valor de AMI calculado e entre parênteses a quantidade de ocorrência da palavra na categoria.



Figura 3.16 – Percentual de distribuição da quantidade total das palavras do conjunto de treinamento “Atividade”

Categoria	Qtd. Palavras
Fabricante	9.824
Serviços	40.221
Venda no atacado	3.029
Venda no varejo	144.189
Total	197.263

Tabela 3.21 – Distribuição da quantidade total das palavras do conjunto de treinamento “Atividade”

Palavras	Fabricante	Serviços	Venda no atacado	Venda no varejo
manufactured	-0,0000180 ( 2)	-0,0000489 ( 3)	-0,0000009 (2)	-0,00014810 ( 8)
repair	-0,0000707 ( 4)	-0,0002393 ( 12)	-0,0000192 (1)	-0,00125568 (82)
scooter	0,0000097 (10)	-0,0000637 ( 4)	não há valor	-0,0001182 ( 5)
services	-0,0001473 ( 8)	-0,0010444 (105)	-0,0000812 (9)	-0,0019268 (97)

Tabela 3.22 – Valores de AMI por palavra por categoria

Analisando a tabela 3.22, nota-se que a faixa de valores da AMI possui valores positivos (palavra *scooter*, coluna “Fabricante”, valor de AMI = 0,0000097), e resultados negativos no valor da AMI, da ordem de  $10^{-7}$  (palavra *manufacture* coluna “Venda no Atacado”, valor de AMI = -0,0000009) ao contrário da categoria “Ramo de Atividade” em que a faixa toda era negativa e possuía os valores da AMI entre  $10^{-2}$  e  $10^{-6}$ .

Na tabela 3.23 pode ser visualizada a quantidade de palavras por valores da AMI. Nota-se que não existe valores da AMI superiores a 0,000135 e nem inferiores a -0,000681, para “Vendas no Atacado”. O mesmo acontecendo para “Fabricante” entre os valores maiores ou igual a -0,001225 e menores que -0,001089. Esse quadro é reflexo dos valores discrepantes existentes nas quantidades de *sites* (tabela 3.19), que se propagou em termos quantitativos para a quantidade de palavras únicas tabela 3.20.

Na figura 3.17 é apresentada uma visão gráfica dos valores da tabela 3.23. Uma relação completa das palavras por categoria e os valores de AMI atribuídos, podem ser visualizados no arquivo “ValoresAMIAtividade.pl”.

<b>Faixa</b>	<b>Fabricante</b>	<b>Serviços</b>	<b>Vendas no atacado</b>	<b>Vendas no varejo</b>
0,000135	1	0	0	0
-0,000001	925	0	430	0
-0,000137	2278	6563	862	15313
-0,000273	73	427	15	1309
-0,000409	25	140	2	546
-0,000545	7	75	2	317
-0,000681	3	38	0	157
-0,000817	4	20	0	129
-0,000953	3	9	0	79
-0,001089	2	9	0	63
-0,001225	0	7	0	46
<-0,001225	2	28	0	279

Tabela 3.23 – Distribuição das palavras por faixa de valores da AMI para o novo agrupamento de “Atividade”

Analisando o gráfico, foi selecionada a faixa de valores de AMI, mais alta e que contivesse todas as categorias do conjunto “Atividade”.

Porém a utilização do conjunto de treinamento de “Atividade” no conjunto de teste idêntico ao utilizado para “Ramo de Atividade” da tabela 3.15, resultou em aproximadamente 10% de classificação correta. Como o classificador sempre

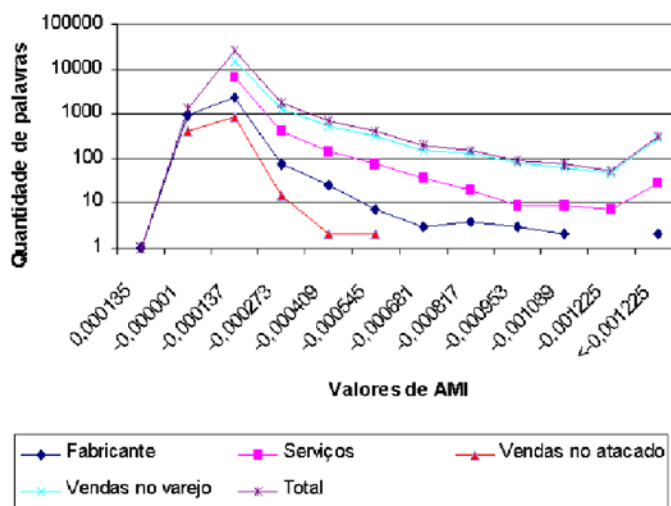


Figura 3.17 – Distribuição das palavras no conjunto de treinamento por faixa de Informação Mútua Média (AMI)

fornece uma categoria como resultado, a categoria mais classificada foi “Vendas no Atacado”.

A título de experimento, a solução adotada foi eliminar a categoria que possuía um valor baixo de quantidade total de palavras e quantidade de *sites*. Como resultado da solução foi criada a categoria “Vendas” que agrupa a categoria de “Venda no Varejo” e a categoria de “Venda no Atacado”.

A composição do novo agrupamento pode ser visualizada na tabela 3.24 em que a categoria “Vendas” agrupa as quantidades referente às duas outras categorias. A quantidade total de palavras únicas permaneceu em 21.354 palavras.

Categoria	Sites	Quantidades	
		Palavras Únicas	Palavras
Fabricante	127	3.322	9.824
Serviços	286	7.317	40.221
Venda	1.304	18.414	147.218
Total	1.717	-	197.263

Tabela 3.24 – Composição quantitativa de “Atividade” com referência ao novo agrupamento de categorias

Na tabela 3.25 são apresentados alguns valores de AMI que foram calculados para a nova composição da categoria “Atividades”, contendo as mesmas palavras da

tabela 3.21.

Palavras	Fabricante	Serviços	Vendas
manufactured	-0,00001801 ( 2)	-0,0000489 ( 3)	-0,0001704 (10)
repair	-0,00007075 ( 4)	-0,0002393 ( 12)	-0,0012765 (83)
scooter	0,00000979 (10)	-0,0000637 ( 4)	-0,0001190 ( 5)
services	-0,00014736 ( 8)	-0,0010444 (105)	-0,0020538 (97)

Tabela 3.25 – Valores de AMI por palavra por categoria, com o novo agrupamento de “Atividade”

Com relação aos valores de AMI visualizados na tabela 3.22 (composição de 4 categorias) e na tabela 3.25 (Composição de 3 categorias), houve um acréscimo na coluna “Vendas” em relação à “Vendas no Varejo”, da ordem de  $10^{-5}$ , na coluna “Fabricante” houve um acréscimo da ordem de  $10^{-8}$ , enquanto que para “Serviços” as alterações foram bem menores, não sendo significativas.

Na tabela 3.26 pode ser visualizada a quantidade de palavras por valores de AMI. Esses valores constam no arquivo de nome “ValoresAMIAtividade3.pl”.

O resultado apresentado na tabela, continua com valores extremos muito distantes (limite superior= 0,0000926 e inferior=  $-0,0000006$ ). Para obter uma visão melhor dessa distribuição, a relação de valores da AMI foi distribuída em 20 faixas de valores. Nota-se que, mesmo dispersando os valores com o aumento da faixa de valores AMI, não existem valores nulos, para os valores baixos da AMI. Conclue-se que mesmo não havendo uma distribuição eqüitativa das quantidades das categorias, essas quantidades estão bem distribuídas, demonstrando que não existe uma quantidade significativa isolada em uma determinada faixa da AMI.

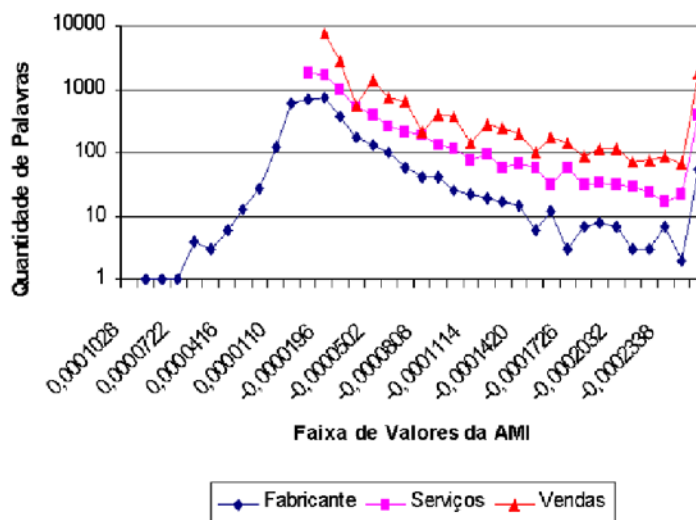
Na figura 3.18 é apresentada uma visão gráfica dos valores da tabela 3.26.

Foi considerado o limite inferior da AMI, a ser aplicado no conjunto de treinamento da “Atividade”, como sendo  $-0,0000196$ , por possuir o maior valor de AMI que abrange todas as categorias.

A identificação da faixa de valores da AMI a ser utilizada no classificador, finaliza o processo que auxiliará identificação de informações implícitas.

<b>Faixa</b>	<b>Fabricante</b>	<b>Serviços</b>	<b>Vendas</b>
0,0001028	0	0	0
0,0000926	1	0	0
0,0000824	1	0	0
0,0000722	1	0	0
0,0000620	4	0	0
0,0000518	3	0	0
0,0000416	6	0	0
0,0000314	13	0	0
0,0000212	28	0	0
0,0000110	123	0	0
0,0000008	582	0	0
-0,0000094	701	1821	0
-0,0000196	735	1710	7762
-0,0000298	363	954	2683
-0,0000400	178	508	542
-0,0000502	135	402	1362
-0,0000604	97	256	722
-0,0000706	58	220	644
-0,0000808	40	181	215
-0,0000910	41	134	402
-0,0001012	25	116	370
-0,0001114	22	75	144
-0,0001216	19	95	288
-0,0001318	17	58	241
-0,0001420	15	64	203
-0,0001522	6	57	97
-0,0001624	12	31	177
-0,0001726	3	56	143
-0,0001828	7	31	87
-0,0001930	8	33	113
-0,0002032	7	32	111
-0,0002134	3	29	73
-0,0002236	3	24	75
-0,0002338	7	17	89
-0,0002440	2	22	65
< -0,0002440	54	390	1806

Tabela 3.26 – Distribuição das quantidades de palavras no conjunto de treinamento, para a categoria “Atividade”, após a aplicação da AMI



< /SCRIPT> (linha 16). Esse código contém a função “getResults()” (linha 6), que verifica se o usuário digitou alguma condição para pesquisa de informação (linha 21) e que apresenta uma caixa de diálogo fornecendo o aviso: “*You must supply some search criteria*” (linha 8) caso uma condição não tenha sido fornecida. Note que a função “getResults()” possui uma referência de chamada na linha 22 que será executada quando for selecionado o botão que possui o valor “Go”;

- A segunda premissa, compreende as seguintes características, *sites* com pouco texto útil e pouco código HTML. Esses *sites*, em contrapartida, possuem muitos *links* além de muito código não HTML. No caso de *sites* com essas características serem analisados, e não possibilitarem a identificação do elemento “Nome da Empresa”, esses *sites* não são considerados para análise, sendo descartados.

```

1 <HTML>
2 <HEAD>
3 <TITLE>eTirePlace.com - Your complete online source for
  tires.</TITLE>
4 <META content="text/html; charset=windows-1252" http-
  equiv=Content-Type>
5 <SCRIPT language=Javascript>
6 <!-- function getResults(){
7   if (" == document.forms.quick_search.quick_text.value) {
8     alert('You must supply some search criteria');
9     document.quick_search.quick_text.focus();
10    return false;
11   }
12   document.forms.quick_search.submit();
13   return true;
14  }
15  //->
16 </SCRIPT>
17 <STYLE type=text/css>
18 ...
19 Enter your city and state, or zip code:<BR>
20 i.e. (Cambridge, MA or 02139)</FONT><BR>
21 <INPUT name=quick_text &nbsp;
22 <INPUT onclick="return getResults();" type=button
  value=Go><BR>
23 ...

```

Figura 3.19 – Código JavaScript em uma página HTML

A seguir serão relacionados os processos de extração da informação, para cada elemento da ontologia de empresas, com o objetivo de preencher uma aplicação XML. Na figura 3.20 pode ser visualizada uma aplicação XML, preenchida com as informações extraídas de *sites* escritos em HTML.

### 3.3.1 Extração do elemento “*Site*”

O elemento *site* corresponde simplesmente ao endereço fornecido inicialmente para a extração das informações do *site*.

Por exemplo, para a busca de informações no *site* “www.pacifictires.com” a aplicação XML resultante toma a seguinte forma:

```
<site>www.pacifictires.com</site>
```

### 3.3.2 Extração do elemento “Nome da Empresa”

Apesar do nome da empresa geralmente estar explícito no *site* e ser de fácil identificação por um processo visual, a extração dessa informação não é tão simples quando feita com o auxílio programas. Para o Minerador Web foi criada heurísticas para identificar padrões de localização de nomes das empresas.

Durante a análise visual efetuada na etapa de categorização, no processo de criação da meta informação, foram feitas as seguintes constatações em relação aos nomes das empresas:

- O nome da empresa pode conter um termo que identifica o tipo de sociedade comercial;
- O nome da empresa pode aparecer entre as marcações “<title>” e “</title>”;

Com base nessas constatações, foram desenvolvidos três algoritmos. O primeiro, algoritmo 3.5, busca o elemento “Nome da Empresa” por termos.

O segundo, algoritmo 3.6, busca o elemento “Nome da Empresa” entre as marcações “<title>” e “</title>”.

Os resultados de ambos os algoritmos são aplicados a um terceiro, o algoritmo 3.8, que contabiliza a frequência de ocorrência das palavras que compõem os supostos nomes das empresas.

```

<?xml version="1.0"?>
<!DOCTYPE empresa SYSTEM "Empresas.dtd" >
<?xml-stylesheet type="text/xsl" href="Empresas.xsl"?>
<empresa>
  <nomeEmpresa>pacific tire sales, inc</nomeEmpresa>
  <site>pacific.cpct</site>
  <ramo>
    <nomeRamo>'Automóvel'</nomeRamo>
    <nomeAtividade>Fabricante</nomeAtividade>
  </ramo>
  <contato>
    <endereco>
      <uf>CA</uf>
      <zpt>92655</zpt>
      <cidade>Midway City</cidade>
      <complemento></complemento>
      <via>8249 Bolsa Ave</via>
    </endereco>
    <comunicacao>
      <telefone>714.892.2093</telefone>
      <fax></fax>
      <email></email>
    </comunicacao>
  </contato>
  <produto>
    <nomeProduto>tire</nomeProduto>
    <especializacao>155R12</especializacao>
    <caracteristicas></caracteristicas>
    <preco>24.00</preco>
  </produto>
  <produto>
    <nomeProduto>tire</nomeProduto>
    <especializacao>155R12</especializacao>
    <caracteristicas>PASSENGER / PERFORMANCE TIRES ME-
    TRIC P-METRIC 155R</caracteristicas>
    <preco>18.00</preco>
  </produto>
</empresa>

```

Figura 3.20 – Aplicação XML, resultado da extração de informações HTML

---

**Algoritmo 3.5** Extração do elemento “Nome da Empresa”, por termos

---

**Declaração:** :

$s$  {Site a ser analisado}

$Termos$  {Conjunto de terminações referente ao tipo de sociedade comercial}

$Trecho$  {Vetor que contém o nome da empresa}

**Retornar:** Frases que possivelmente conterá os nomes das empresas

**Para todo**  $Termos$  **Faça**

**Para** cada ocorrência de  $Termos$  encontrada em  $s$  **Faça**

Identificar a primeira posição de uma marcação de início (<>) anterior ao termo

$vTrecho \leftarrow$  o conjunto de caracteres entre a posição identificada e a posição do

$Termo$ , inclusive

**Fim Para**

**Fim Para**

---



---

**Algoritmo 3.6** Extração do elemento “Nome da Empresa”, pela marcação “<title >”

---

**Declaração:** :

$s$  {Site a ser analisado}

$Trecho$  {Vetor que contém o nome da empresa}

**Retornar:** complementa  $Trecho$  do algoritmo 3.5

**Enquanto** existir a ocorrência do par de marcações <title > e </title > **Faça**

$vTrecho \leftarrow$  o conjunto de caracteres entre a marcação de início <title > e a marcação de fim </title >, sendo essas marcações eliminadas

**Fim Enquanto**

---



---

**Algoritmo 3.7** Extração do elemento “Nome da Empresa”, pela quantidade de ocorrência do nome”

---

**Declaração:** :

$s$  {Site a ser analisado}

$Trecho$  {Vetor que contém o nome da empresa}

$vEmpresa$  {Vetor que contém o resultado da análise de cada  $Trecho$ }

**Retornar:**  $vEmpresa$  terá a frequência de ocorrência de cada sentença de  $Trecho$

**Para** cada  $Trecho_n$  **Faça**

**Enquanto** existir mais de 2 palavras no  $Trecho$  analisado **Faça**

Contabilizar a frequência de ocorrência de  $Trecho$  em  $s$

$vEmpresa \leftarrow$  (Trecho e quantidade de ocorrência)

A cada iteração retirar uma palavra à esquerda de  $Trecho$

**Fim Enquanto**

**Fim Para**{Repetir o processo porém a cada iteração retirar uma palavra à direita de  $Trecho$ }

---

Esse último algoritmo foi desenvolvido com base nas seguintes premissas:

- O nome da empresa se encontra nas extremidades dos trechos selecionados. Esse fato pode ser constatado no algoritmo de busca por termos (algoritmo 3.5), em que são selecionados trechos terminados com termos definidos;
- No caso de busca nas marcações <title>, assume-se que o nome das empresas encontram-se ou no início ou no fim do trecho;
- Os *sites* se referem a atividades comerciais e, conseqüentemente existe o interesse em mencionar o nome da empresa em diversas partes do *site*.

Assim, para cada termo identificado como suposto nome da empresa, foram calculadas as freqüências de ocorrências de componentes dos supostos nomes das empresas. Esse processo constitui-se na retirada das palavras, à esquerda e à direita, contabilizar a freqüência de ocorrência das palavras restantes e acumular no suposto nome da empresa.

Para diminuir a inconsistência nos resultados do algoritmo 3.8, aos valores de ocorrências encontrados foram incrementadas a freqüência de ocorrência das palavras individuais de cada suposto nome da empresa. Essa heurística é justificada pelo fato que as empresas repetem o nome da empresa diversas vezes no documento. Como resultado, o nome da empresa, será aquele que possuir maior valor acumulado.

O processo que implementa essa regra pode ser visualizado por meio do algoritmo 3.8.

### 3.3.3 Extração do elemento “Endereço”

O processo de extração de endereços de empresas, utilizado pelo Minerador Web, está baseado em processo empírico que resultou em premissas sobre a forma com que é usualmente apresentado o endereço nos *sites* das empresas. A observação dos exemplos a seguir permitem compreender melhor as regras adotadas:

8249 Bolsa Avenue. (número e nome da via)

MidWay city, CA 92655-1244 (cidade, uf e zona postal)

---

**Algoritmo 3.8** Extração do elemento “Nome da Empresa”, pela frequência das palavras

---

**Declaração:** :

$s$  {Site a ser analisado}

$vEmpresa$  {Vetor que contém o resultado da análise de cada *Trecho*}

**Retornar:**  $vEmpresa$  {a cada elemento de  $vEmpresa$  será somada a frequência de ocorrência de cada palavra encontrada em  $s$ }

**Para** cada  $vEmpresa_n$  **Faça**

**Enquanto** houver palavras em  $vEmpresa_x$  **Faça**

Contabilizar a frequência de ocorrência da palavra em  $s$

$vEmpresa_x + =$  quantidade de ocorrência contabilizada

A cada iteração analisar uma palavra de  $vEmpresa_x$

**Fim Enquanto**

**Fim Para**

---

8249 Bolsa Avenue. (número e nome da via)

Unit J Regatta Plaza (complemento)

MidWay city, CA 92655-1244 (cidade, uf e zona postal)

As premissas adotadas para a extração dos endereços de empresas estão fundamentadas na ontologia. A estratégia para a extração do endereço consiste em tentar identificar todos os componentes do endereço, com exceção do elemento “Complemento”, que é opcional.

O processo de extração do endereço tem início com a aplicação do algoritmo 3.9, que tenta identificar o elemento “UF” por meio da existência de duas letras, sendo a primeira maiúscula, isoladas no documento, isto é sem caracteres alfanuméricos à esquerda ou à direita.

---

**Algoritmo 3.9** Extração do elemento “Endereço”, atributo “UF”

---

**Declaração:** :

$s$  {Site a ser analisado}

**Retornar:** a posição em  $s$  onde se encontra a suposta “UF”

Buscar em  $s$ , 2 letras com as seguintes características:

Índice	Conteúdo
1	A primeira letra deve ser maiúscula
2	A posição anterior à primeira letra não deve ser caracter
3	A posição posterior à segunda letra não deve ser caracter

---

O próximo passo consiste na aplicação do algoritmo 3.10, que procura identificar o elemento “ZPT”. O algoritmo tenta encontrar uma seqüência de dígitos após o suposto atributo “UF”. O processo de busca não é feito indefinidamente. Nos

testes realizados, as seqüências de dígitos, referentes ao atributo “ZPT”, foram localizados no máximo 20 caracteres após o atributo “UF”. Esses caracteres intermediários formam marcações e atributos de marcações. Considerando-se uma margem de segurança, foi estabelecido uma distância de até 25 caracteres após o atributo “UF”.

---

**Algoritmo 3.10** Extração do elemento “Endereço”, atributo “ZPT”

---

**Declaração:** : {A partir da posição identificada no algoritmo 3.9}

$s$  {Site a ser analisado}

**Retornar:** retorna a posição em  $s$  onde se encontra a suposta “ZPT”

Buscar a posição de uma seqüência, que tenha as seguinte características

Índice	Conteúdo
1	A posição encontrada deve estar no máximo a uma distância de 25 caracteres
2	A seqüência numérica não deve exceder a “9” dígitos e nem ser menor do que 5 dígitos

---

Tendo sido encontrado os atributos “UF” e “ZPT”, são grandes as chances de serem identificados os elementos remanescentes.

Prosseguindo a análise em direção ao início do documento, a partir da posição do elemento “UF”, é possível identificar o nome da cidade. A característica que auxilia na caracterização do nome da cidade é a identificação do símbolo > (que forma uma marcação de início (< ... >)) ou a quebra de uma seqüência alfabética, por uma vírgula. Caso ocorra uma marcação de fim (< / ... >), o processo continua. O processo é descrito no algoritmo 3.11.

O resultado fornecido pelo algoritmo, necessita de ser “limpo”, isto é, deve ser aplicado um processo que identifica a existência de alguma marcação de final. Caso exista, deverá ser retirada, bem como qualquer sinal no extremo esquerdo e direito do nome da cidade.

O algoritmo 3.12 tenta identificar o elemento “Complemento”. O elemento “Complemento” ocorre antes do atributo “Cidade” e após o elemento “Via”. A característica principal para se identificar um complemento é o conjunto de dígitos antes do atributo “Cidade”, complementado por caracteres alfabéticos à esquerda, até a ocorrência de uma marcação ou uma vírgula.

O algoritmo 3.13 tenta extrair o elemento “Via”. O elemento “Via” ocorre antes do atributo “Complemento”, se este existir, ou, caso contrário, antes do atributo “Cidade”. O elemento “Via” é caracterizado pelo fato de terminar com caracteres

---

**Algoritmo 3.11** Extração do elemento “Endereço”, atributo “Cidade”
 

---

**Declaração:** :

$S$  {Site a ser analisado.}

$Ponteiro$  {aponta uma posição em  $S$ , inicializado com o valor de ( $Ponteiro - 1$ ) do atributo “UF” .}

$PosFinal$  {variável auxiliar para apontar o final do atributo “Cidade” .}

Retorna: um conjunto de caracteres alfabéticos representando o atributo “Cidade”, ou vazio se não encontrar.

**Enquanto**  $Ponteiro$  não apontar para um caracter alfabético em  $S$ , e existir símbolos a serem lidos em  $S$  **Faça**

Decrementar  $Ponteiro$  de 1.

**Fim Enquanto**

$PosFinal \leftarrow Ponteiro$ .

**Enquanto** Enquanto  $Ponteiro$  apontar para um caracter alfabético em  $S$ , ou existir uma marcação de final identificada por  $Ponteiro$  **Faça**

Decrementar  $Ponteiro$  de 1.

**Fim Enquanto**

trecho de  $S$ , entre  $Ponteiro$  e  $PosFinal$ .

---



---

**Algoritmo 3.12** Extração do elemento “Endereço”, atributo “Complemento”
 

---

**Declaração:** : {A partir da posição identificada no algoritmo 3.11}

$s$  {Site a ser analisado}

**Retornar:** retorna a posição em  $s$  onde se encontra o suposto nome do complemento  
A busca deve ser feita em direção ao início do *site*.

A partir da posição de Cidade, buscar o primeiro caracter  $< / >$  {Termino do elemento complemento}

A partir da última posição, busca o primeiro símbolo não caracter {Início do elemento Complemento}

---

alfabéticos e iniciar com dígitos.

---

**Algoritmo 3.13** Extração do elemento “Endereço”, atributo “Via”

---

**Declaração:** : {A partir da posição identificada no algoritmo 3.12 ou algoritmo 3.11}  
*s* {Site a ser analisado}

**Retornar:** retorna a posição em *s* onde se encontra o suposto nome da via

A busca deve ser feita em direção ao início do *site*.

A partir da posição de Complemento ou Cidade, buscar o primeiro caracter {Termino do elemento cidade}

A partir da última posição, busca o primeiro símbolo não caracter {Início do elemento Cidade}

---

### 3.3.4 Extração do elemento “Comunicação”

O elemento “Comunicação”, refere-se aos meios eletrônicos de contato de uma empresa, contendo os seguintes atributos: “Telefone”, “Fax” e “E-mail”. Foi constatado por processo empírico, e especificado na ontologia a presença de termos que antecedem os números de telefone e fax:

*call, phone, telephone, tel, voice, fax, toll free e tell me*

Em alguns casos, entre o termo e o número de telefone ou fax existem caracteres que não devem ser considerados. Como resultado de processo empírico, realizado na etapa de categorização do processo de criação da meta informação, foram encontrados até 20 caracteres referentes a marcações, a mesma quantidade com referência ao elemento “Endereço”, atributo “ZPT”. Sendo assim, como margem de segurança, foi adotado o valor de 25 caracteres como distância máxima entre o termo e o número. Um número é formado de 2 a 4 conjuntos de dígitos, podendo o último e penúltimo conjuntos serem formados de caracteres alfabéticos maiúsculos. O conjunto final, pode ser separado dos dígitos que o antecede por meio dos símbolos “-” ou “.”. Como exemplo de um número formado por 4 conjuntos de dígitos tem-se:

1-800-969-7829

1-800-969-STAX

Quando os números são formados de 3 conjuntos, é normal que o primeiro conjunto esteja entre parênteses, separado do segundo conjunto por um espaço. A extração dos números de telefone e fax é feita pelo algoritmo 3.14.

---

**Algoritmo 3.14** Extração do elemento “Comunicação”, atributos “Telefone” e “Fax”

---

**Declaração:** :

$s$  {Site a ser analisado}  
 $Chaves$  {Relação de palavras-chave}

**Retornar:** retorna a posição em  $s$  onde se encontra o suposto telefone/fax, ou zero caso contrário

**Para todo** cada elemento de  $Chave$  **Faça**

  Buscar  $Chave_i$  em  $s$ .

  A partir da posição encontrada localizar a uma distância de até 25 símbolos em direção ao fim de  $s$ , um dígito ou “(”

**Se** encontrou **então**

    Continua a pesquisa até não encontrar um dígito ou os símbolos “( ) . -” ou espaço.

**Se** encontrou **então**

**Se**  $Chave_i = \text{“Fax”}$  **então**

      Atribuir a informação a FAX

**senão**

      Atribuir a informação a Telefone

**Fim Se**

**Fim Se**

**Fim Se**

**Fim Para**

---

A busca do elemento *e-mail* consiste em procurar qualquer seqüência que possua o símbolo ‘@’, e que à direita possua um ponto decimal e à esquerda seja precedido, após alguns símbolos, pela palavra “mailto”. Muitos *e-mails* se referem ao *webmaster*, que não são de interesse direto entre a empresa e o usuário. Sendo assim, esses *e-mails* são desconsiderados.

A extração do endereço de e-mail é feita pelo algoritmo 3.15.

### 3.3.5 Extração dos elementos de “Produto”

O nome do produto, ocorre explicitamente no *site*, porém, não em formato estruturado. A estratégia para a extração dos produtos consiste em criar uma lista de produtos, bem como uma lista de especialização e relacioná-los. Essas listas são preparadas por um processo de busca por comparação. O processo de busca de produto e especialização pode ser visualizado no algoritmo 3.16.

O algoritmo 3.17 identifica as palavras relativas às características da especialização.

---

**Algoritmo 3.15** Extração do elemento “Comunicação”, atributo “e-mail”

---

**Declaração:** :

 $S$  {*Site* a ser analisado.}

 $Ponteiro$  {aponta uma posição em  $S$ , inicializado com o valor 0.}

 $PosInicial$  {variável auxiliar na identificação do início do atributo “Email”.}

 $PosFinal$  {variável auxiliar na identificação do final do atributo “Email”.}

**Enquanto** existir o símbolo ‘@’ em  $S$ , a partir de  $Ponteiro$  **Faça**
 $Ponteiro \leftarrow$  posição do símbolo ‘@’ em  $S$ 
 $PosInicio \leftarrow (Ponteiro + 1)$ .

**Enquanto**  $PosInicial$  em  $S$  apontar para caracteres alfanuméricos ou símbolos “.  
\_” **Faça**

Decrementar  $PosInicial$  de 1.

**Fim Enquanto**
 $PosFinal \leftarrow (Ponteiro + 1)$ .

**Enquanto**  $PosInicial$  em  $S$  apontar para caracteres alfanuméricos ou símbolos “.  
\_” **Faça**

Incrementar  $PosInicial$  de 1.

**Fim Enquanto**
 $Ponteiro \leftarrow PosInicial$ .

**Enquanto** trecho em  $S$ , iniciado em  $(Ponteiro - 6)$  e  $(Ponteiro)$ , for diferente de  
“MAILTO”, e  $(PosInicial - Ponteiro) < 25$  **Faça**

Decrementar  $Ponteiro$  de 1.

**Fim Enquanto**
**Se**  $(PosInicial - Ponteiro) < 25$  **então**
**Se** trecho de  $S$ , entre  $PosInicial$  e  $PosFinal$ , não contiver a palavra “webmas-  
ter” **então**

Adiciona em  $vEmail$  o trecho de  $S$ , entre  $PosInicial$  e  $PosFinal$ .

**Fim Se.**
**Fim Se**
**Fim Enquanto**


---

---

**Algoritmo 3.16** Extração do elemento “Produto” e dos atributos “Nome do Produto” e “Nome da Especialização”

---

**Declaração:** :

*vProdutos* {vetor bi-dimensional, contendo: Nome do Produto e Nome da Especialização.}

*S* {*site* a ser analisado.}

*PalSite* {palavra de *S* a ser analisada.}

*vPRD*[0..n][1..4] {vetor quadrimimensional, contendo:

Índice	Conteúdo
1	Produto
2	Especialização
3	Características
4	Valor

}

*PosPrd* {posição de *PalSite* em *vPRD*.}

**Para** cada *PalSite* de *S* **Faça**

**Se** *PalSite* existe em *vProdutos* **então**

*PosPrd* ← posição de *PalSite* em *vPRD*.

**Se** *PosPrd* é um produto **então**

Incrementar *PosPrd* de 1

**Fim Se**

*vPRD*[1] ← *vProdutos*[*PosPrd* - 1]

*vPRD*[2] ← *vProdutos*[*PosPrd*]

*vPRD*[3] ← *vProdutos*[*PosPrd* + 1]

*vPRD*[4] ← *vProdutos*[*PosPrd* + 2]

**Fim Se**

**Fim Para**

---

---

**Algoritmo 3.17** Extração do elemento “Produto”, atributo “Característica”
 

---

**Declaração:** :

*vProdutos* {vetor bi-dimensional, contendo: Produto e Especialização.}

*S* {site a ser analisado.}

*PalSite* {palavra de *S* a ser analisada.}

*vPRD* {vetor quadrimensional, contendo:

Índice	Conteúdo
1	Produto
2	Especialização
3	Características
4	Valor

}

*vCaract* {vetor unidimensional, contendo as características da especialização.}

*vValor*, {vetor unidimensional, contendo os valores da especialização.}

*PosDoc*, *PosDocInic*, *PosDocFin* { posição de *vPRD*[2] em *vDoc*.}

**Para** *i* = a elemento de *vPRD* **Faça**
**Enquanto** existir *vPRD*[2] em *vDoc* **Faça**

   *PosDoc* ← posição de *vPRD*[2] em *vDoc*.

   *PosDocInic* ← *PosDoc*.

   **Enquanto** *vDoc*[*PosDocInic*] != '.' e *PosDocInic* < (*PosDoc* + 100) **Faça**

     Decrementa *PosDocInic*.

   **Fim Enquanto**

   *PosDocFin* ← *PosDoc*.

   **Enquanto** *vDoc*[*PosDocFin*] != '.' e *PosDocFin* < (*PosDoc* + 100) **Faça**

     Incrementa *PosDocFin*

   **Fim Enquanto**

   Eliminar todas as marcações entre *PosDocInic* e *PosDocFin*, de *Frase*

   **Se** *Frase*, contém 3 conjunto de: no máximo 2 palavras e uma vírgula **então**

     *Frase* ← vazio.

   **Fim Se**

   **Se** *Frase* != vazio, **então**

     Características *vCaracteristica* ← *Frase*

   **Fim Se**
**Fim Enquanto**
**Fim Para**


---

O preço da especialização normalmente se encontra junto às características ou próximo ao nome da especialização. Assim, o processo deve buscar um símbolo ‘\$’ e uma seqüência numérica, terminada por “,nn” ou “.nn”, em que “nn” representa dígitos (0 a 9). Esse processo pode ser visualizado no algoritmo 3.18

---

**Algoritmo 3.18** Extração do elemento “Produto”, atributo “Preço”

---

**Declaração:** :

*vPreco*[0..*n*] {vetor unidimensional, contendo o preço da especialização.}

*Frase* {variável do tipo string, contendo a especialização sem eliminar marcações.}

*PosPrecoInic*, *PosPrecoFin* {posição do símbolo em *Frase*}

**Se** existir o símbolo ‘\$’ em *Frase* **então**

*PosPrecoInic* ← posição do símbolo ‘\$’ em *Frase*

*PosPrecoFin* ← *PosPrecoInic* + 1

**Enquanto** *Frase*[*PosPrecoFin*] = “ .,0123456789” **Faça**

Incrementa *PosPrecoFin* de 1.

**Fim Enquanto**

*vPreco* ← trecho de *Frase* entre *PosPrecoInic* e *PosPrecoFin*

**Fim Se**

---

Os elementos extraídos são armazenados em uma base de conhecimento, sendo para isso aplicada a etapa de “Conversão de um documento XML em uma linguagem lógica”.

### 3.3.6 Extração do elemento “Ramo de Atividade” e “Atividade”

Os elementos “Ramo de Atividade” e “Atividade”, são extraídos com a aplicação do classificador. A “carga da ontologia”, descrita pelo algoritmo 3.4 é feita para ambos os elementos. A ontologia específica para cada elemento é aplicada no classificador (algoritmo 3.2), juntamente com os parâmetros a serem analisados do documento.

Independente da variação de quantidade das classes nas categorias, a única alteração a ser feita é nos parâmetros das classes, que será carregada na etapa de “carga da ontologia”, não havendo necessidade de alterar o classificador.

### 3.3.7 Conversor de aplicação XML para linguagem lógica

Considerando que as informações contidas em aplicações XML constituem uma parte de uma base de conhecimento, a aplicação de mecanismos de busca com-

pletaria a proposta inicial. Mas a existência de diversas aplicações distribuídas em diversos arquivos, dificulta o acesso e a aplicação de mecanismos de busca.

Como exemplo da dificuldade: supondo existir diversas aplicações XML e uma das necessidades de informação do usuário seja identificar os meios de contato com empresas de uma determinada cidade, o processo a ser aplicado necessitaria que cada arquivo contendo a aplicação XML, deveria ser acessado, e em cada arquivo acessado deveria ser aplicada uma pesquisa de busca da informação. O resultado seria armazenado em um outro documento para análise e uso posterior. Uma nova necessidade de informação exigiria a criação de um novo método de busca com vários acessos (por meio do processo de abertura e fechamento de arquivos) às aplicações XML.

Uma solução para minimizar o acesso às diversas aplicações XML, seria juntar todas essas aplicações em uma única aplicação XML. Isso porém solucionaria parte dos problemas, pois não cobriria a criação de diferentes métodos de acesso, que são dependentes das necessidades de informações do usuário. A criação de diversos métodos também não resolveria, por que as necessidades são requisitos inconstantes, variáveis de acordo com o contexto requerido pelo usuário no momento da pesquisa.

Uma outra solução seria agrupar as informações contidas nas aplicações XML em um único documento lógico. De tal forma que, esse documento possibilite ao usuário a aplicação de perguntas, as quais exigiria a construção de novos relacionamentos das informações, resultando novas informações. Dessa forma, simulando uma aproximação ao mecanismo do pensamento humano, que supriria a inconstância das necessidades de informações do usuário. Sendo assim, o último passo desta etapa consiste em converter as aplicações XML em linguagem lógica. A linguagem lógica a ser utilizada é o Prolog, que permite expressar os objetos e suas relações, por meio de lógica simbólica ([7, 3, 25, 14]).

Na figura 3.21, pode ser visualizado um exemplo de um fato em linguagem Prolog, tendo como referência o exemplo parcial da aplicação XML (figura 3.20).

```
% O fato representa o endereço de uma empresa
endereco('pacific tire sales','CA', '92655-1244', 'MidWay
City', -, '8249 BOLSA AVENUE').
```

Figura 3.21 – Representação de um endereço de uma aplicação XML em um fato na linguagem Prolog

Um conjunto de fatos não formam uma base de conhecimento, para isso são necessárias a inclusão de regras que possam inferir conhecimento sobre os fatos. A inclusão das regras junto aos fatos, no Prolog, caracterizam uma “Base de Conhecimento”. Na figura 3.22, pode ser visualizado um conjunto de regras possíveis de serem implementadas, relativas ao fato representado na figura 3.21.

```
%buscar a via Y da empresa X
via(X,Y) : - endereco(X, -, -, -, -, Y).
%buscar a cidade Y da empresa X
cidade(X,Y) : - endereco(X, -, -, Y, -, -).
%buscar o estado Y da empresa X
estado(X,Y) : - endereco(X, Y, -, -, -, -).
%buscar a zona postal Y da empresa X
zonaPostal(X,Y) : - endereco(X, -, Y, -, -, -).
%buscar a via Y da empresa X no estado Z
viaEstado(X,Y,Z) : - estado(X,Z), via(X,Y).
```

Figura 3.22 – Representação de regras em Prolog, relativas ao fato da figura 3.21

Além das regras, uma base de conhecimento pode conter conhecimentos que não foram capturados, neste caso, pelo processo de conversão de páginas HTML. Esse conhecimento não capturado pode ser inserido diretamente à base. A esse conhecimento denomina-se “conhecimento de fundo” (*background knowledge*). Na figura 3.23, pode ser visualizado um exemplo de conhecimento de fundo (linhas 3 e 4) e uma regra (linhas 9 a 11) que o utiliza.

```
1  % Conjunto de fatos que contém o custo de locomoção
2  % até as cidades
3  taxaLocomocao('MIDWAY CITY',25.00).
4  taxaLocomocao('VIRGINIA', 30.00).
5
6  % Busca da empresa X, na cidade Y,
7  % de um tipo de pneu P, que possui a soma do preço
8  % do pneu com o valor de locomoção menor que S.
9  custo(X,Y,S,P,E) : - produto(X, P, E, -, V),
10                          cidade(X,Y),
11                          taxaLocomocao(Y,L), (V+L) < S.
```

Figura 3.23 – Código Prolog contendo exemplo de conhecimento de fundo e sua aplicação

A regra “*custo(X, Y, S, P, E)*”, quando aplicada, utilizará informações coletadas como: cidade (*Y*), empresa (*X*), produto (*P*) e sua especialização (*E*). Além de

dois fatos não coletados que utilizarão o parâmetro de limite de valor aceito ( $S$ ), como critério para selecionar os fatos cuja a soma do valor do pneu com a taxa de locomoção até a cidade, seja inferior a esse limite.

Com base nos exemplos acima e utilizando a linguagem lógica Prolog, a proposta de armazenar as informações em uma base de conhecimento consiste em aplicar um utilitário que leia a aplicação XML, figura 3.20 e converta as informações contidas entre as marcações, em fatos na linguagem Prolog. No arquivo nomeado como “pacific.xml” no diretório “Anexos” constante no cdrom anexo, pode ser visualizado o código completo gerado pelo aplicativo Minerador Web para a empresa “Pacific Tire Sales, inc.”, sendo esse mesmo anexo resumido e descrito na tabela 3.27.

A conversão da aplicação XML para a linguagem lógica Prolog, e a inserção dessa conversão em uma base de conhecimento, completa o processo de busca da informação em um *site* escrito em HTML e a criação da base de conhecimento.

Uma base de conhecimento pode conter conhecimentos que não foram capturados pelo processo de extração de informações de um *site* HTML. Uma necessidade desse conhecimento poderia ser o resultado de uma comparação de custos entre ir buscar o produto na loja ou solicitar que o mesmo seja entregue. Para inferir este resultado é necessário obter as seguintes informações: o custo do frete, a distância entre as cidades e o custo do quilômetro rodado, estas informações não constam no *site* e caso constem não devem ser muito freqüentes. Sendo assim, para suprir essas informações é necessário inserí-las na base de conhecimento.

Na figura 3.24, pode ser visualizado um exemplo de conhecimento de fundo inserido na base de conhecimento Prolog, para suprir a necessidade da informação anteriormente descrita.

XML Prolog	<b>Elemento: Nome da Empresa</b> <nomeEmpresa>pacific tire sales, inc</nomeEmpresa> O nome da empresa será utilizado na construção dos fatos seguintes
XML Prolog	<b>Elemento: Site</b> <site>pacific.cpct</site> site('pacific tire sales, inc', 'pacific.cpct').
XML Prolog	<b>Elemento: Nome do Ramo de Atividade</b> <nomeRamo>'Automóvel'</nomeRamo> ramo('pacific tire sales, inc', 'Automóvel').
XML Prolog	<b>Elemento: Nome da Atividade</b> <nomeAtividade>Fabricante</nomeAtividade> ramo('pacific tire sales, inc', 'Fabricante').
XML Prolog	<b>Elemento: Endereço Físico</b> <uf>CA</uf> <zpt>92655</zpt> <cidade>Midway City</cidade> <complemento></complemento> <via>8249 Bolsa Ave</via> endereco( 'pacific tire sales, inc', 'CA', '92655', 'Midway City', -, '8249 Bolsa Ave').
XML Prolog	<b>Elemento: Endereço de Comunicação - telefone</b> <telefone>4714.892.2093</telefone> <fax></fax> <email></email> meiosCOM( 'pacific tire sales, inc', '714.892.2093', -, - ).
XML Prolog	<b>Elemento: Produto</b> <nomeProduto>tire</nomeProduto> <especializacao>'185/65HR15'</especializacao> <caracteristica>'SPECIALS 185/65HR,'<caracteristica> <preco>35.00</preco> produto( 'pacific tire sales, inc', tire, '185/65HR15', 'SPECIALS 185/65HR,', 35.00).

Tabela 3.27 – Conversão de uma aplicação XML em Prolog

```

% [Conhecimento de fundo]
%Valor de frete para a entrega de uma determinada cidade
frete('MIDWAY CITY',30.00).

%Distância em km de uma determinada cidade
distancia('MIDWAY CITY',110.00).

%Custo por km rodado.
custoKm(0.85).

%Cidade da Empresa que possua o valor da Especialização mais o frete, menor ou
igual a um valor limite
ondeComFrete(Empresa, Cidade, Especializacao, ValorResultado, ValorLimite):-
    cidade(Empresa, Cidade),
    produto(Empresa, -, Especializacao, -, Preco),
    (Preco > 0),
    frete(Cidade, ValorFrete),
    ValorResultado is (Preco + ValorFrete),
    ValorResultado =| ValorLimite.

%Cidade da Empresa que possua o valor da Especialização mais a km multiplicado
pelo custo do km menor ou igual a um valor limite
ondeComBuscar(Empresa, Cidade, Especializacao, ValorResultado,
ValorLimite):-
    cidade(Empresa, Cidade),
    produto(Empresa, -, Especializacao, -, Preco),
    (Preco > 0),
    distancia(Cidade, Distancia),
    custoKm(CustoKm),
    ValorResultado is (Preco + (Distancia * CustoKm)),
    ValorResultado =< ValorLimite.

```

Figura 3.24 – Exemplo de conhecimento de fundo, implementado em Prolog

# Capítulo 4

## RESULTADOS EXPERIMENTAIS

Este capítulo contém os resultados experimentais obtidos com a aplicação dos processos apresentados no capítulo 3. Na primeira seção, são apresentados os resultados experimentais do processo de extração das informações dos documentos HTML. Na segunda seção, é apresentada a base de conhecimento obtida, e na última seção, são apresentados alguns experimentos com consultas em Prolog sobre a base de conhecimento.

### 4.1 Performance do classificador resultante

Esta seção contém os resultados obtidos sobre a busca da informação utilizando o Minerador Web.

Foram aplicados dois conjuntos de testes. O primeiro conjunto de teste foi construído com uma relação de *sites* pertencentes às classes das categorias “Ramo de Atividade” e “Atividade”, isto é, contendo um conjunto de *sites* distribuídos em 4 categorias. A distribuição dos *sites* pode ser visualizada na tabela 4.1.

Na tabela 4.2 podem ser visualizados os resultados obtidos com a aplicação do Minerador Web, para as categorias “Ramo de atividade” e “Atividade”, em que para cada uma dessas categorias foram apresentadas as quantidades de coletas corretas e as quantidades de coletas erradas e seus respectivo valores percentuais, por classe .

Com relação à tabela 4.2, nota-se um acerto de 100% na categoria “Ramo de

<b>Categorias</b>	<b>Qtd. sites</b>
Automóvel	16
Computador	13
Saúde	6
Vestuário	13

Tabela 4.1 – Distribuição dos *sites* selecionados para o primeiro conjunto de teste.

Categoria	Ramo				Atividade			
	Errado		Certo		Errado		Certo	
Classe/	Qtd.	Perc	Qtd.	Perc	Qtd.	Perc.	Qtd.	Perc
<b>Automóvel</b>	10	60%	6	40%	12	75%	4	25%
<b>Computador</b>	4	30%	9	70%	10	77%	3	13%
<b>Saúde</b>	0	0%	6	100%	5	83%	1	17%
<b>Vestuário</b>	4	30%	9	70%	11	84%	2	16%

Tabela 4.2 – Resultados obtidos com o Minerador Web, para informação implícita

Atividade” para a classe “Saúde”, isto é decorrente do fato de que o classificador sempre fornecerá uma classificação, e não havendo informações suficientes, o *site* analisado será classificado sempre na classe que tiver mais probabilidade de ocorrer, isto é a classe que possuir menos informações.

No geral, o percentual de acertos da categoria “Ramo de Atividade” é maior que o percentual de acertos da categoria “Atividade”, essa diferença ocorreu pela forma com que foi construído o conjunto de treinamento. O conjunto de treinamento para a categoria “Ramo de Atividade”, foi construído com base em uma relação existente no portal Yahoo, já o conjunto de treinamento da categoria “Atividade” não possuía uma relação pronta, tendo sido construída com base nas referências dos *sites*. Pelo motivo dessas referências não possuírem informações suficientes, a classificação apresentou baixa performance. Vale ressaltar que as referências dos *sites* foram utilizadas apenas para identificar as classes da categoria “Atividade” e agrupar os *sites* sobre as categorias criadas (Fabricante, Prestação de Serviço e Vendas), e que o conjunto de treinamento foi construído sobre as páginas coletadas.

Na tabela 4.4 estão relacionados os resultados obtidos na coleta das categorias: “Nome da empresa”, “Endereço” e “Comunicação”, sobre o mesmo conjunto de *sites* utilizado nas categorias “Ramo de Atividade” e “Atividade”. Deve ser observado que nem todas as categorias constam nos *sites*, sendo assim, na tabela 4.3 pode ser visualizada a quantidades de *sites* por categorias analisada.

O alto índice de coleta da informação corretas para a categoria “Nome Empresa” é decorrente da aplicação de diversas heurísticas, criadas para cada caso em que as informações tinham grandes possibilidades de ocorrer. O mesmo não ocorrendo com as categorias “Endereço” e “Comunicação”, em que foram aplicadas apenas a busca de estrutura, não sendo dado ênfase em identificar estruturas variantes,

Categoria	Qtd. Sites
Nome Empresa	15
Endereço	12
Comunicação	9

Tabela 4.3 – Distribuição dos *sites* por elemento no conjunto de teste

Categoria	Nome Empresa		Endereço		Comunicação	
	Qtd.	Perc	Qtd.	Perc	Qtd.	Perc.
Automóvel	11	73%	1	8%	2	22%
Computador	11	73%	9	75%	3	33%
Saúde	6	40%	1	8%	1	11%
Vestuário	10	67%	2	17%	7	78%

Tabela 4.4 – Resultados obtidos com o Minerador Web, para informação explícita

como por exemplo na categoria “Endereço”, assumir a falta de algum atributo na informação, isto é, aceitar a informação de endereço mesmo faltando o atributo “UF”, e/ou até o atributo “ZPT”.

Na figura 4.1 pode ser visualizado graficamente os resultados obtidos nas tabelas 4.2 e 4.4.

Em um segundo teste, composto de 76 *sites* aleatórios, foi aplicado ao Minerador Web. Vale ressaltar que esses 76 *sites* aleatórios não constam do conjunto de treinamento e nem do primeiro conjunto de teste. O resultado desse segundo teste pode ser visualizado na figura 4.2.

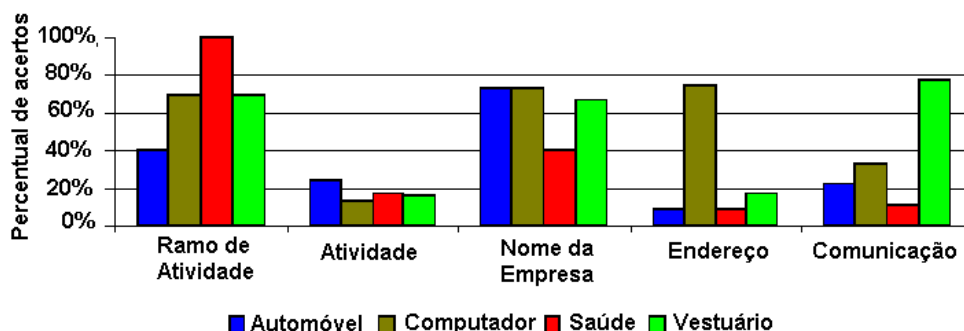


Figura 4.1 – Resultados obtidos com a utilização do Minerador Web no primeiro conjunto de teste

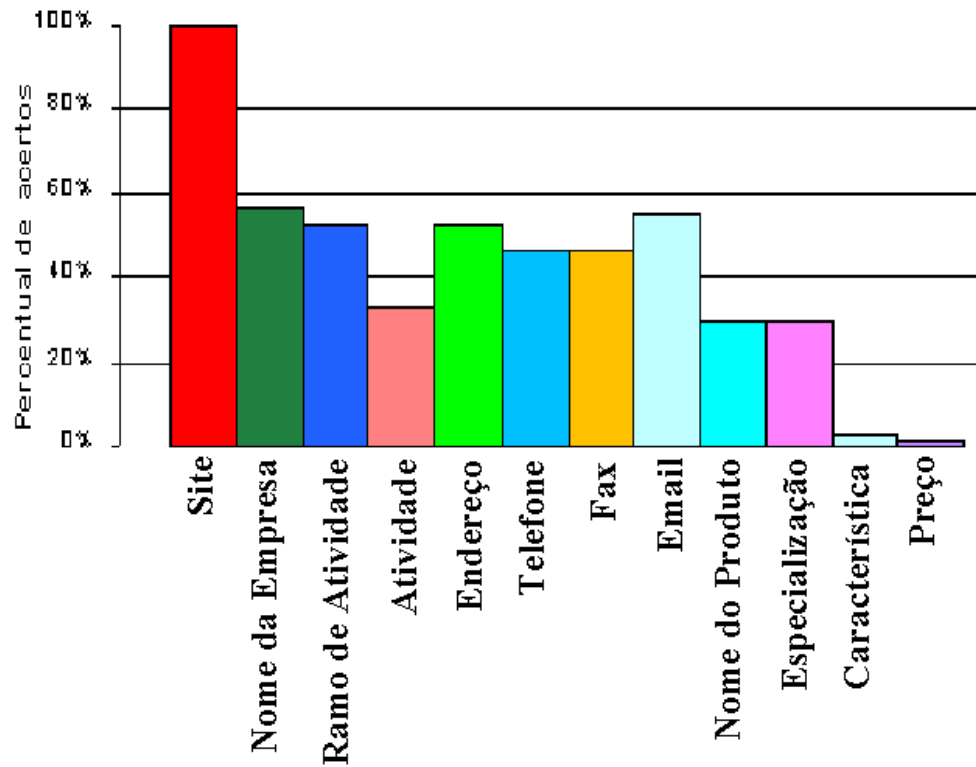


Figura 4.2 – Resultados obtidos com a utilização do Minerador Web no segundo conjunto de teste

Note que este segundo resultado, mais completo, por conter os valores de acertos para o elemento ‘Produto’. Com relação à figura 4.2, pode ser notado a alta percentagem do elemento “Site” (100%), o motivo dessa elevada percentagem está relacionada com a sua extração, em que o endereço do *site* fornecido é a própria informação do elemento “Site”.

O elemento “Nome da Empresa”, apesar de constar em todos os *sites*, não possui uma estrutura definida. Podendo ser encontrado no texto, entre as marcações “<title>” ou ainda abreviados, como já mencionado nos capítulos anteriores. Para sua extração houve a necessidade de criar heurísticas que auxiliam na captura do elemento nessas 3 possibilidades.

Para as categorias “Ramo de Atividade” e “Atividade” é aplicado o classificador descrito na subseção 2.5.3. Para o elemento “Atividade”, o conjunto de treinamento foi criada a partir de análise visual sobre a estrutura montada com as referências dos *sites*. Sendo poucas as informações obtidas nessa estrutura, resultando na baixa percentagem de acerto. Já para a categoria “Ramo de Atividade”, o conjunto de treinamento foi criado a partir de uma pré classificação fornecida pelo portal Yahoo.

O elemento “Endereço” é localizado por busca de estrutura, sendo somente aceitas as estruturas que contivessem todos os atributos, exceto o atributo opcional “Complemento”.

Os atributos “Telefone”, “Fax” e “Email”, do elemento “Comunicação”, também foram coletados por busca de estrutura, tendo sido criadas heurísticas para coleta dos números de telefone e de fax, pois estes aparecem nos textos dos *sites* com caracteres alfanuméricos. No caso do atributo “Email” foi aplicada a restrição de desconsiderar todos os *e-mails* que contivessem a palavra “webmaster”.

Os atributos “Nome do Produto” e “Especialização”, do elemento Produto, são identificados por meio de comparação a uma lista de especialização relacionada a produto. Cada palavra (especialização), dessa lista, é pesquisada no *site*, e caso a palavra ocorra o atributo “Nome do Produto” é preenchido com a palavra que se relaciona com a especialização na lista. Ressaltando que, a lista construída não está completa, por isso a baixa performance no percentual de acertos. Os atributos “Característica” e “Preço”, do elemento “Produto”, só são identificados se existir o elemento “Especialização”. A baixa performance dada ao atributo “Característica” é relacionado à baixa quantidade da especialização relacionada,

site(<nome da empresa>, <endereço de captura do site >).  
 ramo(<nome da empresa>, <Ramo de atividade>).  
 atividade(<nome da empresa>, <Atividade>).  
 endereco(<nome da empresa>, <UF>, <Zona postal>, <Cidade>, <Complemento>, <Via>).  
 meiosCOM(<nome da empresa>, <número do telefone>, <número do fax>, <endereço eletrônico de correspondência>).  
 produto(<nome da empresa>, <nome do produto>, <especialização do produto>, <características da especialização>, <preço da especialização>).

Figura 4.3 – Referência ao conteúdo dos fatos da base de conhecimento

a qual consta em uma lista não completa. Logo, o atributo “Preço” relacionado, também, com o atributo “Especialização” resultou em uma baixa performance no resultado.

## 4.2 Base de conhecimento resultante da aplicação do Minerador Web

A base de conhecimento resultante é apresentada parcialmente nesta seção. A base de conhecimento obtida pode ser visualizada na sua totalidade no arquivo “BaseLógica.pl” (cdrom anexo).

A base de conhecimento está dividida em 3 partes: o conhecimento de fundo, os fatos e as regras que permitem a realização de consultas diretas (diretas no sentido de que não necessitam do conhecimento de fundo) aos fatos capturados.

Na figura 4.3 estão relacionados alguns fatos da base de conhecimento escritos com base na tabela 3.27. Note que a informação útil, que se encontrava entre as marcações, tornaram-se objetos que formaram um fato cujo o predicado é relacionado à marcação da aplicação XML. A título de referência na figura 4.3 estão relacionadas as descrições sobre o conteúdo dos fatos.

Na figura 4.4 pode ser notada a inexistência de valores para os elementos “Complemento” (linha 4) e “Comunicação” (linha 5). Existem empresas que não possuem todas as informações. Esses valores opcionais estão indicados na descrição formal da ontologia em uma DTD pelos símbolos ‘?’ e ‘\*’.

A seguir um exemplo de aplicação dos fatos coletados em *sites* com fatos inseridos

```

site( 'pacific tire sales, inc', 'pacific.cpct' ).
ramo( 'pacific tire sales, inc', 'Automóvel' ).
atividade( 'pacific tire sales, inc', 'Fabricante' ).
endereco( 'pacific tire sales, inc', 'Ca', '92655', 'Midway City', -, '8249
Bolsa Ave' ).
meiosCOM( 'pacific tire sales, inc', '714.892.2093', -, - ).
produto( 'pacific tire sales, inc', tire, '155R12', 'PASSENGER / PER-
FORMANCE TIRES METRIC P-METRIC 155R,', 18.00).

```

Figura 4.4 – Relação parcial de fatos da base de conhecimento gerada

para “ampliar” o conhecimento adquirido, tendo como base as estruturas dos fatos da tabela 3.27.

- Pesquisar o preço, com frete incluso, inferior a \$60,00, a cidade de localização e o nome das empresas que fornecem pneus do modelo '155/80R13'.  
?- OndeComFrete(Empresa, Cidade, '155/80R13', Preco, 60).  
Empresa = 'pacific tire sales, inc'  
Cidade = 'Midway City'  
Preco = 49
- Pesquisar as características da especialização “*Brakes*” de produtos “tire”, o preço e o nome da empresa.  
?- produto(Empresas, tire, 'BRAKES', Caracteristicas, Preco).  
Empresas = 'competition tire south, inc'  
Caracteristicas = 'The first hydraulic disc brakes for airplanes developed by Goodyear, New test equipment for airplane wheels and brakes costing, Air Force contract of its type awarded to Goodyear Aerospace for aircraft wheels, brakes and parts,'  
Preco = 2 ;  
Empresas = 'pacific tire sales, inc'  
Caracteristicas = 'We replace brakes, shocks / struts, batteries, offer oil changes and do alignment service, TIRES BRAKES ALIGNMENTS SHOCKS / STRUTS,'  
Preco = 0

# Capítulo 5

## CONSIDERAÇÕES FINAIS

Um dos produtos primários que resultaram do trabalho de pesquisa realizado durante o desenvolvimento dessa dissertação foi o Minerador Web. O Minerador Web é um conjunto de programas que tem a função de extrair informações das páginas HTML de *sites* acessíveis na Web e armazená-las em uma base de conhecimento descrita por meio do uso da linguagem Prolog. Com o intuito de tornar o problema tratável, foi adotada a restrição de que o estudo seria feito sobre o domínio do conhecimento sobre empresas.

O desenvolvimento do Minerador Web foi motivado pelo fenômeno decorrente da enorme quantidade e diversidade de dados disponíveis na Web, que inviabilizam um processo de busca meramente manual. Atualmente existem diversos portais de busca, que oferecem mecanismos de auxílio ao processo de pesquisa de informações desejadas. Tais mecanismos são baseados primariamente num processo de comparação sintática entre expressões lógicas, que são fornecidas pelo usuário do portal de busca e que tentam modelar o tipo da informação desejada, e o texto presente nas páginas HTML dos *sites* da Web. Entretanto, tais mecanismos de busca muitas vezes acabam fornecendo uma quantidade enorme de *sites* candidatos a conter a informação desejada, sendo que, em termos efetivos, apenas uma fração desses *sites* contém realmente algo de interesse.

O Minerador Web foi construído de forma a permitir um estudo sobre a eficácia de técnicas alternativas de extração da informação. As técnicas utilizadas no Minerador Web fazem uso de:

- Modelagem do domínio do conhecimento (*sites* de empresas), por meio de uma ontologia (ontologia de empresas), descrita formalmente por uma DTD XML;
- Caracterização dos objetos da ontologia (elementos da DTD) que aparecem explicitamente nas páginas HTML (endereço, telefone, produtos, etc), de forma a permitir criar heurísticas e algoritmos especializados na extração desses elementos;

- Construção de classificadores probabilísticos, baseado numa variante do método Naive Bayes, para a extração dos elementos implícitos (ramo de atividade e atividade da empresa);

O processo de extração dos elementos implícitos (ramo de atividade e atividade da empresa) apresentou resultados com “baixa” performance em relação ao esperado, com a utilização de classificadores baseados apenas no método Naive Bayes. Vale ressaltar que, esses resultados foram melhores do que a aplicação de um processo manual de extração dos elementos implícitos. Porém, essa “baixa” performance tem dois motivos:

- Dificuldade de se estimar adequadamente os valores das probabilidades de ocorrência das palavras;
- Conjunto de treinamento inadequado, devido à dificuldade de se separar as palavras realmente significativas para a discriminação de uma classe;

A construção da ontologia seguiu os passos de um método relativamente simples, mas operacionalmente trabalhoso: utilizar uma pré-classificação de *sites* disponível no portal de busca Yahoo, fazer a análise visual de um conjunto representativo desses *sites* de forma a abstrair os principais objetos e relações do domínio de empresas. Uma restrição adicional adotada foi que os elementos da conceituação deveriam estar presentes em boa parte dos *sites*, senão o processo de extração das informações não teria sentido.

Para o conjunto de informações implícitas foi criado um classificador que utiliza um processo mais complexo de estimação de probabilidades, baseado no método de suavização (*smoothing* de Witten-Bell). Também foi melhorada a “qualidade” do conjunto de treinamento, eliminando-se as palavras que não contribuem para o processo de classificação (*stop-words*) e aquelas que não tem poder de discriminação suficiente (descrito por meio do valor da Informação Mútua Média - AMI). Essas providências melhoraram sensivelmente os resultados do processo de classificação de *sites*.

Os processos de extração de informações explícitas dependeu da criação e identificação de heurísticas especializadas para cada elemento. Essas heurísticas foram baseadas nas estruturas com que as informações aparecem nas páginas HTML. Em termos de resultados, bons índices de extração foram obtidos para o elemento

“Nome da Empresa”. Para os elementos “Endereço” e “Comunicação” os resultados não foram bons devido à dificuldade de se encontrar uma boa heurística. Outro motivo encontrado para a baixa performance no caso desses elementos foi o fato de que muitas empresas utilizam mecanismos não abrangidos neste trabalho (*applets* Java) para apresentar esse tipo de informação.

No caso do elemento “Produto”, a pesquisa foi feita apenas por comparação de palavras de uma lista de especialização com as palavras dos *sites*, apenas a título de análise de resultados. Mesmo porque, haveria necessidade de criar uma heurística específica para obter os atributos do elemento “Produto”. Pois esses atributos não aparecem dentro de estruturas, podendo estarem em um texto que comente o produto ou em uma tabela, e ainda assim, estando em uma tabela pode ocorrer de estarem alinhados em colunas ou em linhas.

Sob uma visão geral, o Minerador Web melhora o processo de busca da informação na Web com base em 3 fatores:

1. Eliminação do processo de análise visual dos sites;
2. Seleção apenas das informações de interesse de uma área específica do conhecimento;
3. Possibilidade de fazer perguntas sobre a base de conhecimento, que os mecanismos de busca fornecidos pelos portais não conseguem atender.

Visando uma evolução do Minerador Web, futuros estudos e desenvolvimentos relacionados podem incluir:

1. Descrição das ontologias por meio de “schema” XML, ao invés de DTD;
2. Aplicação de técnicas probabilísticas para os demais elementos, com o objetivo de eliminar ou reduzir a necessidade de heurísticas específicas;
3. Tornar a interface e o processo de utilização do Minerador Web mais amigável;
4. Automatizar o processo de atualização do conjunto de treinamento (categorização), utilizando os *sites* pesquisados;
5. Implementar uma interface de usuário para a base de conhecimento;
6. Melhorar os mecanismos de seleção de palavras para compor os conjuntos de treinamento.

# Referências

- [1] ALTAVISTA. Portal de acesso a informações na Web. Disponível em: <[www.altavista.com](http://www.altavista.com)>. Acesso em: mar. 2001.
- [2] BELL, Timothy C.; WITTEN, Ian H.; CLEARY, John G. Modeling for text compression. **ACM Computing Surveys**, v.21, n.4, December, 1989.
- [3] BRATKO, Ivan. **Prolog programming for Artificial Intelligence**. Addison Wesley, 1990.
- [4] CHEN, Stanley F.; ROSENFELD, Ronald. **A Gaussian Prior for Smoothing Maximum Entropy Models**. School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, February, 1999.
- [5] CHEN, Staley F.; GOODMAN, Joshua. **An Empirical Study of Smoothing Techniques for Language Modeling**. Center for Research in Computing technology Harvard University Cambridge, Massachusetts. August, 1998.
- [6] CLARK, James. **Comparison of SGML and XML**. Disponível em: <<http://www.w3.org/TR/NOTE-sgml-xml-971215>>. World Wide Web Consortium, December, 1997.
- [7] CLOCKSIN, William. F.; MELLISH, Christopher. S. **Programming in Prolog**. Springer, Berlin Heidelberg, 1994.
- [8] CRAVEN, Mark; et. all. Learning to construct knowledge bases from the World Wide Web. **Artificial Intelligence**, v.1, n.118, p.69-113, 2000.
- [9] DONOVAN, Truly. **Industrial-Strength SGML - An Introduction to Enterprise Publishing**. New Jersey. Prentice-Hall, 1997.
- [10] EMDEN, M. H. Van; KOWALSKI R. A. The Semantics of Predicate Logic as a Programming Language. **Journal of the Association for Computing Machinery**, v. 23, n. 4, p. 733-742, October 1976.
- [11] ERDMANN, Michael; STUDER, Rudi. How to structure and access XML, documents with ontologies. **Data Knowledge Engineering**, v.1, n.36, p.317-335, 2001.

- [12] GAMPER, Johann et. al. **Combining Ontologies and Terminologies in Information Systems**. European Academy Bozen, Bolzano, 1999.
- [13] GENESERETH, Michael R.; NILSSON, Nils J. **Logical Foundations of Artificial Intelligence**. Stanford University, 1988.
- [14] GIBBINS, Peter. **Logic with Prolog**. (Oxford Applied Mathematics and Computing Science Series). New York. 1988.
- [15] GRUBER, Thomas R. **A Translation Approach to Portable Ontology Specifications**. Computer Science Department Stanford University, September 1992.
- [16] GUARINO, Nicola. Formal Ontology, Conceptual Analysis and Knowledge Representation. **International Journal of Human-Computer Studies**, 625640, n.2/3:625640, v.43, 1995.
- [17] HELLMAN, Reed. A Semantic Approach Adds Meaning to the Web. *Computer*. Los Alamitos-CA, v.32, n.12,p.13-16, December 1999.
- [18] HEGENBERG, Leônidas. **Lógica: Simbolização e Dedução**. Universidade de São Paulo, São Paulo, 1975.
- [19] KELSEY, Robert L.; HARTLEY, Roger T.; WEBSTER, Robert B. **An Object-Based Methodology for Knowledge Representation in SGML**. IEEE n.1082-3409/97, p.304-310, 1997.
- [20] O'LEARY, Daniel E. Using AI in Knowledge Management: Knowledge Bases and Ontologies. **IEEE Intelligent Systems**, p.34-39. May/June 1998.
- [21] LEWIS, David D.; RIGUETTE, Marc A Comparison of Two Learning Algorithms for Text Categorization. Proceedings of (SDAIR)-94, 3rd Annual Symposium on Document Analysis and Information Retrieval. Las Vegas, US. p.81-93, April, 1994.
- [22] MALER, Even; ANDALOUSSI, Jeanne El. **Developing SGML DTDs from text to model to markup**. New Jersey. Prentice-Hall. 1996.
- [23] McCALLUM, Andrew K.; NIGAM, Kamal **A Comparasion of Event Models for Naive Bayes text Classification**. In AAAI-98 Workshop on Learning for Text Categorization, 1998. Disponível em: [www.cs.cmu.edu/maccallum](http://www.cs.cmu.edu/maccallum), 1998

- [24] MITCHELL, Tom M. **Machine Learning**. The MacGraw-Hill Companies, Inc. 1997
- [25] McCORD, Michael; SOWA, John F.; WILSON, Walter G. **Knowledge System and Prolog**. Addison-Wesley. 1987.
- [26] MOH, Chuang-Hue; LIM, Ee-Peng; NG, Wee-Keong. DTD-Miner: A tool for Mining DTD from XML Documents. IEEE 0-7695-0610-0/00
- [27] MONTEIRO, Silvana Drumond. A Forma Eletrônica do Hipertexto. **Ci. Inf.**, Brasília, v.29, n. 1, p. 25-39, jan/abr 2000.
- [28] NETSCAPE. **Portal de acesso a informações na Web**. Disponível em: <www.netscape.net>. Acesso em: mar. 2001.
- [29] NEWELL, Allen. The Knowledge Level. **Artificial intelligence**, n.18, p.87-127. 1982.
- [30] NIGAM, Kamal et all. **Learning to Classify Text from Labeled and Unlabeled Documents**. 1Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence. AAAI Press, Menlo Park, US. Madison, US. p.792-799, 1998.
- [31] PARSAYE, Kamran et. al. **Intelligent Databases - Object-Oriented, Deductive Hypermedia Technologies**. Malloy Lithographing, Inc., 1989.
- [32] PFEIFFER, Ralf I. Tutorial 1: Overview of XML. IBM XML Technology Group. Disponível em: [www.ibm.com/XML](http://www.ibm.com/XML). Acesso em: nov 2000.
- [33] PFEIFFER, Ralf I. Tutorial 2: Writing XML Documents. IBM XML Technology Group, Disponível em: [www.ibm.com/XML](http://www.ibm.com/XML). Acesso em: nov 2000.
- [34] Promotion Guide, A . Free website promotion tutorial. Dicas de promoção de sites. Disponível em: <[www.apromotionguide.com](http://www.apromotionguide.com)>. Acesso em: nov. 2002.
- [35] RISTAD, Eric Sven. A Natural Law of Succession. **Research Report CS-TR-495-95**, rev. July 1995.
- [36] RUSSEL, Stuart Jonathan; NORVIG, Peter. **Artificial Intelligence - A Modern Approach**. New Jersey. Prentice-Hall. 1995.
- [37] SEBASTIANI, FRABRIZIO Machine Learning in Automated text Categorization. **ACM Computing Surveys**, v. 34, n.1, March 2002, pp.1-47

- [38] SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. Porto Alegre. Bookmann. 2000.
- [39] TIDWELL, Doug **Building an XML application, step 1: Writing a DTD**. IBM XML Technical Strategy Group, TaskGuide Development. Disponível em [www.ibm.com/XML](http://www.ibm.com/XML). Acesso em: 2000.
- [40] TIDWELL, Doug **Building an XML application, step 2: Generating XML from a Data Store**. IBM XML Technical Strategy Group, TaskGuide Development. Disponível em: [www.ibm.com/XML](http://www.ibm.com/XML). Acesso em: 2000.
- [41] TIDWELL, Doug **Building an XML application, step 3: Converting XML into HTML with the Document Object Model (DOM)**. IBM XML Technical Strategy Group, TaskGuide Development. Disponível em: [www.ibm.com/XML](http://www.ibm.com/XML). Acesso: 2000.
- [42] TREAT, Harold. Plugging in to XML. IBM. DB2 magazine, volume 4, numero 4, 1999.
- [43] TURNER, Ronald C. et. all. **Readme.1st SGML for Writers and Editors**. Prentice Hall Inc., New Jersey, 1996.
- [44] VET, Paul E. van der; MARS, Nicolas J. I. Bootom-Up Construction of Ontologies. **IEEE Transaction on Knowledge and Data Engineering**, n. 4, v. 10, p.513-527. July/August 1998.
- [45] W3C. **Portal de acesso a informações de linguagens de marcação como HTML e XML**. Disponível em: [www.wc.org](http://www.wc.org). Acesso em: mar 2001.
- [46] WITTEN, Ian H.; BELL, Timothy C. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. **IEEE Transaction on information theory**, v.37, n.4, p.1085-1094, July, 1991
- [47] YAHOO. **Portal de acesso a informações na Web**. Disponível em: [www.yahoo.com](http://www.yahoo.com). Acesso em: Mar. 2001.

**CD-ROM** : em anexo, contém o código fonte dos programas desenvolvidos, arquivos de programas utilizados, dados dos conjuntos de treinamento dos classificadores e a base de conhecimento resultante.