000179-1

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

SES - SISTEMA DE EXTRAÇÃO SEMÂNTICA DE INFORMAÇÕES

por

Rui Gureghian Scarinci

Dissertação submetida à avaliação, como requisito parcial para
a obtenção do grau de Mestre em
Ciência da Computação

SABi
UFRGS
05232264

Prof. Dr. José Palazzo M. de Oliveira
Orientador

Porto Alegre, março de 1997.

UFRGS
INSTITUTO DE INFORMÁTICA

CIP - Catalogação da Publicação

Scarinci, Rui Gureghian

SES - Sistema de Extração Semântica de Informações / Rui Gureghian Scarinci — Porto Alegre: CPGCC da UFRGS, 1997.
165 f.: il.

Dissertação (mestrado) — Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, BR-RS, 1996. Orientador: Palazzo, José Moreira de Oliveira.

1. Extração de Informações. 2. Recuperação de Informações. 3. Bancos de Dados Temporais. I. Palazzo, José Moreira de Oliveira. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Sistema de Biblioteca da UFRGS

33052

681.32.071(043)
S2855

INF
1997/196204-9
1997/06/30

MOD. 2.3.2

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Profa. Wrana Panizzi
Pró-Reitor de Pesquisa e Pós-Graduação: Prof. José Carlos Ferraz Hennemann
Diretor do Instituto de Informática: Prof. Roberto Tom Price
Coordenador do CPGCC: Prof. Flávio Rech Wagner
Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira

Agradecimentos

Durante os dois anos transcorridos desde meu ingresso no Curso de Pós-Graduação em Ciência da Computação até o presente, conheci, trabalhei e mantive relações de amizade com diversas pessoas, que muito auxiliaram-me nesta caminhada. Particularmente, gostaria de destacar e agradecer às seguintes pessoas:

- Professor Dr. José Palazzo Moreira de Oliveira, pela orientação, conselhos e ações práticas efetivadas com o intuito de oferecer as condições materiais e psicológicas para que eu pudesse realizar o meu trabalho.

- Agradeço a minha mãe e eterna orientadora, Armenuhy Gureghian de Scarinci, a quem dedico este trabalho, por ter me acompanhado em toda a minha trajetória, não apenas como estudante, mas, principalmente, como homem.

- A Ana Carolina Bertoletti pelo auxílio na redação deste trabalho e pela compreensão me dada nos momentos em que estive distante mesmo sem desejar, mas, principalmente, por nunca deixar de estar ao meu lado e amar, mesmo nos momento difíceis.

- Bolsistas de mestrado Carine Geltrudes Weber e Miguel Artur Feldens, que me auxiliaram na busca da bibliografia que serviu de embasamento conceitual deste trabalho; bem como pela amizade demonstrada no transcorrer destes anos.

- Gostaria de agradecer a todos os verdadeiros amigos que encontrei ao longo da minha vida, que, embora tenham sido poucos, certamente foram sinceros e dedicados.

- Finalmente, gostaria de agradecer àquelas pessoas que tentam obstaculisar-me, motivados por seu preconceito cego e pérfido, àqueles que sob o manto da solidariedade ocultam sentimentos vis, pois eles, com seus pensamentos e atitudes, fazem-me lutar com maior energia e convicção em busca de meus objetivos.

Sumário

Lista de Abreviaturas **08**

Lista de Figuras **09**

Lista de Tabelas **10**

Resumo **11**

Abstract **13**

1 Introdução **14**

1.1 Trabalho Desenvolvido 15
1.2 Estrutura do Trabalho 17

2 Contextualização **19**

2.1 Problemática e Motivação 19
2.1.1 Quantidade 19
2.1.2 Diversidade 20
2.1.3 Validade 20
2.1.4 Caches 21
2.2 Extração de Informações 22
2.3 Sistemas e Conceitos Relacionados 23
2.3.1 Archie 24
2.3.2 WAIS 25
2.3.3 Essence 27
2.3.3.1 Classificação *28*
2.3.3.2 Sumarização *28*
2.3.3.3 Estrutura de Arquivos Agrupados *32*
2.3.3.4 Exportação de Índices *32*
2.3.4 *Recall* e *Precision* 33
2.3.5 Extração do Conteúdo Semântico 34
2.4 Tratamento da Informação Temporal 36
2.4.1 Informação temporal explícita 36
2.4.2 Informação temporal implícita 37
2.4.3 Informação temporal deduzida 38
2.5 Manutenção da Informação Temporal 38
2.6 *Knowledge Discovery* 39
2.6.1 Descoberta de Conhecimento em Bases de Dados Temporais 40
2.7 Objetivos e Solução 40
2.8 Restrições e Limitações 43

3 Processos de Extração 45

3.1 Processo Estatístico 45
3.2 Processo Sintático 46
3.2.1 Uso de Informações de Dicionário 49
3.2.2 Construção de Bases de Conhecimento 49
3.3 Sistemas Especialistas e Baseados em Conhecimento 50
3.3.1 Representação do Conhecimento 52
3.3.1.1 Conhecimento Relacional Simples 52
3.3.1.2 Conhecimento Procedimental 53

4 Extração Semântica 55

4.1 Sistema Baseado em Conhecimento 56
4.1.1 Máquina de Inferência 58
4.1.2 Eventos de Entrada 58
4.1.3 Conclusão 58
4.1.4 Base de Conhecimento 59
4.2 Extração de Informações dos Arquivos de Entrada 59
4.2.1 Características Léxicas 59
4.2.2 Características Sintáticas e Relações de Termos 60
4.2.3 Dicionário de Palavras do Sistema 62
4.2.4 Classificação do Arquivo de Entrada 62
4.2.4.1 Base de Conhecimento Auxiliar 63
4.2.4.2 Pré Processamento de Arquivos de Entrada 63
4.3 Regras de Extração 64
4.3.1 Granularidade de Pesquisa 65
4.3.2 Condições 66
4.3.2.1 Condições de Pesquisa 66
4.3.2.2 Condições de Verificação 68
4.3.3 Ações 71
4.3.3.1 Funções de Deslocamento 72
4.3.3.2 Funções de Extração 75
4.4 Arquivo de Saída 77

5 Configuração da Extração 79

5.1 Sistema Aberto 79
5.2 Realimentação 79
5.3 Interface 81
5.3.1 Conceito 81
5.3.2 Características Desejáveis 82
5.3.3 Windows 95 83
5.3.4 *Parser* de Entrada 85

6 SES - Sistema de Extração Semântica 86

6.1 Arquitetura 86
6.1.1 Arquivos de Entrada 87
6.1.2 Arquivo de Saída 87
6.1.3 Extrator Semântico 87
6.1.4 Classificador 87
6.1.5 Base de Conhecimento 88
6.1.6 Base de Conhecimento Auxiliar 88
6.1.7 Configurador 88
6.1.8 Usuário 89
6.2 Visão Geral 89
6.3 Implementação do Protótipo 90
6.3.1 Módulos do Sistema 92
6.3.1.1 Módulo de Extração 92
6.3.1.2 Módulo de Classificação 94
6.3.1.3 Módulo de Configuração 94
6.3.2 Bases de Conhecimento 95
6.3.3 Ferramenta de Programação 95

7 Usando o SES 97

7.1 Instalação 97
7.2 Executando o Configurador 97
7.2.1 Ambiente do Configurador 98
7.2.2 Abrindo uma Base de Conhecimento 99
7.2.3 Criando uma Nova Base de Conhecimento 100
7.2.4 Grupos de Assuntos 101
7.2.5 Regras de Extração 102
7.2.5.1 Ficha de Regra 104
7.2.5.2 Condições de Pesquisa 105
7.2.5.3 Condições de Verificação 107
7.2.5.4 Ações 109
7.2.5.5 Separadores de Termos 112
7.2.6 Configurando Bases de Dados - Dicionários 114
7.2.6.1 Abrindo uma Base de Dados 114
7.2.6.2 Criando uma Nova Base de Dados 114
7.2.6.3 Tabela de Termos 115
7.2.7 Configurando Bases de Extensões - Bases de Conhecimento Auxiliares 116
7.2.7.1 Abrindo uma Base de Extensões 116
7.2.7.2 Criando uma Nova Base de Extensões 117
7.2.7.3 Tabela de Extensões 117
7.2.7.4 Ficha de Extensão 118
7.2.8 Relatórios 120
7.2.9 Formatos de Datas Pré-Definidos no SES 120
7.2.10 Saindo do Módulo Configurador 121
7.3 Extrator 121
7.3.1 Ambiente do Extrator 122
7.3.2 Chamando o Módulo Configurador Através do Extrator 123

7.3.3 Inicialização do Processo de Extração 123
7.3.4 Extraindo Informações 125
7.3.5 Arquivo de Saída 127
7.3.6 Arquivo de Log 128
7.3.7 Saindo do Módulo Extrator 128
7.4 Menu Editar 128
7.5 Menu Janela 129
7.6 Menu Ajuda 129

8 Estudo de Caso - Validação 130

8.1 Casos de Validação 130
8.1.1 Arquivos de *Mail* 130
8.1.1.1 Domínio dos Mails 130
8.1.1.2 Objetivo do uso dos Mails 130
8.1.1.3 Dados a serem Extraídos dos Mails 131
8.1.1.4 Base de Conhecimento Criada para Mails 131
8.1.2 Arquivos HTML 133
8.1.2.1 Domínio dos Arquivos HTML Utilizados 133
8.1.2.2 Objetivo do uso dos Arquivos HTML 133
8.1.2.3 Dados a serem Extraídos dos Arquivos HTML 134
8.1.2.4 Base de Conhecimento Criada para Arquivos HTML 134
8.1.3 Arquivos Fontes Clarion 136
8.1.3.1 Domínio dos Fontes 136
8.1.3.2 Objetivo do uso dos Fontes Clarion 136
8.1.3.3 Dados a serem Extraídos dos Fontes 136
8.1.3.4 Base de Conhecimento Criada para os Fontes Clarion 137
8.1.4 Grupo de Arquivos Distintos 137
8.1.4.1 Domínio dos Grupos de Arquivos 137
8.1.4.2 Objetivos do uso do Grupo de Arquivos 138
8.1.4.3 Dados a serem Extraídos do Grupo de Arquivos 138
8.1.4.4 Base de Conhecimento para o Grupo de Arquivos 138
8.2 Ambiente de Validação 138
8.3 Resultados dos Testes 138
8.4 Considerações sobre o Processo de Validação 142

9 Conclusão 144

9.1 Considerações Finais 144
9.2 Trabalhos Futuros 145

Anexo 1 Arquivos de entrada e saída 148

Bibliografia 159

Lista de Abreviaturas

ASCII	*American Standard Code for Information Interchange*
Adj	Adjetivo
BD	Bases de Dados
BDNE/SE	Bases de Dados Não Estruturadas e/ou Semi-Estruturadas
CP	Condição de Pesquisa
CV	Condição de Verificação
E-Mail	*Eletronic Mail*
FTP	*File Transfer Protocol*
HTML	*Hypertext Markup Language*
HTTP	*Hypertext Transfer Protocol*
IA	Inteligência Artificial
IBM	*International Bussiness Machines*
IE	*Information Extraction*
IR	*Information Retrieval*
Kb	Kilo Bytes
KDD	*Knowledge Database Discovery*
Mb	Mega Bytes
MS-DOS	Microsoft - *Disk Operating System*
PronPess	Pronome Pessoal
RAM	*Random Acess Memory*
SBC	Sistema Baseado em Conhecimento
SES	Sistema de Extração Semântica
SN	Sintagma Nominal
SV	Sintagma Verbal
STAIRS	*Storage and Infomation Retrieval System*
Subst	Substantivo
UCLA	*University of California*
Vi	Verbo Intransitivo
Vtd	Verbo Transitivo Direto
WAIS	*Wide Area Information Server*
WWW	*World-Wide Web*
WYSIWYG	*What You See Is What You Get*

Lista de Figuras

FIGURA 1.1 - Processo de Manipulação Semântica de Informações 16
FIGURA 2.1 - *Wide Area Information Server* 27
FIGURA 2.2 - Essence 29
FIGURA 2.3 - *Recall e Precision* 34
FIGURA 2.4 - Relação entre *Recall* e *Precision* 34
FIGURA 2.5 - Objetivo de Extração 41
FIGURA 3.1 - Gramática 47
FIGURA 3.2 - Árvore Gramatical 48
FIGURA 3.3 - Regra de Produção 54
FIGURA 4.1 - Arquitetura Genérica de um Sistema Baseado em Conhecimento 56
FIGURA 4.2 - Deslocamento Restrito à Regra 73
FIGURA 4.3 - Deslocamento Global 73
FIGURA 4.4 - Estrutura do Arquivo de Saída 78
FIGURA 4.5 - Arquivo de Saída Exemplo para *Mails* 78
FIGURA 5.1 - Típico Programa Windows 84
FIGURA 6.1 - Arquitetura do Sistema de Extração Semântica 86
FIGURA 6.2 - Protótipo do Sistema de Extração Semântica 91
FIGURA 6.3 - Hierarquia Organizacional do SES 92
FIGURA 6.4 - Esquema da Base de Dados do SES 95
FIGURA 7.1 - Ícone do Configurador 98
FIGURA 7.2 - Ambiente de Trabalho do Configurador 99
FIGURA 7.3 - Abrindo uma Base de Conhecimento 100
FIGURA 7.4 - Salvando uma Nova Base de Conhecimento 101
FIGURA 7.5 - Tabela de Grupos 102
FIGURA 7.6 - Tabela de Regras 103
FIGURA 7.7 - Ficha de Regra 104
FIGURA 7.8 - Tabela de Ações de Então 105
FIGURA 7.9 - Formulário de Condição de Pesquisa 107
FIGURA 7.10 - Formulário de Condição de Verificação 109
FIGURA 7.11 - Formulário de Ação 112
FIGURA 7.12 - Tabela de Separadores 113
FIGURA 7.13 - Tabela de Termos 115
FIGURA 7.14 - Tabela de Extensões 118
FIGURA 7.15 - Ficha de Extensão 119
FIGURA 7.16 - Ícone do Extrator 121
FIGURA 7.17 - Ambiente de Trabalho do Configurador 122
FIGURA 7.18 - Ficha de Inicialização 124
FIGURA 7.19 - Janela de Controle de Extração 126
FIGURA 7.20 - Visualizador do Arquivo de Saída 127
FIGURA 8.1 - Gráfico de Tempo de Extração 141
FIGURA 8.2 - Gráfico de Velocidade (bytes/segundo) 142

Lista de Tabelas

TABELA 2.1 - Sumarizadores 30
TABELA 3.1 - Conhecimento Relacional Simples 53
TABELA 4.1 - Associação entre Extensões de Nomes de Arquivos e Bases de Conhecimento 64
TABELA 8.1 - Tabela de Volume de Dados Processados 140
TABELA 8.2 - Tabela de Tempos de Processamento 140
TABELA 8.3 - Tabela de Razão de Volume por Tempo (velocidade) 140
TABELA 8.4 - Tabela de Velocidade por Regra 141
TABELA 8.5 - Resultados para Noventa Arquivos 141

Resumo

Entre as áreas que mais se desenvolvem na informática nos últimos anos estão aquelas relacionadas ao crescimento da rede Internet, que interliga milhões de usuários de todo o mundo. Esta rede disponibiliza aos usuários uma a enorme variedade e quantidade de informações, principalmente dados armazenados de forma não estruturada ou semi estruturada. Contudo, tal volume e heterogeneidade acaba dificultando a manipulação dos dados recuperados a partir da Internet. Este problema motivou o desenvolvimento deste trabalho.

Mesmo com o auxílio de várias ferramentas de pesquisa na Internet, buscando realizar pesquisas sobre assuntos específicos, o usuário ainda tem que manipular em seu computador pessoal uma grande quantidade de informação, pois estas ferramentas não realizam um processo de seleção detalhado. Ou seja, são recuperados muitos dados não interessantes ao usuário. Existe, também, uma grande diversidade de assuntos e padrões de transferência e armazenamento da informação, criando os mais heterogêneos ambientes de pesquisa e consulta de dados. Esta heterogeneidade faz com que o usuário da rede deva conhecer todo um conjunto de padrões e ferramentas a fim de obter a informação desejada.

No entanto, a maior dificuldade de manipulação esta ligada aos formatos de armazenamento não estruturados ou pouco estruturados, como, por exemplo: arquivos textos, *Mails* (correspondência eletrônica) e artigos de *News* (jornais eletrônicos). Nestes formatos, o entendimento do documento exige a leitura do mesmo pelo usuário, o que muitas vezes acarreta em um gasto de tempo desnecessário, pois o documento, por exemplo, pode não ser de interesse deste ou, então, ser de interesse, mas sua leitura completa só seria útil posteriormente.

Várias informações, como chamadas de trabalhos para congressos, preços de produtos e estatísticas econômicas, entre outras, apresentam validade temporal. Outras informações são atualizadas periodicamente. Muitas dessas características temporais são explícitas, outras estão implícitas no meio de outros tipos de dados. Isto torna muito difícil a recuperação de tal tipo de informação, gerando, várias vezes, a utilização de informações desatualizadas, ou a perda de oportunidades.

Desta forma, o grande volume de dados em arquivos pessoais obtidos a partir da Internet criou uma complexa tarefa de gerenciamento dos mesmos em conseqüência da natureza não estruturada dos documentos recuperados e da complexidade da análise do tempo de validade inerente a estes dados.

Com o objetivo de satisfazer as necessidades de seleção e conseqüente manipulação das informações existentes a nível local (computador pessoal), neste trabalho, é descrito um sistema para extração e sumarização destes dados, utilizando conceitos de IE (*Information Extraction*) e Sistemas Baseados em Conhecimento. Os dados processados são parcialmente estruturados ou não estruturados, sendo manipulados por um extrator configurado a partir de bases de conhecimento geradas pelo usuário do sistema.

O objetivo final desta dissertação é a implementação do Sistema de Extração Semântica de Informações, o qual permite a classificação dos dados extraídos em classes significativas para o usuário e a determinação da validade temporal destes dados a partir da geração de uma base de dados estruturada.

Palavras-Chave: Gerenciamento de Informações, Validade Temporal de Informações, Bancos de Dados Temporais, Sistemas Baseados em Conhecimento, Descoberta de Conhecimento em Bases de Dados (*Knowledge Database Discovery* - KDD), Extração de Informações (*Information Extraction* - IE), Recuperação de Informações (*Information Retrieval* - IR).

TITLE: "System of Semantic Extraction of Information"

Abstract

One of the most challenging area in Computer Science is related to Internet technology. This network offers to the users a large variety and amount of information, mainly, data storage in unstructured or semi-structured formats. However, the vast data volume and heterogeneity transforms the retrieved data manipulation a very arduous work. This problem was the prime motivation of this work.

As with many tools for data retrieval and specific searching, the user has to manipulate in his personal computer an increasing amount of information, because these tools do not realize a precise data selection process. Many retrieval data are not interesting for the user. There are, also, a big diversity of subjects and standards in information transmission and storage, creating the most heterogeneous environments in data searching and retrieval. Due to this heterogeneity, the user has to know many data standards and searching tools to obtain the requested information.

However, the fundamental problem for data manipulation is the partially or fully unstructured data formats, as text, mail and news data structures. For files in these formats, the user has to read each of the files to filter the relevant information, originating a loss of time, because the document could be not interesting for the user, or if it is interesting, its complete reading may be unnecessary at the moment.

Some information as call-for-papers, product prices, economic statistics and others, has associated a temporal validity. Other information are updated periodically. Some of these temporal characteristics are explicit, others are implicitly embedded in other data types. As it is very difficult to retrieve the temporal data automatically, which generate, many times, the use of invalid information, as a result, some opportunities are lost.

On this paper a system for extraction and summarizing of data is described. The main objective is to satisfy the user's selection needs and consequently information manipulation stored in a personal computer. To achieve this goal we are employed the concepts of Information Extraction (IE) and Knowledge Based Systems. The input data manipulation is done by an extraction procedure configured by a user who defined knowledge base.

The objective of this paper is to develop a System of Semantic Extraction of Information which classifies the data extracted in meaningful classes for the user and to deduce the temporal validity of this data. This goal was achieved by the generation of a structured temporal data base.

Key-words: Information Management, Information Validate, Temporal Databases, Knowledge Based Systems, Knowledge Database Discovery (KDD), Information Extraction - IE, Information Retrieval - IR.

1 Introdução

A análise do desenvolvimento da tecnologia de processamento de informações, segundo uma perspectiva histórica, mostra-nos que a informática, desde de seu surgimento no final da década de quarenta, até os dias de hoje, foi uma das áreas do conhecimento humano que mais se desenvolveu. Equipamentos que antes tomavam uma sala, hoje são substituídos por outros que podem ser colocados sobre uma pequena mesa. Tecnologias que anteriormente tinham um custo de milhões de dólares, ficando restritas a grandes empresas, sofreram uma drástica redução de seus custos, o que permite até mesmo a uma pessoa física adquirir e manter um computador pessoal em sua residência. Velocidade de processamento, bem como a capacidade e performance das memórias e dos meios de armazenamento secundários, constituem-se em barreiras que a todo o momento são superadas.

Como uma conseqüência direta desta nova realidade esboçada acima, verificou-se uma rápida difusão da tecnologia de informática em todos os setores da vida social e econômica, atingindo indistintamente universidades, escolas, empresas, etc. Adicionalmente, verificou-se uma modificação no perfil das pessoas que utilizam o computador. Assim, um público que anteriormente era constituído apenas por especialistas (engenheiros, físicos, matemáticos, etc.), passou a ser integrado por usuários não intimamente ligados a área de informática.

Outro setor que nas últimas décadas, ao lado da informática, experimentou um grande desenvolvimento e expansão foi a tecnologia (*hardware* e *software*) de redes de computadores, atingindo tanto as redes locais e metropolitanas, como as redes de longa distância.

Este fato fez com que a visão de conceber os computadores como máquinas isoladas umas das outras, constitua-se hoje em um conceito há muito superado. Desta forma, os computadores são interligados por redes locais, metropolitanas ou de longa distância, constituindo-se em usuários e provedores de serviços, cada vez mais complexos e sofisticados.

Um ambiente real, onde podemos observar a aplicação prática dos conceitos e avanços acima descritos é a rede Internet, também conhecida como "a rede". Esta rede criada no início da década de setenta, tem se expandido enormemente nos últimos anos, congregando atualmente milhões de usuários e cerca de 13 milhões de servidores em todo o mundo. Em média, a Internet dobra de tamanho aproximadamente a cada intervalo de doze a quinze meses. A perspectiva é de que o número de servidores deverá ultrapassar os 100 milhões até o ano 2000 [DVO 96].

Por outro lado, o mais importante aspecto a ser enfatizado a respeito da Internet não é o número de usuários, mas sim a enorme variedade e quantidade de informações e serviços, a eles disponibilizados por esta rede. Centenas de organizações, tais como universidades, institutos de pesquisa e empresas, entre outras, colocam em certos computadores conectados à Internet, chamados de servidores, documentos, artigos e

programas de diversos tipos e áreas. Desta maneira, o público em geral pode ter acesso a este conhecimento, antes mesmo que este seja apresentado em um congresso ou publicado em um periódico.

Esta diversidade e quantidade de informação é manipulada atualmente por um conjunto de ferramentas que ajudam as pessoas a procurar, localizar e recuperar informações, tais como o GOPHER, o WAIS (*Wide Area Information Server*) e o AltaVista, que permitem acessar bancos de dados de todas as partes da Internet [WYA 95]. Esta manipulação é realizada a um nível macro, ou seja, a nível de rede, selecionando, dentre a grande variedade e volume de assuntos, um subconjunto dos dados oferecidos. No entanto, tais dados selecionados e recuperados ainda se mantêm volumosos para tratamento a nível micro, ou seja, a nível local.

Neste sentido, existe muita dificuldade no manuseio das informações armazenadas em computadores pessoais (nível local). Isto ocorre em vista das ferramentas de seleção e do próprio volume de informação existente a nível macro. Tais características não permitem uma seleção mais restritiva dos dados armazenados na Internet sem perda de informações importantes, acarretando na recuperação de grandes volumes de dados, muitos destes sem interesse para o usuário. Além disso, a análise exaustiva e empírica das informações recuperadas normalmente é impossível de ser realizada sobre megabytes ou até gigabytes de dados existentes nas bases de dados selecionadas [SHA 94].

Adicionalmente, é importante salientar a diversidade de formatos de armazenamento e a dinâmica das informações disponíveis, consistindo principalmente em arquivos texto com estruturas internas definidas ou não: páginas html (*Hipertext Markup Language*), correspondências eletrônicas, arquivos *PostScript* e textos livres [COW 96].

Neste ambiente, é muito difícil e praticamente impossível manter o controle da validade temporal dos dados disponíveis. Várias informações como preços, estatísticas, convites, etc. são atualizadas periodicamente, fazendo com que muitas vezes dados obsoletos sejam utilizados. O resultado é a perda de oportunidades como, por exemplo, chamadas de artigos para congressos, onde o prazo limite de envio do artigo pode não ser percebido, ou a mensagem é lida em uma data posterior a data limite.

1.1 Trabalho Desenvolvido

Tendo em vista o contexto apresentado acima, observa-se que uma solução tecnológica é necessária a fim de responder ao problema de manipulação dos dados armazenados a nível local, levando em conta o volume, a diversidade e a validade das informações guardadas em um computador pessoal.

Com o objetivo de satisfazer as necessidades de seleção e conseqüente manipulação das informações existentes a nível local, propõe-se nesta dissertação um sistema de extração semântica de informações em bases de dados não estruturadas e semi-estruturadas (BDNE/SE), as quais formam grande parte do conjunto de dados contidos na Internet. Para tal, serão utilizadas técnicas de Sistemas Baseados em Conhecimento e Extração de Informações (IE - *Information Extraction*).

Mais detalhadamente, a solução proposta envolve duas fases: a extração das informações dos arquivos pessoais e a posterior dedução temporal a partir dos dados estruturados obtidos (FIGURA 1.1). Contudo, nesta dissertação será realizado somente o estudo e o desenvolvimento do módulo extrator das informações (primeira fase), ficando a segunda fase para um trabalho futuro.

Os arquivos pessoais não estruturados e semi-estruturados no formato ASCII são o ponto de partida do processo de manipulação de informações. O extrator de informações, através da leitura dos arquivos de entrada, extrai destes os dados relevantes ao processo de manipulação. Cabe salientar que, neste trabalho, dar-se-á destaque às características temporais das informações, visando a avaliação da validade temporal dos dados armazenados. Como resultado deste processo, os dados extraídos irão formar uma base de dados estruturada.

A extração de informações é um procedimento semântico dependente da identificação do tipo do arquivo de entrada. Neste sentido, um prévio trabalho de análise semântica do arquivo pessoal de entrada é realizado, explorando a extensão do nome do arquivo para determinar a forma de extração mais apropriada das informações. Esta técnica tem sido utilizada no sistema de recuperação de informações Essence [HAR 93][HAR 95], onde sumarizadores e procedimentos especiais são utilizados para cada diferente tipo de arquivo, gerando os índices de pesquisa.

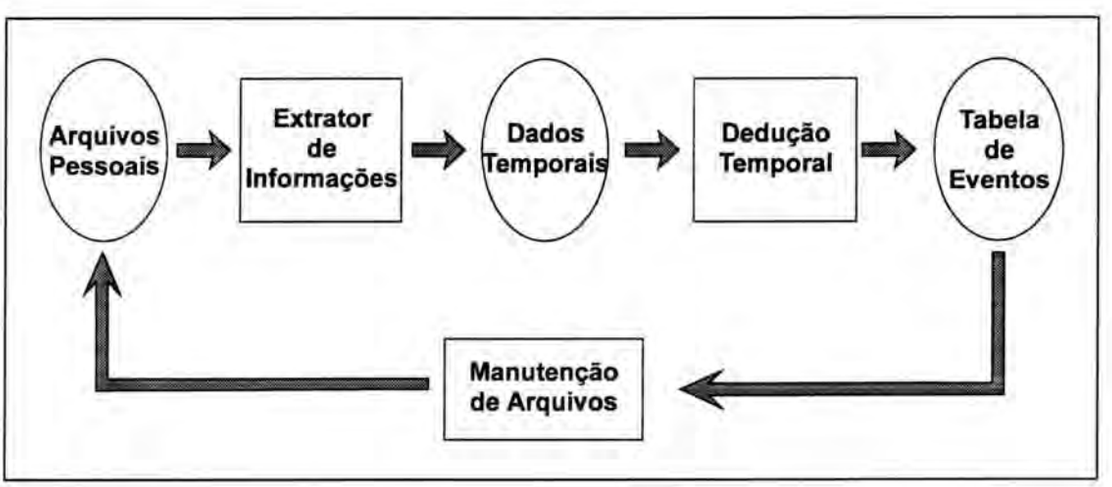

FIGURA 1.1 - Processo de Manipulação Semântica de Informações

A avaliação do conteúdo temporal, incluindo validade, requer o uso da base de dados temporal gerada pelo processo extrator. É a partir dos dados contidos nesta base é que será realizada a manipulação dos arquivos pessoais de forma automática. É importante salientar que a informação temporal está presente nos arquivos pessoais, mas para ser utilizada deve ser encontrada, processada e organizada a fim de tornar-se manipulável. Desta forma, a complexidade do processo de dedução temporal é reduzida, trabalhando sobre uma simples estrutura de dados com um domínio delimitado.

Na base de dados resultante do processo de extração, cada informação definida deve ser associada a um tempo de transação (o instante em que a informação foi

atualizada) e a um período de validade (correspondendo ao início e o fim da validade da informação) [SNO 85]. As modelagens de dados tradicionais enxergam o tempo como um atributo, uma melhor opção é estender o modelo de dados para incluir um comportamento temporal, como no modelo T-FORM [EDE 93] [EDE 94a]. Dentro desta classe de modelo, todas as informações de validade temporal devem ser transformadas em conteúdo temporal. Planeja-se utilizar o modelo temporal desenvolvido em T-FORM, em particular a lógica temporal de primeira ordem para a representação de regras de dedução temporal.

A dedução temporal realizada para a ativação dos eventos para a manutenção dos arquivos pessoais é um típico problema de Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases* - KDD). Segundo [FRA 92], a área de KDD realiza a extração não trivial de informações a partir de dados implícitos, previamente desconhecidos e potencialmente utilizáveis. Tradicionalmente sistemas de KDD não são bem aplicados para este problema particular, ou seja, sobre arquivos pessoais onde as estruturas de armazenamento são muito irregulares. Por outro lado, normalmente as aplicações de KDD tratam um grande volume de dados, como é o caso de bases de dados locais recuperadas a partir da Internet, mas estruturados como um banco de dados relacional, por exemplo. Isto justifica a importância do extrator dentro do processo de manipulação aqui descrito, onde muitos dados estão na forma de textos.

Finalmente, a manutenção dos arquivos pessoais é realizada baseada em uma tabela de eventos acionados por deduções realizadas sobre a base temporal. O usuário poderá separar entre seus arquivos pessoais quais são interessantes, movendo-os para um determinado diretório, e quais não são, apagando-os pois a informações ali armazenada já perderam a validade, por exemplo. Neste momento, alguns alertas são gerados para o usuário como aviso das operações sendo realizadas, inclusive confirmações dos arquivos que o sistema pretende apagar.

O processamento de manutenção dos arquivos é uma atividade administrativa realizada em vários momentos arbitrários. O sistema guarda informações de todas as modificações nos arquivos pessoais e, periodicamente, executa uma varredura nos arquivos criados ou modificados após o último processamento, passando-os pelo extrator de informações.

1.2 Estrutura do Trabalho

Visando abordar de maneira lógica e seqüencial todos os conceitos importantes inerentes ao trabalho aqui descrito, estruturou-se esta dissertação da forma que se segue.

No capítulo 2, o tema abordado na dissertação é contextualizado dentro da área de Extração de Informações e outras áreas da Ciência da Computação correlacionadas com este trabalho. Além disso, é apresentada a motivação que levou ao desenvolvimento deste estudo e seus objetivos.

O capítulo 3 apresenta o embasamento conceitual do sistema proposto nesta dissertação e, conseqüentemente, os estudos realizados para o desenvolvimento deste.

No capítulo 4, realiza-se a ligação entre o embasamento conceitual, apresentado no capítulo anterior, e os objetivos deste trabalho (capítulo 2). Isto demonstrará como os objetivos descritos serão atingidos, resultando no método de extração de informações proposto. Finalmente, descreve-se as regras de extração semântica do sistema definido.

Como parte complementar, mas não mesmo importante, o capítulo 5 descreve o módulo de configuração das regras de extração de informações. Além disso, sendo este o módulo de maior interação com o usuário do sistema, é apresentado um estudo sobre interfaces, ou seja, o embasamento conceitual da interface escolhida para o sistema de extração semântica de informações.

O capítulo 6 apresenta a arquitetura do sistema de extração semântica proposto, descrevendo cada um dos seus módulos. Adicionalmente, é detalhada a estrutura do protótipo implementado.

No capítulo 7, é redigido um pequeno manual introdutório ao uso do sistema de extração semântica, destinado a auxiliar no uso deste programa.

A validação do protótipo desenvolvido, bem como de todo o estudo apresentado nesta dissertação, é colocada no capítulo 8.

Finalmente, no capítulo 9 teremos alguns comentários a respeito do trabalho desenvolvido nesta dissertação, procurando destacar as perspectivas de trabalhos e pesquisas futuras por ele abertas.

Como anexo são apresentados alguns dos arquivos de entrada utilizados no processo de validação do sistema proposto, bem como os respectivos resultados de extração (arquivos de saída).

2 Contextualização

A fim de atingir os objetivos finais desta dissertação e do desenvolvimento do protótipo relacionado, realizou-se o estudo da área da ciência da computação em cujo este trabalho se insere, bem como de áreas co-relacionadas. Além disso, é importante destacar a análise de outros sistemas e trabalhos relacionados com esta dissertação e que influenciaram seu andamento. Contudo, inicialmente, cabe salientar os fatos que levaram a criação deste estudo, motivando-o.

Desta forma, neste segundo capítulo, apresenta-se a realidade onde esta inserido este trabalho, sua motivação e seus objetivos.

2.1 Problemática e Motivação

Não obstante às vantagens desta divulgação do conhecimento gerada pelas redes de computadores de longa distância, nos deparamos com um conjunto de problemas gerados exatamente pela quantidade e diversidade de informações. Foram tais problemas que motivaram o desenvolvimento deste trabalho, sendo descritos mais detalhadamente a seguir.

2.1.1 Quantidade

Existe uma grande quantidade de informação livre na Internet, a qual está crescendo rapidamente [BOW 94]. Por esta razão, foram desenvolvidas várias ferramentas que ajudam as pessoas a procurar, localizar e recuperar informações, tais como o GOPHER, o WAIS (*Wide Area Information Server*), o Alta Vista e o Yahoo, que permitem acessar bancos de dados de todas as partes da Internet [WYA 95]. Contudo, o enorme volume de informação disponível na rede faz com que o resultado de uma pesquisa com o uso de tais ferramentas seja composto por um grande conjunto de documentos.

Na realidade, ferramentas como GOPHER, WAIS e Alta Vista permitem que o usuário informe uma ou mais palavras-chave, e este procurará documentos ou textos que, de fato, contenham estas palavras [FER 94]. Cabe observar, no entanto, que muitos documentos podem conter tais palavras-chave, mesmo alguns que não sejam de interesse do usuário e que serão recuperados.

Neste sentido, mesmo com o auxílio de várias ferramentas de pesquisa na Internet, buscando realizar pesquisas sobre assuntos específicos, o usuário ainda tem que manipular em seu computador pessoal uma grande quantidade de informação. Isto ocorre em vista das ferramentas de seleção e do próprio volume de informação existente a nível macro, ou seja, a nível de rede. Tais características não permitem uma seletiva mais restritiva dos dados armazenados na Internet sem perda de informações importantes, acarretando na recuperação de grandes volumes de dados, muitos destes sem interesse para o usuário. Além disso, a quantidade de informação a nível local

também é elevada por arquivos trazidos por FTP (*File Transfer Protocol*), ou arquivos de E-Mail (*Eletronic Mail*).

2.1.2 Diversidade

Analogamente ao grande volume de dados, existe uma grande diversidade de assuntos e padrões de transferência e armazenamento da informação, criando os mais heterogêneos ambientes de pesquisa e consulta de dados [BOW 94]. Cada um desses ambientes visa oferecer um determinado conjunto de facilidades e vantagens ao usuário, como é claramente destacado no uso de ferramentas de consulta que utilizam o padrão HTTP (*Hypertext Transfer Protocol*), como, por exemplo, NetScape.

Por outro lado, esta variedade de formatos de dados tráz alguns problemas, pois o usuário deve conhecer todo um conjunto de padrões a fim de obter a informação desejada. Além disso, é necessário utilizar várias ferramentas de consulta, como, por exemplo, um visualizador como Mosaic, Netscape ou Internet Explorer, para arquivos em HTML (*Hypertext Markup Language*), e *MailTool* para arquivos de correio eletrônico. Esta variedade cria uma descentralização da consulta, pois não existe uma única ferramenta para manipular os diversos formatos de armazenamento dos dados sob análise. Pode-se ter, por exemplo, um usuário estudando sobre qualquer assunto de seu interesse. Esta pessoa poderá usar ferramentas como WAIS, Alta Vista ou Internet Explorer para encontrar as informações desejadas, sendo que os resultados das pesquisas com cada uma das ferramentas provavelmente será diferente. Com isso, a fim de realizar um estudo mais aprofundado e completo, o usuário deverá utilizar várias destas ferramentas de recuperação de informações na rede.

No entanto, a maior dificuldade de manipulação esta ligada aos formatos de armazenamento não estruturados ou pouco estruturados, como, por exemplo, arquivos de textos, *Mails* (correspondência eletrônica) e artigos de *News* (jornais eletrônicos). Nestes formatos, o entendimento do documento exige a leitura do mesmo pelo usuário, o que muitas vezes acarreta em um gasto de tempo desnecessário, pois o documento, por exemplo, pode não ser de interesse deste ou, então, ser de interesse, mas sua leitura completa só seria útil posteriormente. Além disso, a não existência de uma estrutura formal pré-definida, estimula a falta de padronização da informação dentro do documento, fazendo com que muitas informações sejam organizadas de acordo com o capricho individual de cada autor [HAR 93]. Desta forma, o pouco uso de padrões estruturais acaba dificultando o entendimento e a seleção dos dados disponíveis.

2.1.3 Validade

Várias informações, como Chamadas de Artigos para congressos (do inglês *Call-For-Papers*), preços de produtos e estatísticas econômicas, entre outras, apresentam validade temporal (período de validade da informação). Outras informações são atualizadas periodicamente.

Muitas dessas características temporais são explícitas, outras estão implícitas no meio de outros tipos de dados. Isto torna muito difícil a recuperação de tal tipo de informação, gerando, várias vezes, a utilização de informações desatualizadas, ou a

perda de oportunidades. Como exemplo, pode-se ter o caso de informações sobre chamadas de artigos para congressos, onde dados como a data limite de entrega do artigo para avaliação são de extrema importância, pois, se ultrapassados, geram a invalidação do artigo escrito. Neste caso, a informação temporal está colocada implicitamente no texto, sem uma localização padronizada que facilitaria sua recuperação.

No caso de um *mail*, por outro lado, a data a última modificação é colocada de forma explícita no cabeçalho do arquivo. No entanto, esta data só apresenta de forma consistente o tempo de transação, ou seja, o tempo no qual é feita a atualização do banco de dados [EDE 94]. O tempo de validade (período em que a informação armazenada é válida), contudo, pode ser inconsistente, pois não está explicitado e a informação do *mail* pode ter perdido a validade.

Neste sentido, observa-se que as características temporais são muito importantes para os dados que trafegam pela Internet. Além do grande e variado volume de informação, deve-se destacar a velocidade com que esta informação é atualizada. Há alguns anos atrás, a produção de artigos era mais lenta, permitindo que pesquisadores mantivessem arquivos com resumos e referências aos seus artigos de interesse. Atualmente, contudo, tal atividade é inviável, pois a velocidade de atualização dos dados cresceu proporcionalmente ao volume. Desta forma, o prazo de validade da informação tornou-se menor, exigindo freqüentes atualizações das referências e um maior dinamismo por parte do pesquisador, o que, na maioria das vezes, é impossível.

2.1.4 Caches

Por outro lado, além da alta velocidade de atualização, deve-se lembrar de alguns mecanismos de *cache* que são utilizados na Internet. *Caches* são estruturas locais que mantêm cópias de dados acessados repetidamente para evitar uma nova busca nos servidores, diminuindo o tempo de resposta e o tráfego na rede [BOW 94]. Neste sentido, foi realizado um intenso trabalho na área de servidores de arquivos e de sistemas operacionais para suportarem esta forma de replicação de dados [HOW 88]. A ferramenta WAIS (*Wide Area Information Server*), por exemplo, como diz seu nome, é um servidor de dados com índices temáticos sobre as informações disponíveis, armazenando as referências de forma centralizada, a dita *cache* [MAR 92].

Contudo, a sincronização entre o conteúdo das *caches* e os dados originais é algo que também gera um grande *overhead* de tráfego de comunicação. Sendo assim, muitas das referências, a fim de redução de transmissões, ficam desatualizadas e sem validade. Desta forma, o usuário acaba, várias vezes, manipulando sem necessidade um conjunto inválido de informações.

Além disso, o volume de dados armazenado localmente nas estações de trabalho é muito grande. Estes dados locais podem ser considerados como *caches*, pois constituem-se em cópias de arquivos remotos. Os dados locais correspondem a instantâneos (*snapshots*) de dados na rede, perdendo a validade por apresentarem assincronia em relação aos dados originais.

2.2 Extração de Informações

Sistemas de extração de informações (*Information Extraction* - IE) analisam ilimitadamente o texto para extrair tipos específicos de informação. Este tipo de sistema é preparado para encontrar e relacionar informações relevantes a partir de documentos, enquanto ignora informações estranhas e irrelevantes [COW 96]. Sistemas IE não tentam entender todo o texto em todas as partes do documento de entrada, mas eles analisam porções de cada documento que contenha informações relevantes [LEH 96]. A relevância é determinada por um domínio pré-definido de normas que devem especificar, com tanta exatidão quanto for possível, que tipos de informação o sistema espera encontrar.

Uma maneira de pensar em Extração de Informações é em termos de construção de bases de dados. Um sistema IE tenta converter textos não estruturados em entradas estruturadas para uma base de dados [OVE 96]. Entradas na base de dados podem ser geradas a partir de um conjunto fixo de valores, ou de cadeias de caracteres (*substrings*), retirados do texto original. Para cada tipo de texto (assunto, língua, etc.) sob extração por um sistema de IE, uma base de dados de saída tem de ser criada, a qual especifica o número de campos de dados que podem ser preenchidos com os dados extraídos. Além disso, deve-se determinar quais informações extraídas serão armazenadas em cada um destes campos [LEH 96].

Por exemplo, considere um sistema que aplique processos de extração de informações para textos sobre terrorismo. Pode-se definir, a priori, vinte e quatro possíveis atividades terroristas a serem pesquisadas nos textos de entrada. Então um campo para cada tipo de evento terrorista será definido na base de dados estruturada de saída, devendo cada um ser preenchido com uma destas vinte e quatro possíveis atividades. Estes campos serão carregados com *strings* do texto fonte, onde cada *string* apresenta uma informação classificada em uma destas atividades.

Na perspectiva do processamento de linguagem, os sistemas IE devem operar em muitos níveis, do reconhecimento da palavra à análise sintática, e de um entendimento a nível de sentença até o nível de análise de discurso de um documento inteiro. A cobertura do dicionário de palavras utilizado por alguns sistemas de extração; ou seja, a quantidade de termos que estes contém, é em geral um problema especial e um desafio desde o início até o fim de um documento, que pode conter termos de todos os tipos: abreviaturas, nomes próprios, tipos não mencionados no dicionário e estilos de escrita telegráficos em documentos como anotações [COW 96]. Isto dificulta a construção de um sistema de extração de informações operacional.

O número de sistemas de IE desenvolvidos e em operação está em perspectivas de crescimento dramático na próxima década. Atualmente, existe mais sistemas de extração de informações em laboratórios de pesquisa do que em operação diária. Mas o interesse nesta tecnologia é forte, e a comunidade de pesquisadores está crescendo. Pesquisadores têm investigado e descoberto um número cada vez maior de casos de aplicação para este tipo de sistema. Alguns exemplos são apresentados a seguir, segundo [LEH 96]:

- **Informações para Cuidados com a Saúde**

Sistemas de extração de informações têm sido projetados para resumir registros de pacientes médicos, extraindo diagnósticos, sintomas, características físicas, resultados de testes e tratamentos terapêuticos. Estes sistemas podem ser usados para auxiliar nos cuidados com a saúde a partir de estudos de qualidade de saúde, ou para suporte a procedimentos médicos seguros, onde cada paciente apresenta determinados sintomas, devendo-se encontrar informações históricas de outros casos a partir de características sintomáticas semelhantes.

- **Monitoramento técnico/cientifico literário**

Sistemas de extração de informações têm sido projetados para monitorar artigos técnicos, descrevendo a fabricação de *chips* microeletrônicos, capturando especificamente informações sobre quatro procedimentos tecnológicos: *layering*, *lithography*, *etching* e *packaging*. Avançados relatos de manufatura, venda e distribuição de equipamentos, e sobre o desenvolvimento ou o uso de procedimentos tecnológicos fazem parte do domínio de extração.

- **Pesquisa Inteligente**

Jornais e transcrições de textos de rádio e televisão têm sido analisados por sistemas de extração de informações com o objetivo de monitorar descrições de atividades terroristas. Estes sistemas aplicações específicas de IE, sendo utilizados para identificar o tipo de evento terrorista, identificando os autores suspeitos, as vitimas, e os danos causados, bem com o momento e o local do evento em questão.

- **Competição Inteligente**

Artigos de jornais têm sido analisados para encontrar casos de corporações agrupadas (cartéis) e *joint ventures*. Um sistema de extração de informações neste domínio é usado para identificar os participantes das corporações, bem como seus produtos e serviços, além de outros detalhes como investimentos de capital.

Novas aplicações para a tecnologia de IE são descobertas todos os dias. Não é difícil encontrar uma aplicação em potencial. No entanto, considera-se difícil determinar, no corrente estado da arte em extração de informações, como tornar factível as aplicações possíveis.

2.3 Sistemas e Conceitos Relacionados

Nas três subseções a seguir, procuramos familiarizar o leitor com alguns sistemas de recuperação de informações muito utilizados na Internet.

Tais sistemas trabalham na recuperação de informações a nível macro; contudo, conhecimentos da área de Recuperação de Informações (IR) são parte da base conceitual deste trabalho por suas características e técnicas de extração muito parecidas com as de sistemas de Extração de Informações. Na realidade, sistemas de IR, muitas vezes, utilizam técnicas de IE para a geração de índices de pesquisa baseados nos dados mais

relevantes de cada texto [LEW 96]. Assim sendo, nas duas áreas da ciência da computação existe a importante tarefa de caracterização do conteúdo central do arquivo bruto (textos originais) a ser processado.

Neste sentido, é quanto a esta caracterização ou extração do conteúdo central de textos que o trabalho proposto mais se assemelha com os sistemas a seguir descritos. Por esta razão, dar-se-á maior destaque a este ponto em específico a seguir.

Contudo, além do fato de sistemas de IR usarem técnicas da área de Extração de Informações, é importante salientar que sistemas destas duas áreas trabalham muitas vezes em conjunto. Sistemas de IR trabalham sobre o grande volume de informações existente dentro da Internet, distribuído em milhares de computadores em dezenas de países, comprovando a problemática apresentada anteriormente. Estes sistemas recuperam dados utilizáveis a partir de um vasto campo de material bruto [WEB 96].

A partir do uso de um sistema de recuperação de informações, e com um grande conjunto de informações recuperadas em mãos, um sistema de IE pode então transformar o material bruto recuperado, refinando-o e reduzindo-o a uma pequena parte do texto original. Sistemas de extração de informações iniciam seu processamento com uma coleção destes textos brutos, então transformando-os em informações que podem ser mais facilmente e rapidamente digeridas e analisadas. Estes sistemas isolam os fragmentos de textos relevantes, extraindo a informação relevante de tais fragmentos, e, após, os pedaços são organizados em uma informação coerente [COW 96].

Desta forma, cabe destacar que pressupõe-se que muitas das informações a serem tratadas pelo protótipo proposto nesta dissertação sejam resultantes da utilização de diversos sistemas de pesquisa na rede, ou seja, sistemas de recuperação de informações.

2.3.1 Archie

O Archie é um sistema desenvolvido pela universidade McGill no Canadá, a partir de 1991. Seu objetivo é fornecer aos usuários da Internet uma forma de identificar os servidores de FTP que guardam as informações por eles desejadas. Pela última contagem, existiam 2.500.000 arquivos indexados de mais de 1.000 locais FTP, processando 100.000 consultas por dia [WYA 95].

Este sistema de pesquisa provê um serviço de localização de servidores FTP públicos e seus arquivos ou diretórios, capazes de satisfazer uma consulta formulada por um determinado usuário do sistema.

Assim como muitas outras ferramentas existentes na Internet, o Archie apresenta uma arquitetura cliente/servidor. O servidor Archie gerencia um índice de dados onde estão registrados todos os serviços FTP públicos conhecidos pelo administrador do sistema, bem como o nome de todos os arquivos e diretórios presentes nos mesmos. A atualização total deste índice é realizada automaticamente pelo sistema, normalmente uma vez por mês [EMT 92]. Desta forma, cada dia são pesquisados aproximadamente 1/30 do total de "sítios" cadastrados. Contudo, no caso de "sítios" FTP cuja a validade

das informações é muito curta, esta atualização pode ser realizada quinzenalmente. O servidor contata todos os servidores FTP públicos por ele conhecidos, recuperando os nomes de todos os arquivos e diretórios neles contidos.

No entanto, é importante destacar que o cadastramento dos novos servidores FTP públicos é realizado manualmente pelo administrador Archie [BOW 94]. Esta é uma tarefa difícil, visto as freqüentes alterações de topologia da Internet. Neste sentido, a atualização mensal dos índices e a não existência de um processo automático de mapeamento dos servidores FTP, podem levar algumas vezes a consultas de dados desatualizados.

Por outro lado, a constante atualização de índices seria muito custosa. O servidor Archie existente na Universidade de McGill, por exemplo, é responsável por metade do tráfego de informações entre Toronto e o resto da Internet quando ocorre uma atualização [EMT 91]. Além disso, é exigida uma grande capacidade de processamento para a regeração dos índices de pesquisa.

Uma consulta ao banco de dados Archie pode ser realizada usando um variado número de métodos de acesso e canais. Entre estes incluem-se o *telnet*, E-Mail, WAIS, sistemas WWW e através do protocolo de sistemas de arquivos Prospero [EMT 92]. Todos estes métodos são interfaces para clientes Archie.

O usuário realiza uma consulta ao sistema por intermédio de um cliente Archie, fornecendo-lhe um padrão de busca, representado por uma cadeia de caracteres, a qual é comparada com cada uma das entradas do índice anteriormente referido. Sempre que houver o casamento entre o padrão de busca com alguma das entradas deste índice, o sistema responde ao usuário, informando-lhe o nome do arquivo ou diretório que satisfez a consulta, o nome do servidor FTP público que o contém, bem como a localização (*path*) dessas informações dentro do mesmo.

É relevante observar que para haver o casamento, o padrão de busca e a entrada no índice de dados não precisam ser necessariamente idênticas [FER 94]. Neste particular, o sistema é bastante flexível, oferecendo aos usuários variadas alternativas de operação. Dentre estas, por exemplo, pode-se citar o uso de expressões regulares na especificação do padrão de pesquisa. Neste caso, se usássemos, por exemplo, o padrão "aaa*", seriam respostas válidas para a nossa consulta, entradas da base de dados que contivessem a cadeia de caracteres "aaa", sucedida de quaisquer caracteres.

2.3.2 WAIS

O WAIS, cuja sigla significa *Wide Area Information Server*, é o resultado do esforço combinado de três companhias que compartilham um interesse mútuo em tornar a informação acessível para o público, particularmente de uma forma descomplicada: Apple Computers, Dow Jones e Thinking Machines Corporation [WYA 95]. O desenvolvimento do sistema iniciou em outubro de 1989, com a primeira versão para a Internet surgindo somente em abril de 1991.

A arquitetura do sistema WAIS é constituída por quatro componentes principais [KAH 91]:

- Cliente;
- Servidor;
- Base de Dados;
- Protocolo.

Implementado sobre o modelo cliente/servidor, o WAIS cliente permite que o usuário pesquise em um banco de dados WAIS servidor. O programa cliente é responsável pela interação com o usuário, procurando, em cooperação com os servidores, localizar e recuperar as informações por eles desejadas [FER 94]. De fato, todo o trabalho de pesquisa dentro do sistema WAIS á realizado pelo processo servidor, no qual reside índices para bancos de dados dos mais diversos assuntos e formatos de armazenamento.

Tais índices são muito efetivos quando utilizados sobre arquivos de textos ASCII. No entanto, o uso do WAIS para indexar executáveis e outros arquivos existentes na rede Internet não é muito efetivo [HAR 93]. Os índices tendem a se tornar pobres para o trabalho de localização da informação, além de volumosos. Cabe salientar que os servidores mantêm índices totais, contendo palavras-chave (entradas) de cada palavra que aparece nos documentos textuais (até palavras comuns, como "*the*") [SCH 92].

Existem vários servidores WAIS distribuídos pela rede Internet, cada um destes com seu próprio índice de pesquisa. Esta característica gera um processo de pesquisa escalonável, onde o usuário, primeiramente, seleciona em quais servidores pesquisar e, posteriormente, realiza a pesquisa sobre os índices destes servidores. Não é possível realizar diretamente uma busca sobre um índice "global".

O processo cliente funciona, tão somente, como um intermediário entre o processo servidor e o usuário do WAIS, fornecendo a ele uma interface adequada a suas necessidades de pesquisa. Uma consulta é realizada em linguagem natural, utilizando operadores booleanos. A característica da interface natural permite um acesso facilitado ao conjunto de documentos disponíveis. Os documentos são recuperados pela sua semelhança com o texto da consulta, ou seja, pelo número de palavras coincidentes. A resposta mostra os documentos classificados em ordem decrescente de acertos. Através dos títulos dos documentos o usuário pode avaliar a qualidade da seleção e indicar quais documentos que realmente satisfazem sua consulta.

A próxima etapa da consulta é conhecida como *Relevance Feedback*. Nesta etapa, os documentos selecionados servem de padrão de busca para a próxima pesquisa, refinando, assim, a busca de informações. Dow Jones verificou, em seu banco de dados DowQuest2, que este mecanismo de refinamento é muito eficiente na busca de informações em grandes bancos de dados [KAH 91]. É importante lembrar que a decisão de ressubmeter a consulta é do usuário, não estando prevista busca incremental ou reformulação automática da consulta para permitir uma busca incremental.

Toda a estrutura cliente/servidor que forma o sistema WAIS como um todo é baseada em um protocolo padrão: o Z39.50-1988. No entanto, não foi adotada

integralmente sua especificação, pois este protocolo foi desenvolvido fundamentalmente para serviços de pesquisa e recuperação de dados bibliográficos, o que, indubitavelmente, não satisfazia as necessidades de caráter mais genéricas inerentes ao sistema WAIS. Sendo assim, adotou-se um subconjunto do protocolo Z39.50 original, agregando-lhe algumas características que vêm ao encontro das necessidades bem próprias do sistema WAIS [FUL 94], permitindo a pesquisa e recuperação de informações genéricas e não particularizadas.

Com o intuito de tornar possível a construção de servidores que operam no modo *stateless*, ambas as funções, consulta e recuperação, são implementadas usando-se apenas o serviço de pesquisa proposto no Z39.50-1988. Sendo assim, os servidores não precisam manter os resultados de uma consulta, para que seja possível realizar a posterior recuperação dos documentos por parte dos processos clientes [FER 94].

O uso de um protocolo único permite o acesso a máquinas heterogêneas (FIGURA 2.1). Dada a característica *stateless* deste protocolo, a cada consulta corresponde uma resposta e cada consulta é uma unidade independente.

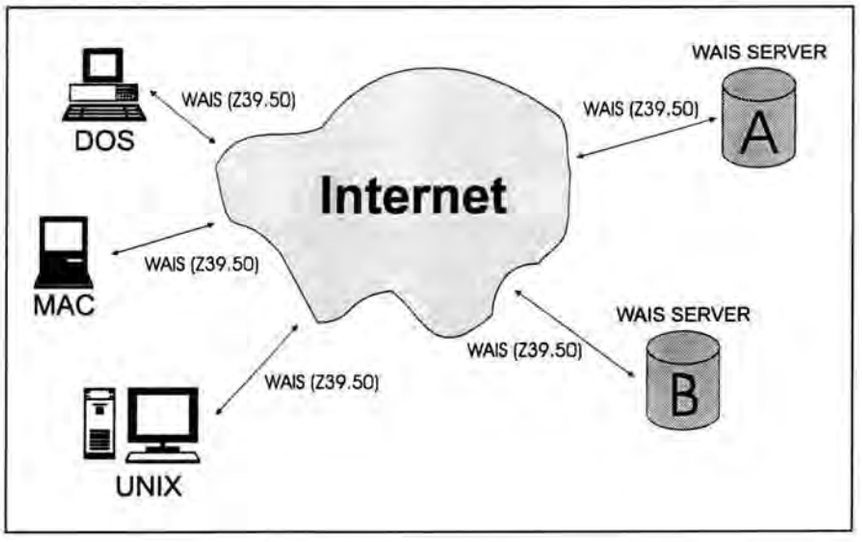

FIGURA 2.1 - *Wide Area Information Server*

2.3.3 Essence

O grande volume e diversidade dos dados, juntamente com a necessidade de índices de pesquisa compactos, sugere o uso de diferentes meios de indexação de dados conforme o arquivo original.

O Essence é uma ferramenta de descoberta de dados sobre os arquivos contidos nos servidores da rede Internet, tendo sido desenvolvido na Universidade do Colorado, nos Estados Unidos da América. Este sistema de descoberta explora a semântica dos arquivos a fim de gerar um índice de palavras-chave para a pesquisa, podendo indexar

tanto arquivos textuais, como binários. Pela exploração semântica, o Essence utiliza para geração dos índices somente as palavras que sumarizam um arquivo, resultando em um índice compacto, mas que representa o conteúdo original.

Indexação semântica envolve a análise da estrutura dos diferentes tipos de informação e formatos de armazenamento, dependendo do tipo do arquivo original [HAR 95]. Por exemplo, as páginas do manual do UNIX são separadas dentro de uma estrutura de seções, a partir das quais é possível extrair informações sobre o nome de um programa ou arquivo, e informações do autor. Pela geração de informações de diferentes tipos de arquivos de diferentes maneiras, a indexação semântica pode gerar palavras-chave representativas do arquivo original sem incluir cada palavra a partir deste. Além disso, poupando espaço e evitando a inclusão de palavras que prejudiquem a qualidade de um índice.

O Essence determina o tipo do arquivo pela exploração das convenções de nomes, como, por exemplo, a extensão ".TXT", e localizando dados de identificação ou estruturas comuns nestes arquivos, como o UNIX "*magic number*". Uma vez determinado o tipo de arquivo, o Essence executa o sumarizador apropriado para extrair as palavras-chave do arquivo. Entre outros tipos, Essence reconhece estruturas de arquivos *nested*, como são os arquivos compactados. Neste caso, o sistema recursivamente extrai os arquivos contidos dentro do arquivo original e indexa-os.

Como pode-se observar, a indexação semântica realizada pelo Essence envolve três estágios [HAR 93], como é apresentado na FIGURA 2.2 e descrito a seguir.

2.3.3.1 Classificação

O sistema determina os tipos de arquivos a partir dos quais serão gerados índices, usando uma combinação de exploração de convenções de nomes de arquivos e heurísticas de identificação de dados e estruturas comuns entre os arquivos.

Observando cada simples convenção de nomes de arquivos, pode-se determinar os tipos de arquivos com um alto grau de certeza. A mais básica convenção de nomes de arquivos está nas extensões. Por exemplo, arquivos com a extensão ".c" são tipicamente arquivos de código fonte em C; nomes de arquivos com a extensão ".ps" são tipicamente arquivos em *PostScript*; e arquivos com a extensão ".txt" são tipicamente arquivos texto em ASCII.

No Essence, as convenções de nomes de arquivos são representas por expressões regulares, como, por exemplo, *.ps ou [Mm]akefile.*, representando os tipos de arquivo *PostScript* e *Makefile*, respectivamente. Expressar os nomes de arquivos como expressões regulares permite a cada sítio de informação integrar sua semântica local dentro do Essence.

2.3.3.2 Sumarização

Os sumarizadores do Essence são programas UNIX escritos e integrados dentro do sistema. Esta característica provê um potente paradigma para exploração semântica.

Cada sumarizador é associado com um tipo específico de arquivo, entendendo o formato deste, bem como sabendo extrair as informações relevantes a partir do mesmo.

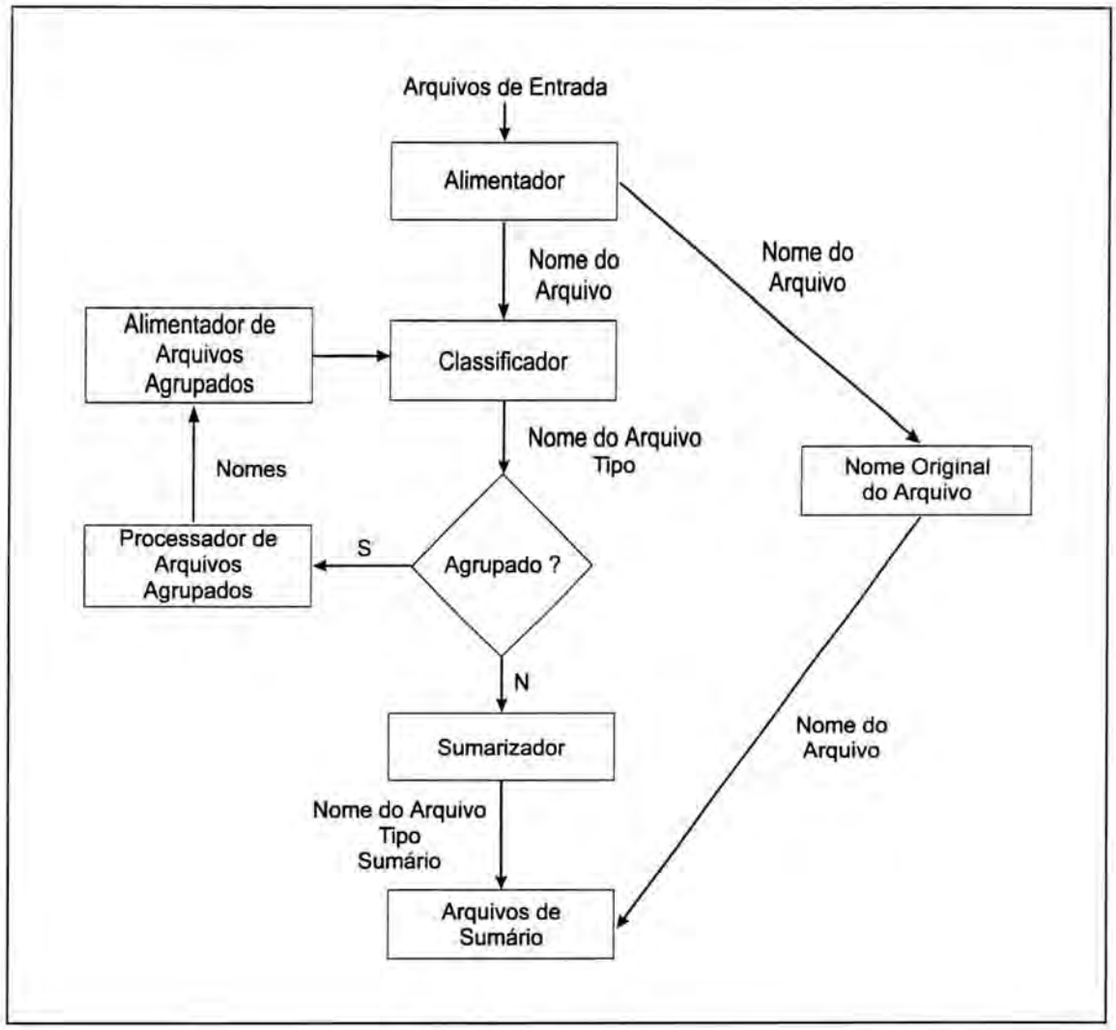

FIGURA 2.2 - Essence

Por exemplo, o sumarizador para uma página do manual UNIX baseado em *Troff* entende a sintaxe Troff e as convenções usadas para descrever programas UNIX. Ele utiliza este conhecimento para extrair a informação sumarizada, como o título do programa, programas e arquivos relacionados, o autor do programa e uma breve descrição deste programa.

No entanto, diversos tipos de arquivo não prestam-se facilmente à interpretação automática. Por exemplo, arquivos ASCII simples, tipicamente contento textos não estruturados, são dificilmente explorados de forma efetiva.

O Essence oferece sumarizadores para diversos tipos de arquivo, tais como: arquivos binários, arquivos de código fonte em C, arquivos de *headers* em C, *mails*, Dvi, *News*, arquivos de código objeto, arquivos texto, Perl e *PostScript*. Cada sumarizador extrai o conteúdo dos arquivos de uma forma diferente, como, por exemplo, os sumarizadores apresentados na TABELA 2.1.

TABELA 2.1 - Sumarizadores

Tipo do Arquivo	**Descrição**	**Sumarizador**
Archive	Arquivos de biblioteca	Extrai tabela de símbolos
Binário	Executável binário	Extrai frases significativas e páginas de manual
C	Código fonte em C	Extrai nomes de *procedures*, arquivos *include* e comentários
C *Header*	Arquivos *header* em C	Extrai nomes de procedimentos, arquivos *include* e comentários
Comando	*Scripts* em Shell no Unix	Extrai comentários
Diretório	Diretórios	Extrai lista de diretórios, informações de *Copyright* e arquivos *README*
Dvi	*Device Indep. Tex output*	Converte para texto ASCII
Mail	Correio eletrônico	Extrai campos do cabeçalho
Makefile	*Makefiles* do UNIX	Extrai comentários
ManPage	Páginas de manual UNIX	Extrai nome do autor, título, etc., baseado nas macros "-man"
News	Artigos USENET *news*	Extrai campos do cabeçalho
Objeto	Arquivo em formato objeto	Extrai tabela de símbolos
Patch	Arquivos de lista de diferença	Extrai nome do arquivo
Perl	*Script* Perl	Extrai nomes de procedimentos e comentários

PostScript	Imagens *PostScript*	Converte para texto ASCII
Tipo do Arquivo	**Descrição**	**Sumarizador**
README	Informações gerais	Extrai arquivo inteiro
SCCS	Arquivo de controle SCCS	Extrai sumário fornecido pelo SCCS
Tex	Documentos fonte TeX	Converte para ASCII
Texto	Texto ASCII não estruturado	Extrai as 100 primeiras linhas
Troff	Documento fonte Troff	Extrai nome do autor, título, etc., baseado nas macros "-man", "-me", "-ms", ou então extrai cabeçalhos de cada seção e sentenças de tópicos

Na realidade pode-se dividir os sumarizadores em cinco grupos de semelhança:

- **Texto**

Arquivos deste tipo são difíceis de serem sumariados, pois este são estruturados. O Essence assume que a informação de maior qualidade e importância, na maioria dos textos não estruturados, está próxima ao início do arquivo. Neste sentido, o *Sumarizador de Arquivos Texto* extrai palavras-chave a partir das cem primeiras linhas do arquivo.

- **Diretório**

Para executar o sumário de um diretório, o Essence obtém a lista dos arquivos contidos no diretório e algumas informações adicionais, tais como informações de *copyright* de cada arquivo (autoria, títulos, etc.). Também é realizada a sumarização de arquivos *README* encontrados nos diretórios, pois estes contêm informações importantes sobre o conteúdo do diretório como um todo.

- **Código Fonte e Código Objeto**

Estes tipos de arquivo são altamente estruturados e oferecem facilidades para exploração de informações semânticas. O *Sumarizador C* extrai nomes de procedimentos, *headers* e comentários a partir do código fonte em C. Similarmente, o *Sumarizador de Código Objeto* extrai a tabela de símbolos existente no arquivo objeto.

- **Binário**

O sumarizador de arquivos binários obtém normalmente as seqüências de caracteres ASCII contidas nestes arquivos, utilizando para isto o comando *strings* do UNIX. No entanto, para conseguir uma maior relação entre as seqüências de caracteres extraídas e o arquivo em questão, o sistema Essence utiliza uma série de heurísticas que mantém somente informações relacionadas com o arquivo, tais como versão, *copyright*, etc. Além disso, o sistema procura informações associadas dentro do diretório, tais como páginas de manual e outras referências.

- **Texto Formatado**

Existem certos tipos de arquivo que são estruturados de alguma forma, tais como arquivos de *Mail*, *News* e Troff. Neste caso, o Essence explora a semântica dos arquivos através da sintaxe dos campos contidos nos arquivos, como, por exemplo, o campo *from* em um *mail*.

- **Tipo desconhecido**

No caso dos tipos de arquivo não tratados são extraídos somente o nome do mesmo e seu proprietário.

2.3.3.3 Estrutura de Arquivos Agrupados

Arquivos agrupados são arquivos que contêm outros arquivos inseridos dentro de sua estrutura. Exemplos deste tipo de estrutura compreendem arquivos comprimidos, formato *tar*, *uuencode*, ZIP e formato Shell. Além disso, pode haver uma recursividade de arquivos agrupados, como, por exemplo, um arquivo no formato *tar* e comprimido, ou no formato *uuencode* e comprimido.

Quando o sistema Essence determina que um arquivo é na verdade uma estrutura de arquivos agrupados, ele extrai os arquivos "escondidos", determina o tipo dos arquivos resultantes e gera o sumário para eles. Este processo continua recursivamente até não serem encontrados mais arquivos agrupados.

A extração de arquivos agrupados é feita utilizando o programa correspondente ao tipo de estrutura existente. Por exemplo, utiliza-se o comando *uncompress* do UNIX para arquivos comprimidos com extensão ".Z", o comando *tar* com o flag "x" para arquivos no formato *tar*, e o comando *uuencode* para arquivos no formato *uuencode*.

2.3.3.4 Exportação de Índices

O Essence exporta os índices gerados através de pesquisas WAIS e sua interface de apresentação dos resultados de pesquisa, permitindo ao usuário utilizar interfaces gráficas como X Windows. Para a geração de índices compatíveis com os do WAIS, o Essence usa o programa de indexação do WAIS para indexar seus arquivos de sumário.

2.3.4 *Recall* e *Precision*

Um sistema de extração de informações pode ser avaliado por sua eficiência em filtrar e selecionar as informações desejadas pelo usuário, ou seja, em outras palavras, a validade dos dados recuperados segundo o usuário. Uma recuperação eficiente depende de dois fatores principais, conforme [SAL 87]:

- os itens relevantes para as necessidades do usuário devem ser recuperados;
- os itens estranhos, ou não relevantes, devem ser rejeitados.

Duas medidas são normalmente utilizadas para descrever a habilidade de um sistema na recuperação dos itens relevantes e na rejeição dos itens não relevantes de uma base de dados: *recall* e *precision* (FIGURA 2.3). *Recall* é a proporção de itens relevantes recuperados, sendo calculado pela divisão do número de itens relevantes recuperados pelo número total de itens relevantes na base de dados. *Precision* é a proporção de itens recuperados que são relevantes, sendo calculado pela divisão do número de itens relevantes recuperados pelo número total de itens recuperados.

Em princípio é preferível que um sistema produza alto índice de *recall*, pela recuperação de tudo que é relevante, e também alto índice de *precision* pela rejeição de todos os itens que são estranhos às necessidades do usuário. No entanto, tais condições de *recall* e *precision* são conflitantes, pois quando tem-se a recuperação de muitos itens relevantes (*recall*), ocorre também a recuperação de dados não relacionados à pesquisa realizada. Por outro lado, quando a maioria dos itens recuperados são relevantes (*precision*), muitas outras informações relevantes acabam não sendo recuperadas. Na prática, sabe-se que *recall* e *precision* tendem a variar inversamente, pois é muito difícil recuperar tudo que é relevante, desprezando tudo o que não é relevante. A FIGURA 2.4 mostra a relação entre recuperação e precisão.

A função de *recall* na recuperação de dados é normalmente melhor satisfeita pelo uso de termos amplos, ou de alta freqüência, que ocorrem em vários documentos da base de dados. Termos com estas características podem ser usados para colocar de fora da seleção várias informações de um arquivo de entrada (ou todo o arquivo). Contudo, nesta exclusão, podem estar incluídos vários dados relevantes que deveriam ser recuperados.

Por outro lado, o fator *precision* é normalmente melhor satisfeito pelo uso de termos restritos, ou altamente específicos, que são capazes de isolar poucos itens relevantes a partir da massa dos não relevantes.

Como exemplo de medida destes indicadores, podemos citar o estudo realizado com o sistema STAIRS (*Storage and Information Retrieval System*) da IBM. Em um teste com este sistema, conduzido por Blair e Maron [SAL 86], um valor médio de *precision* encontrado foi de 75%, e um valor médio de *recall* foi de 20%. Isto significa que a cada quatro documentos encontrados, três tinham ligação com a consulta do usuário, e aproximadamente um quinto do total de itens relacionados com a consulta eram encontrados.

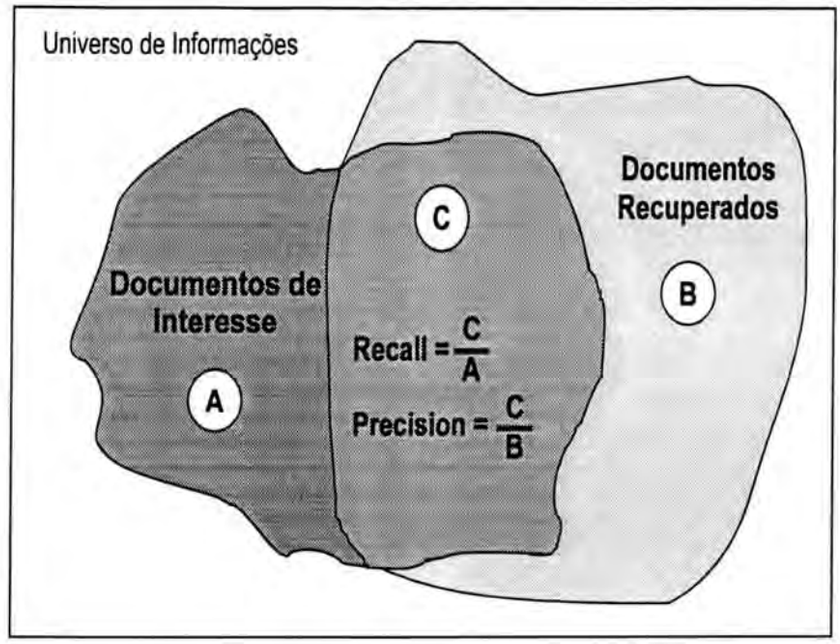

FIGURA 2.3 - *Recall e Precision*

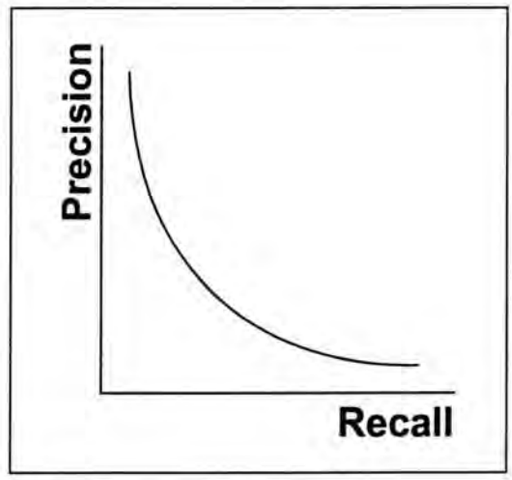

FIGURA 2.4 - Relação entre *Recall* e *Precision*

2.3.5 Extração do Conteúdo Semântico

Como pode ser observado, segundo o uso de técnicas de extração de informações, os três sistemas de recuperação apresentados utilizam-se de métodos diferentes para a extração do conteúdo semântico dos arquivos disponíveis na rede Internet. Estes diferentes processos de extração, visando posterior recuperação de arquivos de dados, são importantes no desenvolvimento deste trabalho, pois algumas dessas técnicas foram utilizadas no protótipo implementado.

O WAIS opera sobre informações genérica (documentos textuais sobre qualquer assunto), explorando a redundância característica da linguagem humana em textos [SCH 92]. Neste sentido, o usuário informa a(s) palavra(s) para pesquisa, e o sistema retorna a

este os arquivos com maior freqüência destas palavras. Assim sendo, um conjunto de palavras que fazem parte de um arquivo, aparecendo com uma determinada freqüência em relação a outros arquivos onde também existam tais palavras, define o conteúdo semântico do arquivo em relação à pesquisa realizada.

No entanto, esta característica de trabalhar sobre informações genéricas exige a indexação de todas as palavras de um texto, resultando em índices volumosos, e respostas de pesquisas muito amplas. Ou seja, o conjunto das respostas a uma pesquisa é grande, podendo muitas vezes conter informações não interessantes ao usuário. Assim, tem-se alto *recall* e médio para baixo *precision*.

Por exemplo, pode-se realizar uma pesquisa com as palavras "*Internet Implementation*", com o objetivo de estudar suas características técnicas. Contudo, o resultado da consulta contará com centenas de arquivos, alguns deles falando, por exemplo, em publicidade na Internet. Neste caso, dentro do conjunto de respostas (arquivos com a palavra Internet), muitos arquivos não serão interessantes ao pesquisador.

Este alto *recall* implica em uma maior carga sobre o usuário do WAIS, pois este deve selecionar bem as palavras a serem utilizadas em uma pesquisa, a fim de obter resultados de maior qualidade. No entanto, o *Relevant Feedback* auxilia muito na tarefa de refinar os resultados de uma consulta, a fim de atingir uma melhor precisão e conseqüente melhora de qualidade nos resultados.

O Archie, por outro lado, utiliza um volume muito menor de informações para avaliar o conteúdo semântico de um arquivo: somente o nome deste ou do diretório onde este esta localizado. Isto permite a geração de índices de pesquisa menores; contudo, o resultado de uma pesquisa apresenta, normalmente, uma grande quantidade de respostas (*recall*) de baixa qualidade (*precision*).

O Essence, por outro lado, junta características dos dois sistemas acima. O uso do nome do arquivos é uma das características levadas em conta no processo de extração dos dados que irão fazer parte dos índices de pesquisa. Mais especificamente, o nome do arquivo é muito importante na fase de classificação deste.

No entanto, deve-se observar que o nome do arquivo passa a ter conteúdo semântico dentro do processamento deste mesmo arquivo, e não somente quando relacionado a uma consulta.

No Essence, como no WAIS, o conteúdo interno dos arquivos também é levado em conta no processo de extração de informação, no entanto seus índices de pesquisa são menores, além de serem formados por palavras-chave de melhor qualidade. Esta característica deve-se ao processo de extração semântica utilizado, ou seja, um processo adaptado ao tipo de cada arquivo que irá compor o índice de pesquisa.

Utilizando esta técnica, o sistema Essence obtém resultados de *recall* e *precision* superiores aos sistemas anteriores, como é apresentado em [HAR 93].

2.4 Tratamento da Informação Temporal

O sistema proposto para manipulação de informações locais utiliza-se de dados temporais para a realização de tal manipulação. Contudo, para o tratamento da informação temporal é necessário buscar esta informação de diferentes fontes ou serviços disponíveis. Como descrito anteriormente, uma vez localizada a informação relevante na rede e criada a entrada correspondente no sistema de armazenamento local, surge o problema de determinação da validade temporal desta informação. Um exemplo simples é a obtenção do endereço físico de correio eletrônico válido de um usuário, como alguns exemplos a seguir irão apresentar.

2.4.1 Informação temporal explícita

O visualizador NetScape permite definir explicitamente o tempo de validade dos *links* percorridos. Desta forma, o usuário pode declarar por quanto tempo a informação de acesso é mantida válida. Esta definição atua a partir da descrição dos dados de data da criação e do último acesso ao *link*, mantida na descrição dos arquivos *http* acessados. O tempo é mantido como uma cadeia de caracteres no formato de data interno e característico do sistema empregado (exemplo 1). Entretanto a definição de prazo de validade é global, não sendo permitida a especificação de validade para um *link* em particular.

- **Exemplo 1: HTTP**

> Linha de um arquivo html empregado pelo NetScape para referenciar https remotos:
>
> <DT> <A HREF="http://opera.cit.gu.edu.au:80/dstc/" **ADD_DATE= "786168648" LAST_VISIT="786168547"**>Distributed Databases - Australia </A>

No caso de artigos de *news groups* ou mensagens de correio eletrônico há um atributo com a informação explícita do instante de geração da mensagem. Este atributo contém a data e a hora de criação da mensagem bem como informação sobre o fuso horário (exemplos 2 e 3).

- **Exemplo 2: News**

> Path: caracol-gw.inf.ufrgs.br!lakesis.fapesp.br! ...
> From: pop3@interramp.com (pp000775@interramp.com)
> Newsgroups: psi.interramp,comp.infosystems.www
> Subject: Re: Netscape V1.0
> **Date: 23 Dec. 1994 08:42:29 -0500**
> Organization: PSI Public Usenet Link

Neste caso, a informação temporal explícita consiste no registro do tempo de transação que, neste caso, corresponde à data e hora em que a mensagem foi enviada e está contida no atributo **Date: 23 Dec. 1994 08:42:29**. O último componente **-0500** registra a diferença entre a hora local e o tempo GMT.

- **Exemplo 3: Correio eletrônico**

> Return-Path: <palazzo@inf.ufrgs.br>
> **Date: Fri, 16 Dec 94 15:42:04 EDT**
> X-Sender: palazzo@castor.inf.ufrgs.br
> To: cpgcc@inf.ufrgs.br
> From: palazzo@inf.ufrgs.br (Jose Palazzo M. de Oliveira)
> Subject: Selecao

Neste exemplo, a informação temporal é idêntica ao do exemplo 2, mas a informação sobre a diferença de fuso em relação ao fuso horário GMT é codificada na cadeia de caracteres **EDT**.

2.4.2 Informação temporal implícita

Em alguns casos a informação temporal pode ser determinada a partir de atributos temporais de outros objetos do repositório local (exemplo 4), ou, ainda, a partir de regras heurísticas armazenadas no sistema (exemplo 5).

- **Exemplo 4: E-mail X-400**

> Return-Path: < Yyyyy @rz.informatik.uni-hamburg.d400.de>
> X400-Received: by mta d400relay in /PRMD=dfnrelay/ADMD=d400/C=de/; Relayed; Thu, 6 Jan 1994 19:33:37 +0100
> Date: Thu, 6 Jan 1994 17:09:12 +0100
> From: Xxxxx Yyyyy < Yyyyy @rz.informatik.uni-hamburg.d400.de>
> To: palazzo <palazzo@inf.ufrgs.br>
> **Subject: Re:IFIP '94 question**

Neste exemplo, há uma informação temporal implícita. O Congresso IFIP '94 foi realizado entre 28 de agosto e 2 de setembro de 1994, provavelmente qualquer resposta a uma consulta posterior à data final do Congresso não seria mais útil e isto deveria ser sinalizado para o usuário. Para que possa ser definida esta restrição, é necessário (**i**) identificar a cadeia de caracteres **IFIP '94** como se referindo a um congresso e (**ii**) determinar o período de realização a partir da análise do conteúdo de *outras* mensagens, provavelmente um artigo no grupo *news_anounce_conferences*, e armazenar esta informação em um banco de dados temporal de referência.

Uma regra heurística para determinar informação temporal implícita pode ser a seguinte:

- **Exemplo 5: Regra heurística**

> Uma propriedade importante para a determinação da validade de uma informação é o tempo de validade, mas muitas vezes o mesmo não está disponível. Pode ser definida a regra empírica que no instante em que uma informação é obtida na rede a mesma é valida. Uma referência a um artigo disponível para FTP pode ser considerada válida no instante de sua obtenção, imagina-se que ao remover o arquivo o responsável removerá, também, as

referências incluídas em outros objetos acessíveis remotamente. Esta heurística não pode produzir resultados irreparáveis pois no pior caso será realizada uma tentativa de acesso a um arquivo não existente.

2.4.3 Informação temporal deduzida

A informação temporal explícita ou implícita, para ser útil, deve ser mantida em uma estrutura de dados adequada. Além disto, a manipulação de arquivos e registros pelos usuários permite a descoberta de conhecimento sobre a validade temporal dos dados armazenados. Os usuários, ao manipularem seus bancos de dados pessoais, apresentam um comportamento que é derivado, parcialmente, das características de validade temporal dos dados. Assim, *call-for-papers* com data passada são removidos, *home pages* acessadas há algum tempo são removidas ou recarregadas.

Este comportamento varia de usuário para usuário e permite a descoberta de características de validade temporal pela análise de uma série de ações. É suposto que o processo de aprendizado, isto é, a identificação de uma classe de procedimentos é determinada unicamente pelas variáveis de predição. Muitas vezes estas variáveis não são conhecidas, pois podem existir variáveis relevantes para a classificação mas não registradas no banco de dados. Há três alternativas para tratar as variáveis de decisão:

- regras determinísticas, construindo regras apenas quando todas as variáveis relevantes à classificação são conhecidas;
- tentativa de descobrir regras probabilísticas que indicam a possibilidade de um objeto, coberto pela descrição, pertencer à classe;
- um processo híbrido, combinando regras determinísticas com probabilísticas, quando necessário. Esta última alternativa foi a aplicada neste trabalho.

2.5 Manutenção da Informação Temporal

Para a manutenção das informações temporais pode ser utilizado o modelo de dados temporal orientado para objetos TF-ORM [EDE 93]. Para a manutenção de regras de derivação pode-se usar a lógica temporal apresentada em [EDE 94a].

Bancos de dados temporais apresentam atributos temporais associados aos dados. Na classificação proposta em [SNO 85] e refinada em [TAN 93] são identificadas as seguintes quatro categorias diferentes de bancos de dados:

- bancos de dados instantâneos: bancos de dados comuns, não temporais, que não apresentam informações temporais associadas aos dados;
- bancos de dados de tempo de transação: somente o tempo de transação é associado aos dados, correspondendo este, portanto, à validade dos dados;

- bancos de dados de tempo de validade: utilizam o tempo de validade como atributo temporal e não o tempo em que a informação é introduzida no banco de dados;

- bancos de dados bitemporais: apresentam os dois tempos como atributos - o tempo de transação e o tempo de validade.

O modelo de dados é temporal somente quando é definido um banco de dados bitemporal. Neste sentido, a cada informação existente em um banco de dados temporal é associado tanto o tempo de transação como o tempo de validade. O elemento temporal primitivo utilizado é o ponto no tempo, sendo o tempo considerado como linearmente ordenado, variando de forma discreta. Os objetos apresentam métodos, são instâncias de classes e evoluem com o passar do tempo. Uma classe é definida através de um nome, um conjunto de propriedades que o objeto apresenta neste papel, um conjunto de estados abstratos que o objeto pode assumir neste papel, um conjunto de mensagens que podem ser enviadas e recebidas por este papel e um conjunto de regras (regras de transição de estados e regras de integridade). Fórmulas lógicas podem ser associadas às regras de transição, atuando como restrições às possíveis transições de estados. A evolução dinâmica de um objeto é representada através destas regras. As propriedades podem ser estáticas, quando apresentam o mesmo valor durante toda a existência da instância, ou dinâmicas, quando estas propriedades podem assumir diferentes valores com a passagem do tempo. Às propriedades dinâmicas são associados os atributos temporais. Um domínio de valores é associado a cada propriedade definida.

No entanto, muitas das informações recuperadas pela Internet não estão armazenadas em bancos de dados temporais, o que permitiria sua fácil manipulação. É neste momento que observa-se a necessidade de uma solução tecnológica, ou seja, uma solução que, de forma automatizada, utilize os mais atuais conhecimentos, a fim de auxiliar na manipulação de dados pessoais.

2.6 *Knowledge Discovery*

Knowledge Discovery, ou, em português, Descoberta de Conhecimento, é uma área da Ciência da Computação que nasceu da necessidade de gerenciar uma maçissa quantidade de dados produzida diariamente [FRA 92]. Neste sentido, dentro deste grande volume de informações coletadas é necessário encontrar o conhecimento escondido. Contudo, analisar estes dados manualmente não é somente custoso, mas também impossível visto que o processo de coleta é muito mais rápido do que o processo manual de descoberta e extração do conhecimento. Atualmente, tem-se empregado ferramentas de descoberta de conhecimento para encontrar modelos de dados interessantes a partir de bases de dados coletadas, e usar estes modelos para a extração do conhecimento armazenado.

A noção de *Knowledge Discovery in Databases* (KDD) tem sido denominada *Data Mining*. Em sua essência, consiste em extração (não trivial) de informação implícita, previamente desconhecida e potencialmente útil [SHA94]. Encontrando um modelo que considerado interessante, a saída do processo pode ser considerada como conhecimento.

O grande desafio da área de KDD é automatizar o processamento de grandes quantidades de dados brutos, identificando os mais significativos e as configurações relevantes e apresentando estes como o conhecimento apropriado para a realização das conclusões do usuário [SHA 94].

KDD engloba uma série de técnicas como determinação de grupos, sumarização de dados, aprendizagem de regras de classificação, descoberta de redes de dependências, análise de mudanças e detecção de anomalias [MAT 93]. Neste sentido, a Descoberta de Conhecimento é a combinação de áreas de Bancos de Dados e Inteligência Artificial, ou mais especificamente, KDD é a aplicação de técnicas de inteligência artificial sobre um banco de dados.

2.6.1 Descoberta de Conhecimento em Bases de Dados Temporais

Bases de dados clássicas modelam aspectos estáticos dos dados. Assim, a informação nas bases de dados consiste em dados instantaneamente verdadeiros no mundo real. No entanto, em muitas aplicações dinâmicas, como são aplicações para tratamento de informações financeiras, a história dos dados é tão importante quanto a informação estática.

A descoberta de conhecimento a partir de bases de dados clássicas tem valor limitado a partir do ponto em que a natureza temporal dos dados não é observada dentro de um valor, mas somente entre o valor corrente e o último *snapshot*. Segundo [SAR 95], a nova geração de sistemas de bancos de dados e em particular os que acomodam características temporais são as mais apropriadas plataformas para descoberta de conhecimento. Quando a história é tomada em valores, as técnicas de descoberta de conhecimento também consideram a evolução dos dados através do tempo e seu período de validade. Sistema de bases de dados temporais tem tornado-se crescentemente populares e influenciam no desenvolvimento de novas gerações de sistemas de bases de dados. Estas características têm motivado o estudo e a pesquisa de mecanismos de descoberta de conhecimento em bases de dados temporais.

No caso de KDD aplicado a modelos temporais, a bibliografia existente é em geral muito restrita. Entre os trabalhos relatados está o de Theodoulidis e Saraee [SAR 95], em particular o artigo Knowledge Discovery in Temporal Databases que demonstra a importância de bancos de dados temporais para a descoberta de conhecimentos. Entretanto, existe bibliografia específica sobre a descoberta de conhecimento em séries históricas ligadas às ciências naturais. Neste último caso, a regularidade das séries facilita bastante a tarefa de busca. Muito trabalho tem sido desenvolvido pelo UCLA Data Mining Laboratory [MES 96].

2.7 Objetivos e Solução

Conforme destacamos anteriormente, a grande disponibilidade de informações hoje acessível aos usuários de computadores tem trazido claros benefícios à humanidade Contudo, ainda existem problemas de manipulação deste volume de informação. A

maior dificuldade está exatamente na não estruturação da grande maioria dos dados trazidos pela Internet, por exemplo.

Por outro lado, há muito tempo a importância do texto tem sido reconhecida por pesquisadores na área de recuperação de conhecimento como uma base de dados não estruturada, como no caso de textos livres. No mesmo sentido, ainda existem os textos semi-estruturados, quando estes apresentam algum padrão interno de estruturação previamente definido, como, por exemplo, cabeçalhos de arquivos de correspondência eletrônica. Desta forma, tem-se buscado novas técnicas de descoberta do conhecimento a partir de bases de dados com estas características de não estruturação formal.

É com o propósito de auxiliar na manipulação de bases de dados não estruturadas e semi-estruturadas, que apresenta-se o estudo e o protótipo descritos nesta dissertação. O SES (Sistema de Extração Semântica) utiliza um conjunto de técnicas de programação na busca desta solução.

O objetivo é analisar um grande conjunto de bases de dados não estruturadas, extraindo destas o núcleo do conhecimento, ou seja, o assunto principal, conforme as necessidades do usuário, e gerar uma base de dados de saída estruturada e que possa ser rápida e facilmente analisada e manipulada pelo próprio usuário ou outro sistema, neste caso uma base de dados temporal (FIGURA 2.5).

No entanto, é importante salientar que diferentemente de outros sistemas já existentes na área de Extração de Informações, os quais centralizam-se na extração de informações de documentos com temas e línguas pré-determinadas, o protótipo proposto neste trabalho visa atender necessidades de extração com as mais diversas características. Desta forma, o sistema irá adaptar-se às necessidades e anseios do usuário.

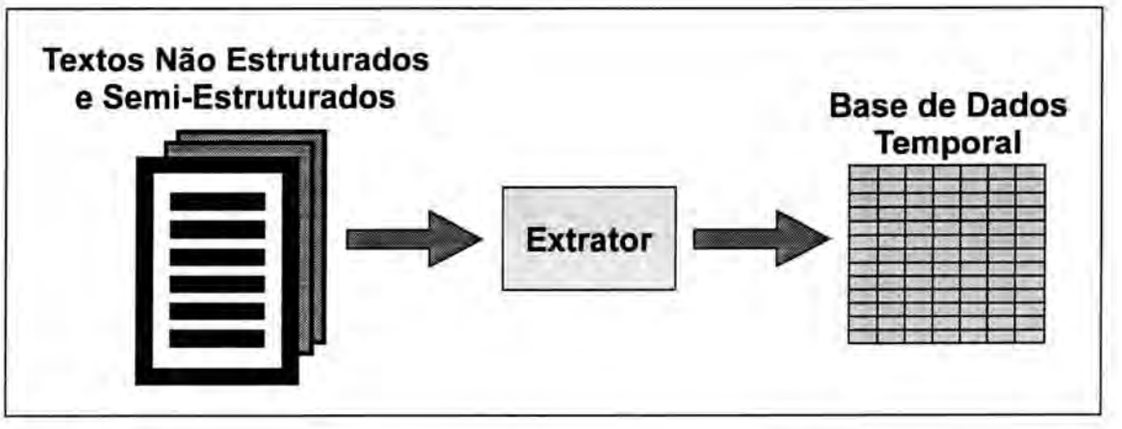

FIGURA 2.5 - Objetivo de Extração

Na prática, o protótipo desenvolvido, buscará gerar como resultado do processo de extração de informações uma base de dados temporal. Desta forma, será possível verificar a validade temporal das informações não estruturadas originais, estruturando-as e facilitando sua manipulação por um banco de dados temporal.

A manipulação das informações de forma estruturada, facilitará a descoberta de novos conhecimentos a partir dos originalmente existentes e extraídos das bases de dados não estruturadas. Dados poderão ser relacionados, alterados e até simulados de forma rápida através do uso de outras ferramentas que usarem o resultado do processo de extração de informações descrito nesta dissertação.

A seguir são apresentados os objetivos intermediários da ferramenta proposta, a fim de atingir seu objetivo final.

- **Sistemas de Simples Utilização**

Todos os usuários de computadores devem ser capazes de utilizar o sistema e manipular as informações que têm disponíveis a nível local, mesmo sem ter grandes conhecimentos de computação.

- **Acessar Bases de Dados Não Estruturadas/Semi-Estruturadas**

A fim de atingir o objetivo final de manipulação e conversão de bases de dados não estruturadas, o sistema deve prover funções de leitura de dados armazenados de forma textual em bases de dados não estruturadas ou semi-estruturadas. Tais funções têm que se adaptar às diversas características de armazenamento em forma de texto, como, por exemplo, trabalhar com arquivos que utilizam diferentes caracteres como separadores de linhas.

- **Possuir Flexibilidade de Utilização**

As necessidades do usuário podem variar de momento para momento, conforme seu interesse de pesquisa e suas fontes originais de dados. Desta forma, deve ser permitido o tratamento de diversos tipos de informação, ou seja, assuntos, línguas e outras características intrícicas ao conteúdo de um arquivo, destacando-se as informações centrais referentes a estes arquivos de entrada.

- **Manipular Grande Volume de Dados**

O sistema deve ser capaz de manipular volumes de dados considerados grandes a nível local, diminuindo o tempo gasto na absorção da informação armazenada, bem como no seu entendimento. No entanto, a definição do domínio dos dados de entrada deve ser configurada conforme as necessidades momentâneas do usuário.

- **Apresentar Flexibilidade de Evolução**

As informações a serem selecionadas a partir das bases de dados, e os formatos destas bases podem variar conforme as necessidades momentâneas do usuário. Além disso, com o passar dos anos e o surgimento de novas tecnologias, os formatos de armazenamento podem se alterar. Desta forma, o sistema deve ser flexível a tais mudanças, permitindo sua utilização nas mais diversas situações.

- **Tratar Aspectos Temporais**

Neste trabalho, a análise dos aspectos temporais é um ponto chave na extração das informações, visto o protótipo proposto. Desta forma, o sistema deve apresentar um

conjunto de características e funções próprias para a manipulação deste tipo de dado, como, por exemplo, identificação de diversos formatos de datas.

- **Apresentar uma Saída de Dados Facilmente Exportável**

As informações resultantes do processo de descoberta do conhecimento devem ser armazenadas em um formato estruturado que facilite sua exportação para outros sistemas de tratamento de dados, como, por exemplo, bancos de dados temporais. Desta forma, a manipulação do conhecimento será facilitada.

- **Apresentar uma Saída de Dados Independente**

Apesar da necessidade de exportar os dados resultantes do sistema SES para outros sistemas, estes resultados devem ser armazenados em um formato independente, que possibilite sua manipulação direta pelo usuário quando desejado, mesmo sem o uso de outras ferramentas. Desta forma, o usuário deve, por exemplo, poder abrir o arquivo de dados resultante, entender seu conteúdo e alterá-lo de forma "manual".

- **Possuir Precisão no Processo de Extração**

O sistema deve descartar as informações sem interesse para o usuário e destacar os aspectos centrais dos arquivos de entrada, viabilizando o entendimento das informações de forma ágil e fácil. No entanto, a seleção e filtragem dos dados deve ser realizada detalhadamente, viabilizando um bom grau de precisão e qualidade nas informações extraídas.

2.8 Restrições e Limitações

Na busca de um sistema factível, algumas limitações do mesmo devem ser descritas.

- **Velocidade de Processamento**

O tempo gasto na extração de dados não estruturados deve ser aceitável conforme o tipo e o nível de detalhamento e precisão desejado e definido pelo usuário. No entanto, a extração de informações de maneira precisa exige um processamento complexo e demorado. Isto gera o aumento do tempo necessário para análise dos dados em relação a outros sistemas já existentes para informações distribuídas em rede, como, por exemplo, o WAIS.

- **Dados Locais**

A utilização do sistema a nível local (em um computador pessoal) é um dos objetivos do sistema. No entanto, este objetivo é, ao mesmo tempo, uma limitação, pois o tratamento de um volume de dados muito grande, como o existente a nível de rede, seria proibitivo devido ao tempo de processamento.

- **Limitação de Domínio**

Os vários formatos de armazenamento da informação, bem como, os diferentes assuntos a serem tratados pelo sistema, formam diversos domínios de extração. Neste sentido, para que seja possível a análise dos diferentes arquivos, o usuário deverá definir cada um destes domínios dentro do sistema.

No entanto, a definição de um domínio representa o armazenamento do conhecimento do próprio usuário, o qual deve ser declarado da forma mais coesa e fechada possível, a fim de limitar o escopo de extração e viabilizar o processo de descoberta do conhecimento. Isto porque domínios muito amplos podem gerar processamentos de longa duração, bem como, a redução da precisão de extração.

3 Processos de Extração

O estudo sobre Extração de Informações em textos tem se tornado um ponto de pesquisa e interesse mundial devido à importância do texto como uma base de dados não estruturada ou semi-estruturada.

A fim de ampliar o embasamento conceitual deste trabalho e justificar o processo de extração proposto nesta dissertação, foi aprofundado o estudo das três principais técnicas envolvidas no desenvolvimento de sistemas de extração de informações atualmente: análise estatística, análise sintática e sistemas baseados em conhecimento [COW 96]. Tais técnicas exploram diferentes características dos textos com o objetivo de explorar seu conteúdo central (tema).

No entanto, a busca de novas técnicas ou variações das existentes é algo em constante avanço, a fim de melhorar os resultados e a qualidade das informações extraídas dos textos, como se propõe esta dissertação. O objetivo é utilizar os conhecimentos já existentes na área de IE para a geração de técnicas melhoradas e com resultados de maior qualidade.

Um aspecto interessante sobre os textos, e problema central para os pesquisadores da área de IE, é que a semântica de um texto não é bem representada por características superficiais como palavras individuais [CRO 93]. Conseqüentemente, a performance e a eficácia de extração da informação armazenada de forma automática é algo de grande complexidade, normalmente, com resultados abaixo dos satisfatórios.

3.1 Processo Estatístico

Muitas pesquisas e sistemas de extração e recuperação de informações atuais são baseados somente em termos simples para a representação do conteúdo de textos, consistindo normalmente na avaliação de palavras individuais. Nesta avaliação, é usado um enfoque estatístico, ou seja, a exploração das propriedades estatísticas do texto, como a observação da freqüência em que as palavras ocorrem no mesmo. Cabe salientar que um termo pode ser composto por uma ou mais palavras.

A partir da freqüência com que uma palavra ocorre em um texto, pode lhe ser atribuída um peso, permitindo a comparação entre as palavras que compõem este texto. Desta forma, as palavras com maior peso são as palavras-chave na descrição do conteúdo global deste texto no momento de uma consulta. Os resultados de recuperação dependem crucialmente da escolha de pesos efetivos para os termos no sistema.

Normalmente, 50% de um texto é formado por palavras que são menos utilizadas para recuperação de informações. Estas palavras são ditas de conexão ou reiteração como, por exemplo, pronomes, artigos, conjunções, verbos auxiliares, adjetivos quantitativos, etc. [LAN 68]. Desta forma, pode-se ignorar tais palavras quando aparecerem no corpo de um texto, pois não servem para caracterização do conteúdo deste, já que aparecem tão comumente em todos os texto. Dependendo do

campo específico de assunto dos textos sobre avaliação, as palavras a serem ignoradas podem variar.

Eliminando as palavras não significativas, ordena-se o restante das palavras do texto em ordem alfabética e realiza-se a contagem da freqüência de ocorrência de cada palavra. Algumas vezes também pode-se controlar a freqüência de pares de palavras ou um conjunto de palavras fisicamente justapostas no texto (termos). O resultado desta contagem será uma listagem ordenada de freqüência dos termos no texto. Os termos de maior freqüência serão utilizados como identificadores deste texto, pois devem representar o conteúdo semântico do mesmo.

A quantidade de termos utilizados para identificação dos textos é realizada conforme o índice de *recall* e *precision* desejado nas consultas. Normalmente defini-se um número fixo de termos de identificação ou especifica-se uma percentagem de termos conforme a quantidade total de termos que compõem o documento.

Cabe salientar que existem algumas variações do processo estatístico básico descrito acima; contudo, o núcleo (características principais) deste processo mantém-se o mesmo.

A estratégia de recuperação segundo esta técnica é facilmente implementada e razoavelmente efetiva [CRO 93]. No entanto, esta estratégia não leva em conta a relação entre as palavras do texto, a fim de obter novas informações intrínsecas ao contexto do documento como um todo. Conjuntos de termos simples não podem prover completa identificação do conteúdo de documentos. Por esta razão, várias outras pesquisas na análise de conteúdo de textos têm sido propostas ao longo dos anos como uma forma de gerar representações de textos para recuperação de informação, como descreve o item a seguir.

3.2 Processo Sintático

Ultimamente, os avanços em sistemas de recuperação de informações têm se baseado na modelagem do problema da análise da linguagem natural [SAL 88]. As pesquisas destes métodos devem ser usadas para a extração do conteúdo central de documentos e consultas, e para relacionar a informação desejada pelo usuário com as bases de dados existentes [MET 89].

Sistemas de análise sintática vêm freqüentemente sendo utilizados para extrair unidades de identificação completas, como nomes próprios ou preposições, a partir de textos. Isto destaca a relação entre as palavras, ultrapassando os limites do conhecimento contido puramente no termo simples [LEV 88]. Desta forma, obtém-se um conjunto de informações maior sobre cada palavra que compõe o texto, pois a classificação sintática de uma palavra em uma frase amplia e até altera o seu significado individual. Por exemplo, na frase "João joga bola", a palavra "João" ganha um novo atributo através da análise sintática, não detectado em análises de termos simples, pois poderá ser classificada como um nome próprio.

Pode-se dizer que o processamento sintático converte uma frase "plana" em uma estrutura hierárquica que corresponde às unidades de significado sintático da frase. Este processo é chamado de análise (*parsing*). Embora haja muitas maneiras de produzir uma análise, quase todos os sistemas usados têm dois componentes principais segundo [RIC 93]:

- **Gramática**

A maneira mais comum de representar a gramática é como um conjunto de regras de produção. Embora variem os detalhes das formas permitidas nas regras, a idéia básica permanece a mesma. Como está ilustrado na FIGURA 3.1, que mostra uma estrutura gramatical. Leia a primeira regra como: Uma frase é composta por um sintagma nominal seguido de um sintagma verbal . Nesta gramática, a barra vertical deve ser lida como "ou". O ε denota a cadeia de caracteres vazias. Os símbolos que são ainda mais expandidos pelas regras são chamados de símbolos não terminais. Os símbolos que correspondem diretamente às cadeias de caracteres que têm de ser encontradas em uma frase de entrada são chamados de símbolos terminais.

Os formalismos das gramáticas fundamentam muitas teorias lingüísticas, que por sua vez dão base para muitos sistemas de compreensão da linguagem natural.

F → SN SV
SN → o SN1
SN → PronPess
SN → SubstPr
SN → SN1
SN1 → S LocAdj
LocAdj → Adj | LocAdj
SV → Vi
SV → Vtd SN
Subst → arquivo | impressora
SubstPr → Mário
PronPess → Eu
Adj → pequeno | longo | rápido
Vtd → imprimiu | criou | quero
Vi → corre

FIGURA 3.1 - Gramática

- **Analisador**

O processo de análise usa as regras da gramática e compara-as com a frase de entrada. Cada regra coincidente acrescenta algo à estrutura completa que está sendo criada para a frase. A estrutura mais simples que se pode criar é a árvore gramatical (*parse tree*), que simplesmente registra as regras e como elas são comparadas. A

FIGURA 3.2 mostra a árvore gramatical que seria produzida para a frase "Mário imprimiu o arquivo", usando-se a gramática acima.

Observe que cada nó da árvore gramatical corresponde a uma palavra da frase de entrada ou a um símbolo não terminal da gramática. Cada nível da árvore gramatical corresponde a aplicação de uma regra da gramática.

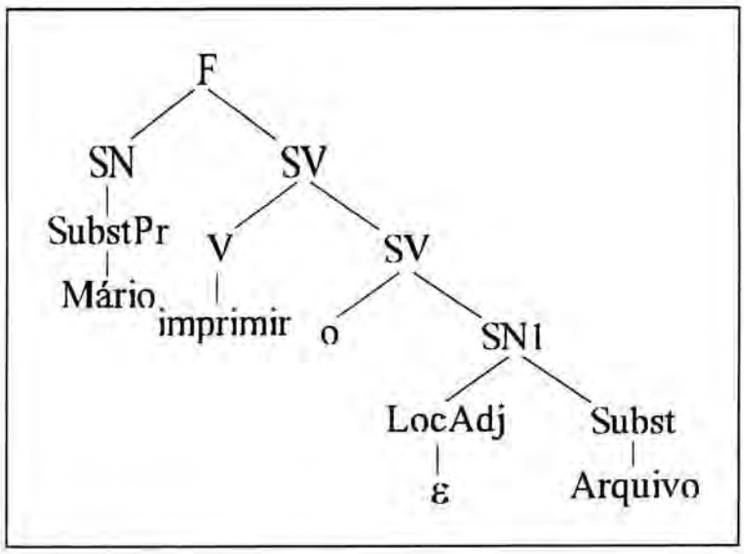

FIGURA 3.2 - Árvore Gramatical

É importante destacar que programas de análise sintática são normalmente grandes e demandam o armazenamento de uma grande quantidade de termos, bem como a necessidade de equipamentos com maior poder de processamento, impedindo a análise de grandes quantidades de textos.

Na prática, métodos sintáticos são freqüentemente aplicados em um modo "seguro de falhas"; ou seja, analisando textos particulares, onde a informação completa sobre cada palavra do texto pode não estar avaliada, e certas regras gramaticais podem ser violadas [SAL 88]. Normalmente, quando a quantidade de arquivos é grande e analisadores sintáticos são utilizados, o resultado é pobre. Neste caso, o uso de métodos estáticos é preferido. Alternativamente, um analisador mais simples pode ser usado, concentrando-se em certas passagens do texto em preferência a outras.

Um texto contém componentes sintáticos, bem como semânticos. Desta forma, uma mesma linguagem de representação ou conjunto de palavras pode ser utilizado para descrever diferentes situações, dependendo do conteúdo do texto como um todo, ou o contexto propriamente dito. Além disso, a estruturação das frases e o vocabulário baseados no conhecimento individual do autor exige a inclusão deste conhecimento no sistema que pretende realizar a análise do documento.

Neste sentido, a sintaxe por si só não pode resolver muitas ambigüidades que complicam a tarefa de análise de conteúdo, fazendo com que os métodos baseados puramente no entendimento da sintática não sejam suficientemente poderosos para produzir uma análise apropriada dos textos. Características contextuais, ou dependentes

do discurso e conhecimento são necessários para esta proposta. Desta maneira, novos sistemas auxiliares têm sido considerados, incluindo especialmente informações adicionais sobre palavras a partir de dicionários, ampliando o vocabulário armazenado, e informação obtida a partir de bases de conhecimento, preparadas para refletir propriedades semânticas de áreas particulares. Conforme descrito em [SAL 88], comentamos estes sistemas auxiliares a seguir.

3.2.1 Uso de Informações de Dicionário

Muitas das falhas do método sintático sugerem que a análise correta do conteúdo de um texto pode depender da possibilidade de informação adicional relacionada com os termos individuais e suas inter-relações. Uma possibilidade consiste no uso de descritores de termos contidos em dicionários, provendo a exatidão dos termos que formam a frase. Esta informação pode ser usada para retirar a ambigüidade de significado de termos e, além disso, gerar grupos de similaridade entre palavras pela identificação de relações entre o contexto de várias entradas do dicionário, por exemplo *Thesaurus*.

Várias atenções têm se focalizado para extração de informações muito utilizadas a partir de dicionários, e a experiência indica que muitos termos relacionados são relativamente fáceis de obter, como, por exemplo:

- **sinônimos**: normalmente identificados de forma explícita no dicionário;
- **hierarquias**: princípios de relações entre termos que são identificados, seguindo a análise das definições do dicionário. Como exemplo de hierarquia pode-se ter a palavra carro, onde no dicionário esta definida como "veículo". Assim, pode-se obter informações sobre uma palavra a partir da definição de outra.

No entanto, ainda encontram-se muitos problemas no uso de dicionários, tais como:

- muitos termos contidos no dicionário apresentam mais de uma definição, gerando dificuldades para a escolha da definição que melhor se aplica a um dado caso;
- a definição utilizada em um termo de um arquivo de entrada pode ser de difícil análise, neste caso, o significado da definição contida no dicionário pode se tornar obscuro no contexto do arquivo sob análise;
- as relações entre diferentes definições do dicionário podem ser difíceis de serem avaliadas de forma automática.

3.2.2 Construção de Bases de Conhecimento

Grande parte das atenções tem sido tomada quanto à incorporação de bases de conhecimento construídas manualmente a partir de assuntos de áreas particulares na análise automática de textos. A base de conhecimento é uma representação abstrata dos

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

tópicos de uma área ou de um ambiente particular, incluindo os principais conceitos de interesse desta área e várias relações entre estas entidades. Diversos modelos são usados para representação de conhecimento, incluindo redes semânticas, bem como coleções de estruturas lógicas e *scripts*. Quando uma base de conhecimento é utilizada, representando o assunto de uma área particular, as seguintes extensões de estratégias de recuperação de informação podem ser usadas:

- a consulta para recuperação da informação é analisada sobre uma forma padrão de representação para comparação com a base de conhecimento;
- uma operação de combinação *fuzzy* é realizada para comparar a consulta formalizada com elementos do conhecimento representado;
- uma resposta para a consulta requerida é construída se o grau de combinação entre a base de conhecimento e a consulta é suficientemente boa.

No entanto, o primeiro problema na representação do conhecimento é a própria representação. É normalmente impossível isolar pedaços particulares de conhecimento de uma forma totalmente contida e fechada em si própria. Isto é, a interpretação de pedaços particulares requer não somente o conhecimento de um assunto local, mas também um amplo contexto que não é geralmente contido em bases de conhecimento particulares. Neste caso, o texto de entrada pode não ser corretamente comparado com uma estrutura de conhecimento incompleta.

Outro problema central com a representação do conhecimento é a dificuldade de análise e retirada de ambigüidade dos textos processados, além da transformação destes textos em uma forma comparável padrão para avaliação com a base de conhecimento. Normalmente, quando os termos de entrada são corretamente analisados, o vocabulário de entrada pode ser muito diferente das especificações da base de conhecimento, complicando as regras de inferências e podendo então ser necessária a relação entre diferentes formas de conhecimento armazenadas.

3.3 Sistemas Especialistas e Baseados em Conhecimento

Sistemas de Extração de Informações sobre BDNE/SE podem apresentar características importantes de sistemas especialistas. Segundo [MIC 82], um sistema especialista pode ser definido como um sistema no qual é personificada a organização do conhecimento referente a uma área especifica de um especialista humano, suficiente para realizar consultas com habilidade e eficiência.

Diferentes dos programas convencionais, os sistemas especialistas podem resolver problemas que exigem julgamentos do mesmo tipo que as pessoas se utilizam em seus trabalhos diários. Alguns destes sistemas, como alguns sistemas tradicionais, fornecem respostas em termos de porcentagem de certeza, propagando dentro do programa graus de confiança associados às diversas informações. Alguns sistemas têm capacidade de dar explicações, que demonstram como as respostas foram encontradas, aumentando a credibilidade que os usuários neles depositam.

Em suas capacidades, projetos e forma de operação, sistemas especialistas e baseados em conhecimento, segundo [GEN 86], diferem profundamente dos sistemas convencionais. Provavelmente, as maiores diferenças residam na habilidade dos primeiros simularem raciocínio humano, inferirem e fazerem julgamentos, freqüentemente com informações incompletas, enquanto que os últimos efetuam tarefas puramente mecânicas e processam dados, ainda que em alta velocidade. Outra diferença é que aqueles sistemas derivam suas conclusões e soluções de heurísticas, baseadas em um domínio específico de conhecimento, enquanto que os convencionais geram seus resultados através de algoritmos. Também estes trabalham exclusivamente com números e caracteres; enquanto que sistemas especialistas e baseados em conhecimento trabalham com símbolos e conceitos.

É importante destacar que compreender como a mente humana realmente trabalha na solução de problemas especializados não é necessário para produzir, com êxito, sistemas especialistas que aumentem a capacidade e a produtividade do homem. É suficiente ser capaz de extrair o conhecimento de um ou mais especialistas e de estruturar este conhecimento em uma representação computacional uniforme que permita a aplicação de métodos consistentes de processamento em computador.

O desenvolvimento de um típico sistema especialista se inicia com o trabalho de um engenheiro de conhecimento que exaustivamente entrevista uma reconhecida autoridade em um campo particular (domínio específico) e codifica a perícia obtida em regras e fatos. Depois de representado simbolicamente, o conhecimento extraído é transportado para um computador que eletronicamente repete análises peritas e estratégias de solução de problemas.

No entanto, sistemas especialistas e baseados em conhecimento, apesar de semelhantes, diferem levemente um do outro. O conhecimento do domínio do problema, no caso de sistemas baseados em conhecimento, é organizado separadamente de outros tipos de conhecimento do sistema, como os procedimentos de resolução de problemas ou de interação com o usuário. Esta coleção de conhecimento especializado é chamada base de conhecimento, e os procedimentos gerais de solução de problema, de máquina de inferência. Desta forma, uma mesma máquina de inferência pode ser utilizada com diferentes bases de conhecimento, as quais pertencem a domínios específicos distintos.

A base de conhecimento de um sistema especialista contém fatos (dados) ou regras (fatos condicionais), os quais são usados como base para tomada de decisão. A máquina de inferência decide como aplicar as regras e em que ordem, a fim de deduzir novos conhecimentos.

A solução de um problema por um sistema especialista pode ser vista como um fluxo de raciocínio, que vai das evidências (fatos e regras) para as conclusões. Grande parte da habilidade dos especialistas humanos na solução de problemas reside na capacidade de estreitar o campo das soluções possíveis a partir de cada informação adicional recebida. Da mesma forma, um sistema especialista, dotado de um modelo de inferência bem projetado, fará uso das diversas evidências, combinando-as em hipóteses intermediárias sobre conclusões parciais, a partir das quais os resultados serão, finalmente, deduzidos.

3.3.1 Representação do Conhecimento

A representação do conhecimento é a estrutura formalizada e o conjunto de operações que envolve a descrição, relacionamento e procedimento de um determinado domínio. Neste sentido, procura-se métodos de representação que sejam poderosos e, além disso, sejam aplicáveis de maneira genérica na solução e descrição de uma ampla classe de problemas úteis.

Para solucionar os problemas complexos encontrados na área de inteligência artificial (IA), é preciso uma grande quantidade de conhecimento e também alguns mecanismos para a manipulação desse conhecimento a fim de criar soluções para novos problemas. Uma série de maneiras de representar o conhecimento (fatos) tem sido explorada em programas de IA. Contudo, é preciso considerar que um bom sistema de representação de conhecimento em um determinado domínio deve possuir as seguintes quatro propriedades segundo [RIC 93]:

- **Adequação Representacional**: capacidade de representar todos os tipos de conhecimento necessários naquele domínio.

- **Adequação Inferencial**: capacidade de manipular as estruturas representacionais de modo a derivar novas estruturas que correspondam a novos conhecimentos, inferidos a partir de conhecimentos antigos.

- **Eficácia Inferencial**: capacidade de incorporar à estrutura de conhecimento informações adicionais que podem ser usadas para focalizar a atenção dos mecanismos de inferência nas direções mais promissoras.

- **Eficácia Aquisitiva**: capacidade de adquirir novas informações facilmente. O caso mais simples envolve a inserção direta, por meio de uma pessoa, de novos conhecimentos na base de dados. Idealmente o próprio programa deveria ser capaz de controlar a aquisição de conhecimentos.

Infelizmente, ainda não foi encontrado um único sistema que otimize todas as capacidades para todos os tipos de conhecimento. Como resultado, existem várias técnicas para a representação do conhecimento, e muitos programas baseiam-se em mais de uma técnica.

A seguir serão apresentadas duas técnicas bastante comuns de representação que serviam de embasamento conceitual para atingir-se os objetivos desta dissertação.

3.3.1.1 Conhecimento Relacional Simples

O modo mais simples de representar fatos declarativos é como um conjunto de relações do mesmo tipo usado em sistemas de banco de dados. A TABELA 3.1 mostra um exemplo deste tipo de sistema relacional, neste caso tratando sobre jogadores de Beisebol.

Esta representação é simples porque, sendo independente, fornece capacidades inferenciais muito fracas. Por outro lado, o conhecimento representado desta forma pode servir como entrada para mecanismos de inferência mais poderosos. Dados apenas os fatos da TABELA 3.1, por exemplo, não é possível nem mesmo responder a esta simples pergunta: "Quem é o jogador mais pesado?". Contudo, se for fornecido um procedimento para que se descubra qual o jogador mais pesado, então estes fatos permitirão ao procedimento gerar a resposta. Se, no entanto, tivermos um conjunto de regras para decidir que rebatedor deve enfrentar um determinado lançador (com base, digamos, no fato de eles serem canhotos ou destros), então esta mesma relação pode proporcionar pelo menos parte das informações exigidas pelas regras.

TABELA 3.1 - Conhecimento Relacional Simples

Jogador	Altura	Peso	Rebate-Lança
João	1,80	90	Direita-Direita
Carlos	1,75	85	Direita-Direita
Mário	1,85	105	Esquerda-Esquerda
Pedro	1,87	100	Esquerda-Direita

Os sistemas de banco de dados são projetados para proporcionar suporte ao conhecimento relacional. Muitos produtos comerciais já solucionaram as questões práticas que surgem quando se liga um sistema de banco de dados que fornece este tipo de suporte a um sistema de representação de conhecimento que fornece qualquer um dos outros recursos de inferência.

3.3.1.2 Conhecimento Procedimental

O conhecimento procedimental, também chamado de operacional, pode ser representado em programas de várias maneiras. A mais comum é simplesmente como código de programação através de linguagens, como LISP, onde é descrito como fazer algo.

Desta forma, o conhecimento é utilizado quando se executa o programa para a realização de uma tarefa. Infelizmente, esta forma de representar o conhecimento procedimental não é bem qualificada com relação às propriedades da adequação inferencial (porque é muito difícil escrever um programa que consiga raciocinar sobre o comportamento de um outro programa) e eficiência aquisitiva (porque o processo de atualizar e depurar programas grandes torna-se complicado).

Devido a esta dificuldade de raciocínio com linguagens de programação comuns, foram feitas tentativas para encontrar outras maneiras de representar conhecimentos procedimentais, para que estes sejam de manipulação relativamente fácil, tanto por outros programas quanto por pessoas.

A técnica mais comumente usada para a representação de conhecimento procedimental em programas de IA é a das regras de produção. Esta representação provê um conjunto modular de regras na forma situação/ação. Um exemplo de representação comum e muito natural de expressar conhecimento são regras de produção do tipo "Se [condição] Então [ação]" [WEI 88]. A FIGURA 3.3 mostra um exemplo de regra de produção que representa certos conhecimentos operacionais que todo jogador de beisebol normalmente possui.

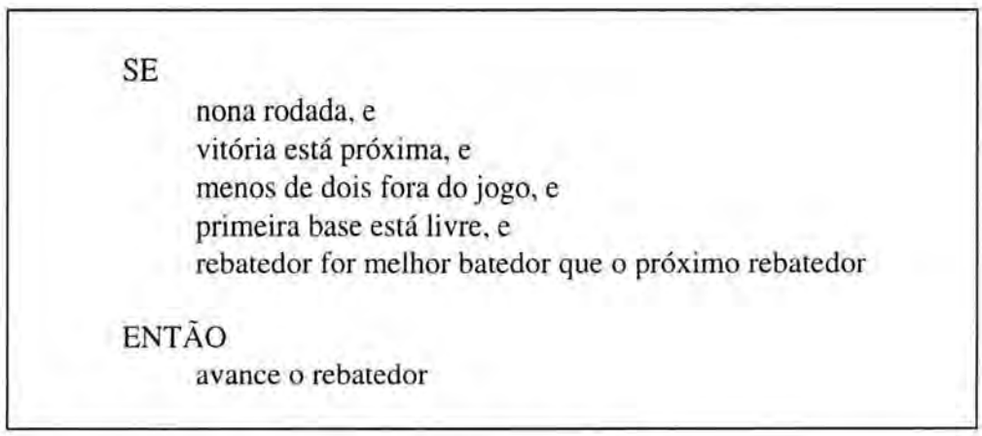

FIGURA 3.3 - Regra de Produção

4 Extração Semântica

Um leitor pode adquirir conhecimento a partir de um texto, facilmente e naturalmente, identificando as informações relevantes, e memorizando-as. Contudo, automatizar esta atividade é tão complexo quanto construir um sistema que entenda a linguagem natural [MOU 92].

A busca do conhecimento a partir de bases de dados não estruturadas e semi-estruturadas, como textos, exige o entendimento dos dados armazenados, transformando-os em informação de forma automática. Desta maneira, um processo de extração da informação é necessário a fim de simular, com a maior proximidade possível, o papel do leitor ao entender o conteúdo de um texto.

Neste sentido, propõe-se e apresenta-se um método de extração do conhecimento, o qual é descrito e conceituado neste capítulo. Este método utiliza-se do embasamento conceitual gerado nos estudos apresentados nos capítulos anteriores, sendo, agora, este estudo colocado de forma mais específica ao objetivo desejado.

O processo de extração de informações proposto não visa a construção de um processo que entenda a linguagem natural. Ao contrário, visa, através da convergência das técnicas de sistemas baseados em conhecimento (SBC) e sistemas de extração de informações (IE), uma maneira automatizada para auxiliar o usuário na extração de informações a partir de BDNE/SE. Chamaremos este processo de **Extração Semântica**, visto que este apresenta ao usuário um conjunto de dados menor que o armazenado, permitindo-lhe avaliar e entender o conteúdo das bases de dados armazenadas, não sendo objetivo do sistema o entendimento semântico de um texto.

A Extração Semântica fornecerá ao usuário os dados contidos nas BDNE/SE conforme uma filtragem prévia, seleção e processamento destes dados. Com isto, o método de extração proposto é, na realidade, auxiliado pelo próprio usuário, pois a análise e o entendimento dos dados resultantes do sistema ficam a cargo deste. No entanto, é importante destacar que este auxílio só é válido se os resultados da análise dos arquivos de entrada, identificando e extraindo as informações principais de cada texto, forem de boa qualidade, conforme as necessidades do próprio usuário.

Segundo [HED 95], um conjunto de ferramentas computacionais, como redes neurais, sistemas baseados em regras e programas estatísticos, podem ser aplicadas na solução do problema de descoberta do conhecimento. Neste trabalho, serão utilizadas as técnicas descritas anteriormente (capítulo 3) com um importante objetivo: dar ênfase aos aspectos cognitivos envolvidos na leitura de um texto, ou seja, entender como as pessoas recuperam a informação armazenada em bases de dados não estruturadas/semi-estruturadas.

É com este objetivo que o próprio conhecimento do usuário (neste caso, como leitor) de como extrair as informações relevantes a partir dos arquivos de entrada será utilizado: "**conhecimento para extrair conhecimento**".

4.1 Sistema Baseado em Conhecimento

Conforme o objetivo do processo de extração semântica, observamos que, na realidade, este processo pode ser classificado como um Sistema Baseado em Conhecimento (SBC). Isto confirma o que é dito por [TUR 91]: sistemas de recuperação e extração de informações muitas vezes requerem a adaptação de modelos de inferência de sistemas baseados em conhecimento para a tarefa de análise de documentos.

Segundo esta visão de SBC, temos um conjunto de **Eventos de Entrada** que, processados por uma **Máquina de Inferência**, geram uma **Conclusão**, a qual é extraída a partir dos conhecimentos armazenados na **Base de Conhecimento**, como é apresentado na FIGURA 4.1.

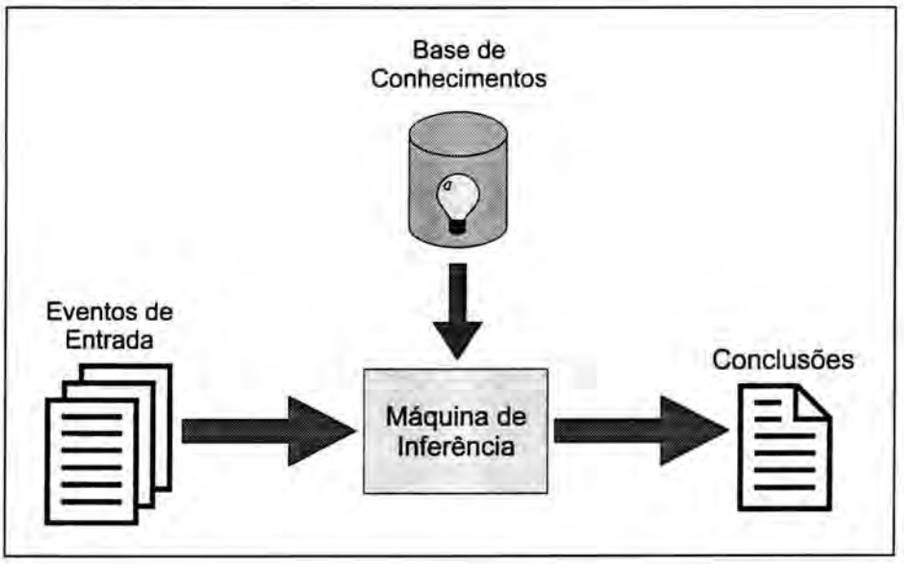

FIGURA 4.1 - Arquitetura Genérica de um Sistema Baseado em Conhecimento

Deve-se salientar, contudo, que a técnica proposta para o desenvolvimento deste estudo (SBC) e a estatística, não são opostas. A estatística viabiliza a análise de grandes bases de dados, com uma vasta variedade de domínios de assuntos, visto sua maior velocidade de recuperação a partir de um processamento mais simples. No caso das técnicas baseadas em conhecimento, que usam um processamento normalmente mais complexo, é viabilizada a aplicação em domínios mais restritos, porém com uma qualidade de seleção da informação maior (*precision*), e um tempo de processamento razoável.

É importante salientar que o desenvolvimento do protótipo do sistema proposto, levando em conta o uso das técnicas de sistemas baseados em conhecimento, só é possível se não esquecidas algumas características apresentadas na definição dos objetivos do sistema (capítulo 2) e colocadas a seguir de forma mais específica para com a área de sistemas baseados em conhecimento (capítulo 3):

- **Restrição de Domínio**

O conhecimento a ser extraído terá origem em de bases de dados não estruturadas ou semi-estruturadas locais, ou seja, no mesmo computador. Isto indica que, normalmente, estas bases se referem, de forma muito próxima, a um mesmo assunto, ou que assuntos diferentes estejam organizados separadamente, em diretórios distintos, por exemplo.

Outro fato importante quanto a restrição de domínio é que muitas das BDNE/SE que servirão de entrada ao processo de extração semântica, provavelmente tenham advindo de um processo prévio de seleção, como servidores WAIS (*Wide Area Information Server*), que utilizam um processo estatístico de seleção de informações para indexação. Desta forma, o domínio de assuntos torna-se mais restrito e homogêneo.

Esta restrição do domínio permitirá o uso de bases de conhecimento mais específicas e mais completas quanto ao domínio de extração, viabilizando o uso de técnicas baseada em conhecimento.

- **Volume de Informação**

Estando as bases de dados em um repositório local, o volume de informação a ser processado será muitas vezes menor do que o existente em meio externo, como a Internet [WYA 95] [ALC 95]. Isso possibilita o uso de técnicas de extração do conhecimento mais complexas e conseqüentemente mais lentas quanto ao tempo de processamento.

- **Seleção e Filtragem de Precisão**

Os arquivos de entrada, como já dito, normalmente pertencerão a um mesmo domínio (assunto, características estruturais, etc.), muitas vezes advindos de um prévio processo de filtragem. Desta forma, o extrator semântico poderá realizar uma seleção e filtragem sobre as bases de entrada de uma forma mais detalhada e restritiva, ou seja, aumentando a precisão dos dados resultantes no arquivo de saída, conforme as necessidades do usuário.

- **Flexibilidade**

Este é um ponto muito importante do processo de extração, pois o uso de bases de conhecimento permitirá ao usuário definir o domínio de recuperação da informação conforme suas necessidades e as características das bases de dados de entrada.

- **Múltiplas Bases de Conhecimento**

Como característica de sistemas baseados em conhecimento, é possível que uma mesma máquina de inferência utilize diversas bases de conhecimento diferentes. Assim, pode-se definir múltiplos domínios de recuperação, utilizados conforme as necessidades do usuário e as características das bases de dados a serem analisadas no momento.

É importante destacar que as diferentes representações do conhecimento, através do uso de múltiplas bases, podem enfatizar diferentes características dos textos a serem analisados, mesmo quando aplicadas sobre os mesmos arquivos de entrada.

4.1.1 Máquina de Inferência

Seguindo a estrutura de um sistema baseado em conhecimento (FIGURA 4.1), temos a máquina de inferência que no sistema proposto será representada pelo módulo que irá realizar a **extração semântica.** Este módulo tem por finalidade analisar os arquivos de entrada (bases de dados não estruturadas e semi-estruturadas) conforme o conteúdo das bases de conhecimento definidas pelo usuário. Como resultado, será gerado um arquivo de saída.

A máquina de inferência apresenta entre suas funções, a tarefa de interpretar regras, as quais servem para armazenar o conhecimento necessário à extração. Tais regras, contêm conhecimento sobre características léxicas, sintáticas e cognitivas (como o usuário extrai a informação das BDNE/SE) que definem as informações a serem selecionadas e filtradas a partir dos arquivos de entrada. Neste sentido, efetua-se a análise de um texto em relação ao conjunto de termos individuais que o compõe e, também, com respeito a estrutura sintática expressa pelo agrupamento e relacionamento entre estes mesmos termos.

Além disso, para que o processo de análise dos textos ocorra, a máquina de inferência necessita realizar a função de abrir os arquivos de entrada e efetuar sua leitura palavra a palavra de maneira ordenada. Desta forma, a máquina de inferência é subdivida em três sub-módulos principais: leitor de arquivo, interpretador de conhecimento, analisador semântico.

4.1.2 Eventos de Entrada

Os eventos de entrada fazem parte da estrutura de um Sistema Baseado em Conhecimento, sendo que, dentro do sistema de extração proposto, estes eventos são representados pelo conjunto de **Arquivos de Entrada** que compõem a base de dados a ser analisada pela máquina de inferência. Cada arquivo contém um conjunto de características e informações que serão analisadas conforme as definições do usuário, contidas na base de conhecimento, as quais darão margem ao processo de inferência para filtragem e seleção da informação.

4.1.3 Conclusão

A conclusão é caracterizada por um **Arquivo de Saída**. Este arquivo no formato texto contém o resultado da análise de todos os arquivos que compõem a base de dados de entrada. Desta forma, dentro do arquivo de saída existirão as informações filtradas e selecionadas pela máquina de inferência de maneira estruturada.

4.1.4 Base de Conhecimento

Esta base é composta por um conjunto de arquivos de dados, armazenando o conhecimento de como realizar a extração de informações a partir de BDNE/SE.

A representação do conhecimento de como filtrar e selecionar as informações dos arquivos de entrada, será realizada de uma forma procedimental, conforme embasamento apresentado em 3.3.1.2. Neste sentido, utiliza-se para representação um conjunto de regras com a estrutura situação/ação. Esta maneira de representar o conhecimento é bastante comum e muito natural para expressar ações.

É importante ressaltar, contudo, o fato de que o sistema proposto será utilizado sobre um domínio de conhecimento restrito, o que permitirá e facilitará a definição do contexto representado de maneira formal.

As regras, na realidade, representam conhecimentos empíricos, gerados a partir do conhecimento do usuário. Estes conhecimentos referem-se a utilização de um grupo de características léxicas e sintáticas exploradas para a seleção das informações de entrada. Desta forma, o sistema processa a análise dos textos levando em conta conhecimentos cognitivos, ou seja, o conhecimento envolvido na leitura de um texto por um ser humano.

4.2 Extração de Informações dos Arquivos de Entrada

Conforme as necessidades envolvidas em um sistema de Extração de Informações para bases de dados não estruturadas, objeto de estudo desta dissertação, buscou-se agregar características das duas técnicas de extração descritas inicialmente no capítulo 3 - estatística e sintática - com o uso de técnicas de Sistemas Baseados em Conhecimento. O objetivo é a exploração léxica e sintática do arquivo de entrada através do uso do conhecimento do próprio usuário do sistema em extrair dados (aspecto cognitivo).

As regras armazenas na base de conhecimento permitem a configuração do extrator semântico conforme o conjunto de características léxicas e sintáticas que o usuário irá explorar para extração das informações desejadas.

4.2.1 Características Léxicas

Quanto às características léxicas, o extrator semântico buscará obter o conhecimento a partir de termos, ou seja, a partir das palavras que compõem os arquivos de entrada. Neste sentido, o texto é percorrido palavra a palavra, analisando-as uma a uma. Esta qualidade de obter conhecimento através de termos individuais é usada no processo estatístico, onde os termos que ocorrem com maior freqüência em um texto servem como identificadores deste texto.

Neste caso, no entanto, apesar de igualmente extrair informações armazenadas em termos, o processo de extração proposto apresenta algumas características distintas,

resultando em uma avaliação léxica dos termos. A seguir são apresentadas as análises realizadas pelo sistema proposto para extração das características léxicas dos arquivos de entrada:

- **Simples Ocorrência**

Cada palavra extraída do texto deve ser comparada com a base de dados definida pelo usuário, verificando se esta palavra serve para identificação e seleção do conteúdo do texto. Desta forma, a simples ocorrência de uma palavra pode auxiliar na extração de informações armazenadas nas BDNE/SE, trazendo consigo um significado semântico importante dentro do contexto global do texto. Pode-se ter, por exemplo, a palavra "*deadline*" como sendo um termo importante na análise de textos cujo assunto é Chamadas de Artigos para Congressos. Desta forma, a partir da informação extraída e retornada ao usuário, este poderá saber, detalhes importantes do texto sob análise. Além disso, é relevante destacar que a existência de uma determinada palavra no arquivo de entrada pode desencadear um processo de análise sintática, conforme descrito no item 4.2.2.

- **Características Léxicas**

Existem cadeias de caracteres ou palavras cujo significado só pode ser extraído a partir de um processo de análise mais profundo, caractere a caractere, e/ou realizando a comparação com padrões pré definidos de palavras. Um exemplo é a cadeia de caracteres "10/10/95", onde a formatação dos números identifica a palavra como uma data. Esta identificação atribui um importante conhecimento semântico sobre a cadeia de caracteres recuperada.

- **Termos Compostos**

Deve-se observar que um termo pode ser composto por uma ou mais palavras. Neste sentido, na busca do conhecimento a partir da análise léxica, é importante destacar que muitas palavras podem ter seu significado obtido de forma correta pelo extrator somente quando combinadas com palavras posteriores ou anteriores. Um exemplo é a combinação em inglês *Call for Paper*. Estas palavras individualmente, podem não apresentar um significado relevante na extração do conhecimento de uma base de dados, ou até podem gerar uma análise errônea. Contudo, combinadas, apresentam um conteúdo semântico único.

4.2.2 Características Sintáticas e Relações de Termos

Pessoas claramente tomam informações sobre o conteúdo de um texto (isto é, uma sentença) tanto a partir do conteúdo contido em palavras individuais do texto, como a partir da estrutura na qual as partes individuais estão contidas [MET 89]. A sintaxe reflete esta estrutura e, sendo assim, este processo de extração objetiva o uso da informação sintática em complemento à informação baseada na presença de termos. Isto aumenta a precisão do sistema de extração proposto.

No entanto, neste trabalho, o uso da informação sintática para a recuperação de informação é baseado na relação entre termos de sentenças e frases. Não será realizada a

identificação de elementos sintáticos como nomes próprios. O objetivo da análise das características sintática é encontrar as propriedades estruturais de como um termo individual esta inserido dentro de um documento.

Na análise sintática um termo é colocado como cabeça de pesquisa, a partir do qual os demais termos relacionados serão extraídos. Normalmente, este termo cabeça é resultado de uma análise léxica prévia, destacando uma palavra como importante dentro do contexto do documento conforme as definições do usuário. O objetivo é ampliar o significado atribuído ao termo cabeça na análise léxica através da extração das informações sintáticas associadas a este termo.

Os demais termos envolvidos na análise sintática são chamados termos dependentes, ou seja, são os termos relacionados com o termo cabeça. O escopo das relações entre termos deve ser definido pelo próprio usuário através do conjunto de regras que fazem parte da base de conhecimento.

Esta forma de pesquisa sintática apresenta como motivação e base os trabalhos de [MET 89] e [SAL 68], cujos objetivos são semelhantes aos desejados nesta fase do processo de extração: obtenção do conhecimento armazenado em múltiplos termos.

Cabe lembrar que várias palavras iguais podem ocorrer em documentos diferentes, combinadas de formas distintas, ou seja, por exemplo, dois textos, ou duas sentenças, contêm os mesmos termos, mas não se referem ao mesmo conceito. Neste sentido, o significado de cada uma das ocorrências destes termos depende em muito da relação dentre os mesmos em cada ocasião, ou, até mesmo, da relação com outros termos próximos.

Conforme o exemplo do item anterior (4.2.1), definiu-se a cadeia de caracteres "10/10/95" como uma data. No entanto, esta data deve ter um significado no contexto global do texto sob análise. Com o uso das características sintáticas, podemos associar esta data com a palavra que a antecede, por exemplo: "*deadline*". Isto atribui, a partir do contexto, um significado semântico maior à cadeia de caracteres "10/10/95".

Num escopo maior, um segundo conjunto de regras, aplicado sobre outra parte do texto, poderá definir seu tema central, por exemplo: uma chamada para o envio de artigos a um congresso. Desta forma, a partir da informação extraída do arquivo de entrada e retornada ao usuário, este poderá saber, com um certo grau de certeza, sobre o que trata o texto e a data final de envio dos artigos.

É importante lembrar que a descoberta do conhecimento contido em um texto tende a funcionar melhor com termos que não são tão raros, nem tão freqüentemente encontrados no documento sob análise [MET 89]. Desta forma, quando termos individuais são muito comuns, é muito provável identificar combinações destes termos que são muito mais específicas em seu significado, e ocorrem menos freqüentemente nos arquivos de entrada como um todo, por exemplo a palavra DATA. Esta palavra pode aparecer muitas vezes no texto, mas seu significado passa a ser melhor definido no contexto do documento quando associada a outras palavras dependentes, como: Data de Inscrição, Data de Entrega, ou Data de Avaliação.

Como pode ser observado, existe uma interdependência entre os procedimentos envolvidos na análise léxica e sintática do texto. No que concerne ao seu funcionamento, o analisador sintático é requisitado pelo léxico, sempre que este identificar no arquivo de entrada um termo dado como relevante pelo usuário através da base de conhecimento. Por exemplo, no caso da data, o analisador léxico recebeu do leitor de arquivos a palavra "10/10/95", destacando-a como relevante e realizando a interpretação do significado semântico associado ao termo. A partir deste significado inicial, o analisador léxico requisita uma análise sintática, ampliando o entendimento semântico do termo inicial e de outros termos relacionados a este.

4.2.3 Dicionário de Palavras do Sistema

Durante a análise das características léxicas de um texto, as palavras que devem ser encontradas no documento a fim de identificar seu conteúdo, estão armazenadas na base de conhecimento. No entanto, tais palavras podem ser declaradas de duas formas. A primeira e mais simples, diretamente nos comandos de representação do conhecimento (Se [palavra] então [ação]).

A segunda forma de definição das palavras de pesquisa é realizada através do uso de uma base de dados, denominada Dicionário de Palavras do Sistema, composta por um conjunto de arquivos onde estão armazenadas todas as palavras identificáveis pelo analisador léxico. Cada um destes arquivos pode conter um grupo de palavras relacionadas entre si, como, por exemplo, nomes de países, formando um *thesaurus*.

A facilidade de flexibilizar a junção entre os termos da base de conhecimento com os dos documentos desejados, conforme é possibilitado pelo uso do dicionário, traz muitas vantagens ao processo de extração com um todo. Isto ocorre porque a pobre conjunção do vocabulário usado para expressar o conhecimento de extração e o vocabulário usado nos documentos aparecem como sendo a maior causa de baixo *recall*. Desta forma, o *recall* pode ser melhorado usando domínios de conhecimento específicos, como os países, através do dicionário de palavras, sem significante degradação de precisão. Além disso, o uso do dicionário gera uma relação entre termos, simplificando a manipulação deste conhecimento armazenado como parte da base de conhecimento.

4.2.4 Classificação do Arquivo de Entrada

A fim de aumentar a qualidade da informação recuperada e diminuir o volume de conhecimento armazenado nas bases de conhecimento, uma avaliação semântica prévia do arquivo de entrada é realizada através de seu nome. Observando cada convenção de nome de arquivo, podem ser determinados os tipos de arquivos com um alto grau de acerto [HAR 93]. Como exemplo, pode-se observar a extensão dos nomes de arquivos fontes na linguagem de programação C, ou seja, a extensão ".c".

O extrator identifica o tipo de cada um dos arquivos a serem analisados, explorando convenções de extensões de nomes. Esta classificação permite a geração de grupos de arquivos baseados nos diferentes tipos dos arquivos de entrada. Desta forma, arquivos semi-estruturados são facilmente agrupados conforme as diferentes estruturas

internas de armazenamento de dados. Enquanto isso, os arquivos sem uma estrutura padrão, como normalmente são os arquivos com extensão TXT, também formam um outro grupo.

A geração destes grupos de arquivos viabiliza o uso de regras de extração específicas para cada um dos formatos de entrada, aumentando a eficiência do processamento e, principalmente, levando em conta características estruturais dos tipos, a fim de destacar as informações relevantes em cada arquivo [GIF 91]. Como exemplo de arquivos ".c", podemos destacar uma característica comum quanto aos delimitadores de comentários: "/*" e "*/".

Neste sentido, todos os arquivos de entrada, antes de serem processados quanto ao seu conteúdo interno pelo extrator semântico, passam por um prévio processamento de classificação.

4.2.4.1 Base de Conhecimento Auxiliar

Segundo a estrutura de um sistema baseado em conhecimento, a base de conhecimento auxiliar, aqui definida, é parte da base de conhecimento descrita anteriormente. No entanto, trataremos esta base de forma separada, pois ela esta relacionada com o processo de classificação do arquivo de entrada.

A base de conhecimento auxiliar tem o objetivo de guardar informações, associando às extensões dos nomes dos arquivos de entrada com as bases de conhecimento correspondentes necessárias ao processo de extração. Este relacionamento é usado para a classificação dos arquivos de entrada, selecionando o conjunto de regras, ou seja, a base de conhecimento propícia ao processo de extração semântica, conforme a extensão do arquivo de entrada. O próprio usuário fornece os dados armazenados nesta base auxiliar.

Na realidade, dentro da base de conhecimento auxiliar está guardada uma tabela com as extensões dos nomes de arquivos de entrada e as respectivas bases de conhecimento para extração. Esta forma de armazenar conhecimento, conforme embasamento conceitual apresentado em 3.3.1.1, utiliza uma estrutura relacional, como podemos observar no exemplo da TABELA 4.1.

4.2.4.2 Pré Processamento de Arquivos de Entrada

Existem arquivos de entrada que contêm outros arquivos inseridos dentro de sua estrutura. Exemplos deste tipo de estrutura compreendem arquivos comprimidos, formato ZIP e ARJ, e arquivos de *News*. Além disso, pode haver arquivos que devem ser convertidos para o formato texto, passando por um pré-processamento, a fim de possibilitar a extração das informações, como, por exemplo, arquivos em *PostScrip*, muito comumente encontrados na Internet.

TABELA 4.1 - Associação entre Extensões de Nomes de Arquivos e Bases de Conhecimento

Extensão	Arquivo de Conhecimento
TXT	Conhecimento para Arquivo Texto
HTM	Conhecimento para Arquivo HTML (*HyperText Markup Language*)
C	Conhecimento para Arquivo Fonte da Linguagem C
MAI	Conhecimento para Arquivo de Correspondência Eletrônica

Quando é realizada a classificação do arquivo de entrada, sabe-se qual base de conhecimento utilizar no processo de extração. Contudo, o usuário do extrator pode chamar um programa executável externo ao extrator, somente ativado por este. O objetivo é realizar um pré processamento do arquivo de entrada. A extração de arquivos que exijam um pré-processamento é feita utilizando um programa correspondente ao tipo de estrutura existente, conforme detectado pelo classificador. Por exemplo, pode-se ativar um programa que divida um arquivo de *News* em vários arquivos menores contendo cada uma das mensagens do arquivo original separadamente.

Este tipo de pré-processamento do arquivo de entrada é apresentado no sistema Essence (item 2.3.3.3), o qual determina que um arquivo é na verdade uma estrutura de arquivos agrupados, extraindo os arquivos "escondidos". Posteriormente, é explorado o tipo dos arquivos resultantes, conforme sua extensão e gerado o sumário para eles. Este processo continua recursivamente até não serem encontrados mais arquivos agrupados.

4.3 Regras de Extração

Seguindo a linha procedimental para armazenamento de conhecimento, mais especificamente, uma estrutura na forma SE [condição] ENTÃO [ação], definiu-se um conjunto de regras que satisfizessem os objetivos de extração semântica através de característica léxicas e sintáticas dos textos.

O texto de entrada a partir do qual a informação será extraída será percorrido do início ao fim ou conforme definição do usuário, na busca de termos, os quais também chamaremos de *tokens*, que ativem as ações especificadas nas regras da base de conhecimento. Neste sentido, o extrator funciona como um compilador de textos, lendo o arquivo de entrada, comparável ao código fonte, e gerando um arquivo de saída (código executável).

4.3.1 Granularidade de Pesquisa

A granularidade de pesquisa é um ponto importante para a extração da informação. Ela representa a unidade na qual a informação contida no texto será pesquisada. Neste trabalho, definiu-se que o texto seria percorrido com uma granularidade a nível de palavras, ou seja, o texto é lido pelo sistema palavra a palavra, e estas tornam-se os *tokens* de entrada para as regras da base de conhecimento.

Esta granularidade de pesquisa foi escolhida porque permite um bom detalhamento do texto para o processo de extração. Características léxicas contidas em palavras poderão ser pesquisadas utilizando-se a granularidade a nível de palavra, e, combinando-se as palavras lidas a partir do texto original pelo sistema, serão extraídos componentes sintáticos pela relação de palavras em termos compostos e frases.

Assim sendo, é proporcionada uma boa flexibilidade de configuração da extração pelo usuário através da base de conhecimento, pois este pode configurar o extrator utilizando as palavras como unidades de informação para extração do conteúdo do texto de entrada. Como exemplo, pode-se ter a palavra *from* no contexto de uma correspondência eletrônica. O usuário poderá configurar o extrator semântico para extrair todas as palavras subseqüentes ao *from*, utilizando uma palavra como ponto de referência para a extração do endereço do remetente da correspondência.

Granularidades menores, como caracteres, por outro lado, exigiriam um tempo maior de extração semântica e uma maior complexidade das regras, pois estas deveriam permitir a combinação dos caracteres na busca de *tokens* maiores: palavras, por exemplo. Além disso, o texto deveria ser pesquisado caracter a caracter. No entanto, o uso de uma granularidade de pesquisa a nível de palavras não impede que estas palavras tenham tamanho de um caractere, como o artigo 'a', ou, então, que o sistema extraia do texto original um simples caractere. Como exemplo, pode-se ter que ao encontrar-se a palavra *from* em um texto, o caractere seguinte deva ser extraído, como o caractere ':'.

Contudo, neste trabalho, a fim de ampliar a flexibilidade de extração e a precisão sem um aumento prejudicial do tempo de extração e da complexidade de configuração das regras que compõem a base de conhecimento, ainda é permitida uma granularidade de pesquisa a nível de caractere. Para tal, a parte condicional das regras de extração foi dividida em duas partes menores, sendo que a primeira parte trabalha com uma granularidade a nível de palavras e a segunda a nível de caractere.

A primeira parte da condição será chamada de **condição de pesquisa** e a segunda de **condição de verificação**. Sempre que a condição de pesquisa for verdadeira a condição de verificação será testada, podendo-se ai usar uma granularidade baseada em caracteres. Como exemplo, pode-se ter uma condição de pesquisa com a palavra *from* combinada com uma condição de verificação com o caractere ':'. Neste exemplo, sempre que a palavra *from* for encontrada, será verificado se o caractere subseqüente é o ':', e, somente se a condição de verificação for verdadeira, a ação contida na parte ENTÃO da regra será executada. A seguir será apresentada de forma mais detalhada tal estrutura condicional proposta.

4.3.2 Condições

As condições contidas nas regras da base de conhecimento definem quando as ações associadas serão executadas. Neste sentido, cada *token* extraído do arquivo de entrada é comparado com as condições definidas pelo usuário, verificando se existe alguma ação a ser realizada.

No entanto, na busca de um processo de extração de melhor qualidade de precisão, e de um sistema de fácil utilização, separou-se a condição em duas partes relacionadas e complementares.

4.3.2.1 Condições de Pesquisa

As condições de pesquisa também podem ser chamadas de "cabeças" de pesquisa, pois todo o *token* de entrada é comparado com todas as condições de pesquisas existentes nas regras da base de conhecimento, verificando-se as verdadeiras, para que somente estas sejam testadas quanto a condição de verificação.

As condições de pesquisa são baseadas em termos com uma ou mais palavras, os quais são comparados com os *tokens* gerados a partir da leitura do texto de entrada, palavra a palavra. Neste sentido, como já foi dito anteriormente, a condição de pesquisa utiliza uma granularidade a nível de palavra.

No caso de termos com mais de uma palavra, a primeira palavra do termo é comparada e, caso esta comparação seja verdadeira, é verificado se o *token* seguinte do texto também é verdadeiro, e assim por diante até a última palavra do termo existente na condição de pesquisa.

No entanto, testar todas as condições de pesquisa para cada palavra extraída do texto de entrada tornaria o processo de extração muito lento. Além disso, existem regras cuja validade só existe em determinado trecho do arquivo de entrada, como, por exemplo, extrair o *subject* de um *mail*. Esta informação, por exemplo, esta sempre localizada nas primeiras linhas do arquivo, não sendo necessário que a regra que extrai o assunto de um *mail* seja testada a partir de determinado ponto do texto.

Neste sentido, todas as funções de teste que compõem as condições de pesquisa apresentam uma entrada chamada faixa, onde o usuário poderá informar, em linhas, o escopo de validade daquela função. Caso a faixa não seja informada, assume-se uma validade da condição de pesquisa sobre todo o texto.

Os termos de comparação contidos nas condições de pesquisa (CP) podem ser declarados de quatro maneiras diferentes:

- ***String***

Termos declarados diretamente na condição de pesquisa para comparação com os *tokens* de entrada. É permitido definir se a caixa dos caracteres que compõem o termo é livre, ou seja, não influi no processo de comparação; ou conforme especificada pelo usuário na regra.

- Exemplos:

  string([termo],[caixa],[faixa_início],[faixa_fim])

  SE CP(String('universidade', CaixaLivre))
  ENTÃO .....

  SE CP(String('universidade federal', CaixaLivre))
  ENTÃO .....

  SE CP(String('Rio', CaixaEspecificada))
  ENTÃO .....

- **Base de Dados ou Dicionário**

Os termos para comparação com os *tokens* de entrada estão declarados em um dicionário de termos, que pode conter um ou mais termos, sendo que cada termo pode ser composto de uma ou mais palavras (item 4.2.3). Caso o *token* de entrada exista no dicionário de termos a condição é verdadeira. Para tal todas as palavras do dicionário são comparadas com o *token*.

O uso do dicionário de palavras tem como objetivo facilitar a comparação das palavras do texto com um grupo de palavras relacionadas entre si, como, por exemplo: nomes de países.

- Exemplos:

  BaseDados([base],[caixa],[faixa_início],[faixa_fim])

  SE CP(BaseDados(BaseDePaises, CaixaLivre))
  ENTÃO .....

  SE CP(BaseDados(BaseDeUniversidades, CaixaEspecificada))
  ENTÃO .....

- **Data**

Esta função verifica se o *token* de entrada é uma data válida, conforme um conjunto de padrões de formatos de datas pré-definidos no extrator semântico, tais como: Dia/Mês/Ano. A escolha dos padrões implementados no sistema é baseada nos padrões utilizados nos editores de textos mais utilizados em ambiente Windows (Word [MIC 95] e WordPerfect[WOR 93]) e levantamento realizado no NEWS da Universidade Federal do Rio Grande do Sul.

Esta forma de verificação tem como objetivo a busca de informações temporais, caso específico de aplicação e validação do sistema descrito e embasado nesta dissertação.

- Exemplo:

Data([caixa],[faixa_início],[faixa_fim])

SE CP(Data())
  ENTÃO .....

- **Livre**

Neste caso, a condição de pesquisa sempre é dada como verdadeira independentemente do *token* de entrada. Desta forma, permite-se ao usuário realizar o processo de extração com granularidade de pesquisa baseada em caracteres, pois a condição de pesquisa sempre será verdadeira, sendo levado em conta somente a comparação realizada na condição de verificação.

- Exemplo:

Livre([faixa_início],[faixa_fim])

SE CP(Livre())
  ENTÃO .....

Cabe salientar que uma condição de pesquisa pode ser composta, ou seja, podem ser especificadas varias condições para a mesma ação. Desta forma, as diferentes condições são ligadas por um operador do tipo OU.

- Exemplo:

SE CP(string('Porto Alegre', CaixaLivre) ou
string('Rio', CaixaEspecificada))
  ENTÃO .....

4.3.2.2 Condições de Verificação

As condições de verificação servem como complemento das condições de pesquisa, sendo testadas somente quando as condições de pesquisa forem verdadeiras.

O objetivo é permitir uma melhor avaliação do *token* de entrada quanto a outros termos relacionados ao termo testado na condição de pesquisa. Desta forma, são analisadas as características sintáticas que envolvem um termo do arquivo de entrada.

Utilizando o exemplo do item 4.2.2, pode-se ter uma condição de pesquisa que verificou a existência de uma cadeia de caracteres "10/10/95", identificando-a como uma data. No entanto, esta data deve ter um significado no contexto global do texto sob análise. Com o uso das condições de verificação, o usuário do sistema busca associar esta data com outros termos relacionados existentes no texto, por exemplo, com a palavra que supostamente a antecede: "*deadline*". Isto atribui, a partir do contexto, um significado semântico maior à cadeia de caracteres "10/10/95", viabilizando,

posteriormente, um banco de dados (arquivo de saída) mais consistente com relação ao texto original.

- Exemplo:

```
SE CP(Data()) and
        CV(string('deadline',CaixaLivre,1,palavra,ANT))
   ENTÃO .....
```

As condições de verificação sempre estão associadas às condições de pesquisa, sendo a verificação relacionada com o termo encontrado na condição de pesquisa, na busca de uma extração mais precisa. Neste sentido, não existe, normalmente, validade ou necessidade de análises realizadas por condições de verificação em escopos distantes do termo inicialmente identificado pela condição de pesquisa. Sendo assim, as funções de comparação em condições de verificação apresentam delimitadores da distância de verificação, tendo como referência a condição de pesquisa associada.

Este controle de distância permite ajustar o nível de relacionamento entre termos do arquivo de entrada dentro do processo de extração semântica. Distâncias maiores, ampliam o escopo de relacionamento das palavras, mas podem gerar avaliações erradas, associando palavras com pouca ou nenhuma relação. Verificações em distâncias menores retornam palavras, normalmente, mais fortemente relacionadas, devido a sua proximidade no texto. Contudo, outras palavras também relacionadas com o termo localizado na condição de pesquisa e importantes no contexto semântico do texto podem ser excluídas do processo de extração.

O controle da distância é realizado dentro de cada função de verificação (exemplo anterior), permitindo que para cada verificação associada a uma mesma condição de pesquisa, existam distâncias diferentes, conforme as características a serem avaliadas pelo usuário do sistema no processo de extração. Como exemplo, pode-se ter a palavra *from*, onde deseja-se verificar se após ela existe o caractere ':' com 1 palavra de distância e o caractere '@' com 1 linha de distância. Este exemplo, permitirá concluir se o *from* encontrado é ou não, com um bom grau de certeza, o *from* que identifica o remetente em uma correspondência eletrônica.

As distâncias, visando a obtenção de uma maior flexibilidade de uso, podem ser medidas nas seguintes unidades:

- Palavras
- Linhas (até um caractere de quebra de linha)
- Frases (até um caractere '.')
- Parágrafos (até encontrar dois ou mais caracteres de quebra de linha subseqüentes)

Intrinsecamente à distância, está associado o sentido, onde é caracterizado se a condição de verificação deve ser testada na parte do texto de entrada anterior ou posterior à condição de pesquisa avaliada como verdadeira. Sendo assim, o usuário, juntamente com a distância, deverá configurar o sentido na geração das condições de verificação que compõem a base de conhecimento.

Por outro lado, é importante salientar que algumas características do texto só poderão ser válidas ao processo de extração quando ocorrerem mais de uma vez, como, por exemplo, na seguinte situação: verificada a palavra "período" em uma condição de pesquisa, é desejado que a condição de verificação associada avalie a existência de duas datas até o final da linha. Estas datas informariam ao usuário o início e o fim do período sob extração.

A fim de facilitar este processo de verificação repetitiva, as funções de verificação apresentam o número de vezes na qual uma determinada condição deve ocorrer dentro da distância estabelecida igualmente na função. Esta característica permite a redução do número de condições de verificação dentro da mesma regra, bem como auxilia no processo de declaração das regras da base de conhecimento.

Os termos de comparação contidos nas condições de verificação (CV) podem ser declarados de quatro maneiras diferentes, conforme as funções abaixo.

- **Caracter**

Como dito anteriormente, a condição de verificação pode trabalhar com uma granularidade a nível de caractere, permitindo um processo de análise mais detalhado do arquivo de entrada, apesar de mais lento. O uso desta característica é feito através da função de comparação Caractere.

É permitido definir a caixa do caracter sob verificação (maiúsculo ou minúsculo), conforme é feito na condição de pesquisa. A caixa do caracter será levada em conta na avaliação da regra.

- Exemplo: verificar se após a palavra *from* existe o caractere ':' com uma distância máxima da condição de pesquisa de uma palavra depois.

Caractere([termo],[caixa],[vezes],[distância],[unidade],[sentido])

```
SE CP(String('from', CaixaLivre)) and
        CV(Caractere(':',CaixaLivre,1,1,palavra,DEP))
   ENTÃO .....
```

- *String*

Termos declarados diretamente na condição de verificação para comparação com os *tokens* de entrada.

- Exemplos: verifica se existe a palavra *deadline* antes de uma data.

String([termo],[caixa],[vezes],[distância],[unidade],[sentido])

```
SE CP(Data()) and
      CV(String('deadline',CaixaLivre,1,1,palavra,ANT))
   ENTÃO .....
```

- **Base de Dados ou Dicionário**

Semelhante à função usada para condições de pesquisa, a função BaseDados em uma condição de verificação permite que os termos para comparação com os *tokens* de entrada estejam declarados em um dicionário de termos, que pode conter um ou mais termos, sendo que cada termo pode ser composto de uma ou mais palavras (item 4.2.3). Caso o *token* de entrada exista no dicionário de termos a condição de verificação é verdadeira. Para tal todas as palavras do dicionário são comparadas com o *token*.

- Exemplos: verifica se após a palavra 'local' existe o nome de uma cidade cadastrada na base de dados CIDADES.

BaseDados([base],[caixa],[vezes],[distância],[unidade],[sentido])

```
SE CP(String('Local', CaixaLivre)) and
        CV(BaseDados('cidades',CaixaLivre,1,1,frase,DEP))
   ENTÃO .....
```

- **Data**

Esta função verifica se o *token* de entrada é uma data válida, conforme um conjunto de padrões de formatos de datas pré-definidos no extrator semântico.

- Exemplo:

Data([caixa],[vezes],[distância],[unidade],[sentido])

```
SE CP(String('deadline', CaixaLivre)) and
        CV(Data(CaixaLivre,1,1,frase,DEP))
   ENTÃO .....
```

Cabe salientar que uma condição de verificação pode ser combinada com outras, ou seja, podem ser especificadas varias condições de verificação para uma mesma condição de pesquisa. Desta forma, as diferentes condições são ligadas por um operador do tipo AND.

- Exemplo:

```
SE CP(String('from',CaixaLivre)) and
        CV(Caractere(':',CaixaLivre,1,1,palavra,DEP) and
            Caractere('@',CaixaLivre,1,1,linha,DEP))
   ENTÃO .....
```

4.3.3 Ações

Conforme o modelo condicional de regra utilizado neste trabalho, uma ação A é executada quando a condição que compõe a regra for verdadeira (então); ou, caso contrário, uma ação B pode ser executada quando esta condição for falsa (senão).

Contudo, neste trabalho a condição é dividida em duas partes: pesquisa e verificação. Neste sentido, definiu-se que tanto a ação positiva (então) quanto a negativa (senão) só seriam executadas quando a condição de pesquisa fosse verdadeira. Desta forma, caso esta primeira condição seja verdadeira, fica a cargo da condição de verificação definir qual das ações será realizada, conforme o resultado do teste condicional de verificação.

Esta restrição de executar as ações somente quando a condição de pesquisa for verdadeira é que viabiliza o uso de dois tipos de ação: então e senão. Caso contrário, as regras da base de conhecimento cuja condição não contivesse todos os possíveis *tokens* lidos do texto de entrada, ou as regras cujo resultado da condição que contivesse o *token* fosse falso, teriam a ação de senão (negativa) executada. Como, normalmente, a grande maioria das palavras do texto não estão previstas nas regras da base de conhecimento, as ações negativas de todas as regras seriam quase sempre executadas.

Tanto as ações positivas, como as negativas, utilizam o mesmo conjunto de funções, as quais serviram para extrair a informação do texto conforme definição do usuário.

4.3.3.1 Funções de Deslocamento

A leitura do arquivo de entrada irá gerar os *tokens* necessários para a verificação das condições existentes nas regras da base de conhecimento. Pode-se dizer que o leitor utiliza-se de um ponteiro que indica a posição do próximo caractere a ser lido. Desta forma, é com base no andamento do ponteiro que a leitura do texto de entrada é realizada e os *tokens* são gerados.

No processo de extração semântica, é importante que o usuário tenha controle deste ponteiro, pois lhe será permitido controlar o andamento da leitura do arquivo de entrada. Desta forma, pedaços do arquivo poderão ser ignorados diretamente no processo de leitura, reduzindo o tempo de extração, pois as palavras deste pedaço não serão testadas pelas regras da base de conhecimento.

Além disso, muitas vezes é necessário extrair informações que já foram lidas anteriormente do texto, atrasando o ponteiro de leitura. Como exemplo, pode-se ter encontrado um *token* em uma posição X do texto; contudo, este *token* indica a existência de informações que o usuário gostaria de extrair, as quais estão em uma posição X-1.

Partindo das necessidades e justificativa acima, tornou-se importante permitir que o usuário tivesse controle sobre o ponteiro de leitura, ampliando, assim, o controle sobre a geração dos *tokens*. Desta forma, funções de deslocamento do ponteiro foram criadas para uso nas regras que compõem a base de conhecimento.

Por outro lado, é importante lembrar que uma regra pode executar mais de uma ação, ou seja, o então pode ser composto pela chamada de mais de uma função de extração. Neste sentido, definiu-se que o escopo das alterações sobre o ponteiro de leitura poderiam ser de dois tipos:

- **Restrito à regra**

Neste modo, as alterações realizadas sobre o ponteiro de leitura são válidas somente para as funções que fazem parte de uma mesma ação. Ao término da execução da última função de uma ação, o ponteiro de leitura retorna à sua posição original, antes da primeira função desta ação ser executada (FIGURA 4.2).

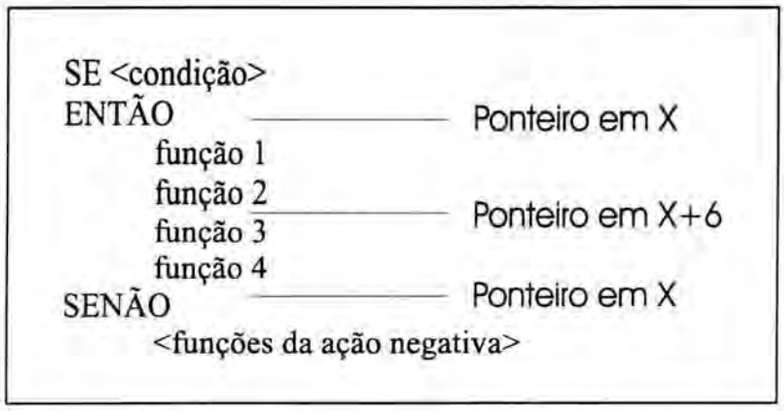

FIGURA 4.2 - Deslocamento Restrito à Regra

- **Global**

Uma alteração do ponteiro de leitura pode ter valor além do escopo da regra que realizou a alteração deste (FIGURA 4.3). Neste sentido, todas as regras executadas posteriormente a esta alteração, serão atingidas pela alteração do ponteiro de leitura.

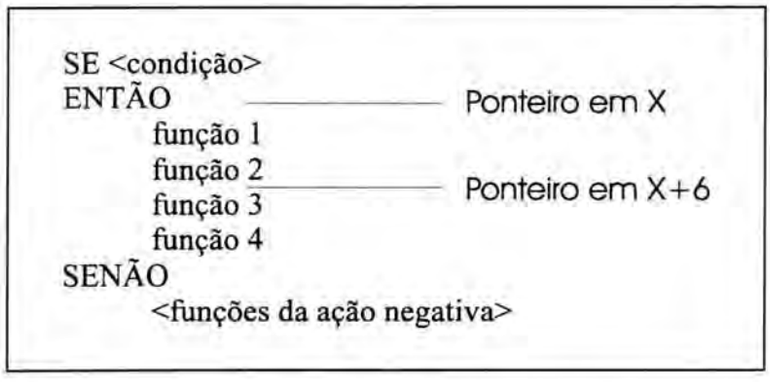

FIGURA 4.3 - Deslocamento Global

Existem duas funções que realizam o deslocamento do ponteiro de leitura, cada uma delas com suas características próprias.

- **Desloca**

A função Desloca realiza o deslocamento do ponteiro de leitura baseado em uma distância informada pelo usuário do sistema. Às distâncias podem ser medidas nas seguintes unidades:

- Palavras
- Linhas (até um caractere de quebra de linha)
- Frases (até um caractere '.')
- Parágrafos (até encontrar dois ou mais caracteres de quebra de linha subseqüentes)

Intrinsecamente à distância, está associado o sentido, onde é caracterizado se o ponteiro deve ser deslocado para o início (ANT) ou o fim (DEP) do texto a partir de sua posição atual.

- Exemplo:

Desloca([distância],[unidade],[sentido],[escopo])

SE <condição>
  ENTÃO Desloca(1,linha,DEP,GLOBAL)

- **DeslocaAte**

A função DeslocaAte realiza o deslocamento do ponteiro de leitura baseado em um ponto de referência informado pelo usuário do sistema. Este ponto de referência pode ser um caractere, um termo (*string*), uma data ou um termo de uma base de dados. Desta forma, o ponteiro é deslocado até o ponto desejado pelo usuário.

Juntamente ao ponto de referência deve ser especificada uma distância limite de pesquisa e um sentido, para que, caso o ponto de referência desejado não seja encontrado, o ponteiro de leitura desloque-se até um determinado limite. Além disso, a informação de um limite impede que a busca do ponto de referência seja realizada, por exemplo, até o fim do texto, resultando em aumento do tempo de extração e, até mesmo, deslocamentos indesejados. Cabe salientar que a distância pode ser medida em palavras, linhas, frases e parágrafos.

Adicionalmente, pode ser especificado pelo usuário o número de vezes que uma determinada referência deve ser encontrada no texto, por exemplo, deslocar o ponteiro de leitura até a terceira data encontrada no texto de entrada.

O objetivo desta função é permitir que trechos do arquivo de entrada não sejam analisados pelo extrator semântico sempre que o usuário achar que tal análise é desnecessária. Para tal, determinada condição expressa em uma regra da base de conhecimento deve ser satisfeita. Além disso, o usuário pode retroceder o ponteiro de leitura, visando extrair informações lidas anteriormente pelo extrator e que só agora são interessantes para extração. Como exemplo, pode-se ter uma frase inteira que deveria ser extraída, mas tal necessidade só surgiu ao final de sua leitura pelo processo de extração. Neste caso, o ponteiro de leitura poderá ser retrocedido, e a frase extraída.

- Exemplo:

DeslocaAte([tipo referência],[referência],[vezes],[caixa],

[distância],[unidade],[sentido],[escopo])

SE <condição>
ENTÃO DeslocaAte(*String*,'subject',1 vez,CaixaLivre,1,linha, DEP, GLOBAL)

4.3.3.2 Funções de Extração

As funções de extração têm por objetivo copiar o conteúdo selecionado e filtrado dos arquivos de entrada para o arquivo de saída.

Associado a cada função de extração existe a informação de um grupo de assunto que deve ser definido previamente pelo usuário juntamente com as regras de extração. Estes grupos de assunto tem por objetivo classificar e organizar a informação extraída, melhorando a qualidade resultante no arquivo de saída.

Existem quatro funções de extração:

- **Copia**

A função Copia realiza a extração do conteúdo do arquivo de entrada baseado em uma distância informada pelo usuário do sistema. Desta forma, todo o conteúdo entre a posição atual do ponteiro de leitura e o limite da distância informada são copiados para o arquivo de saída. As distâncias podem ser medidas nas seguintes unidades:

- Palavras
- Linhas (até um caractere de quebra de linha)
- Frases (até um caractere '.')
- Parágrafos (até encontrar dois ou mais caracteres de quebra de linha subseqüentes)

Intrinsecamente à distância, está associado o sentido, onde é caracterizado se a extração deve ser do conteúdo anterior (ANT) ou posterior (DEP) a partir da posição atual do ponteiro.

É permitido ao usuário definir se o ponteiro de leitura deve ser alterado para a posição final do conteúdo copiado do arquivo de entrada. Esta característica torna possível a não leitura do conteúdo extraído pelas funções posteriores à atual que podem existir na regra de extração. Desta forma, extraído um trecho do arquivo de entrada, o ponteiro é avançado e a próxima função de extração passa a extrair a partir desta nova posição, sem extrair informações duplicadas.

Contudo, é importante salientar que o escopo desta alteração é do tipo restrito à regra, conforme explicado no item anterior. Isto porque podem existir outras regras que têm por objetivo extrair informações diferentes das desejadas na regra atual sobre o mesmo trecho de texto, por exemplo, sobre a mesma frase uma regra extrai datas e outra

locais. Caso contrário, se o usuário desejar alterar o ponteiro em escopo global ele poderá utilizar a função Desloca ou DeslocaAte após a função Copia.

- Exemplo:

Copia([distância],[unidade],[sentido],[altera ponteiro],[assunto])

SE <condição>
ENTÃO Copia(1,linha,DEP,ALT,Locais)

- **CopiaAte**

A função CopiaAte realiza a extração do conteúdo do arquivo de entrada baseado em um ponto de referência informado pelo usuário do sistema. Este ponto de referência pode ser um caractere, um termo (*string*), uma data ou um termo de uma base de dados (dicionário). Desta forma, é extraído o trecho do arquivo de entrada entre a posição atual do ponteiro e o ponto desejado pelo usuário.

Juntamente com o ponto de referência deve ser especificada uma distância limite de pesquisa e um sentido. Caso o ponto de referência desejado não seja encontrado, a extração será realizada somente até um determinado limite, impedindo uma pesquisa até o fim do texto, aumentando do tempo de extração e, até mesmo, gerando a extração de informações indesejadas. Cabe salientar que a distância pode ser medida em palavras, linhas, frases e parágrafos.

Adicionalmente, pode ser especificado pelo usuário o número de vezes que uma determinada referência deve ser encontrada no texto, por exemplo: copiar o conteúdo do texto até a terceira data encontrada no arquivo de entrada.

O objetivo desta função é permitir que trechos do arquivo de entrada sejam extraídos com base no prévio conhecimento do usuário de que uma determinada informação contida no texto termina quando um determinado ponto de referência é encontrado. Como exemplo pode-se ter que, encontrada a palavra *from* em uma correspondência eletrônica, deva-se copiar o trecho do arquivo de entrada até a palavra *subject*, para que toda a informação de procedência seja extraída.

É permitido ao usuário definir se o ponteiro de leitura deve ser alterado para a posição final do conteúdo copiado do arquivo de entrada, conforme especificado na função anterior.

- Exemplo:

CopiaAte([tipo referência], [referência], [vezes], [caixa], [distância], [unidade], [sentido], [altera ponteiro], [assunto])

SE <condição>
ENTÃO CopiaAte(*String*,'subject',1 vez,CaixLivre,2,linhas,DEP,ALT,
Procedência)

- **CopiaCond**

A função CopiaCond realiza a cópia do termo testado na condição de pesquisa para o arquivo de saída. Muitas vezes, a simples existência de um termo já significa uma informação completa, não sendo necessária a extração de termos relacionados a este. É este o objetivo da função CopiaCond, pois a condição de pesquisa indicará a existência do termo desejado e a função CopiaCond irá copiá-lo para o arquivo de saída.

- Exemplo:

CopiaCond([assunto])

SE <condição>
ENTÃO CopiaCond(Datas)

- **CopiaTexto**

A função CopiaTexto permite que o usuário coloque no arquivo de saída informações não provenientes do texto sob processo de extração, realizando uma avaliação semântica prévia do conteúdo do texto original. O objetivo é a inclusão de conclusões geradas a partir de dados existentes no arquivo de entrada. Desta forma, o usuário, através das regras, define que, sempre que determinada condição ocorrer, uma determinada informação deve ser colocada no arquivo de saída.

Cabe salientar que, como várias funções podem ser combinadas em uma mesma regra, é possível inserir informações no arquivo de saída que sejam compostas por dados provenientes de conclusões do usuário e extraídos do texto.

- Exemplo:

CopiaTexto([conclusão],[assunto])

SE <condição>
ENTÃO CopiaTexto('Data de Início',Datas)

4.4 Arquivo de Saída

Arquivo de Saída contém o resultado da análise e processamento de todos os arquivos que compõem a base de dados de entrada (Eventos). Este resultado é formado pelas informações filtradas e selecionadas pelo **Extrator Semântico**, devendo elas estarem associadas ao nome do arquivo de origem (FIGURA 4.4). A organização proposta permite que de um mesmo **arquivo de entrada** sejam extraídas pelo sistema uma ou mais informações relevantes ao usuário. No caso de um *mail*, por exemplo (FIGURA 4.5), podem ser extraídos os dados do remetente.

Contudo, juntamente à associação da informação extraída ao nome de seu arquivo de origem, é importante poder classificar as informações em grupos de assuntos definidos pelo usuário do sistema (item 4.3.3.2). Tais grupos permitem uma melhor

organização da informação, visando facilitar o entendimento da mesma pelo usuário. Adicionalmente, esta característica de estruturação do arquivo de saída gera um relacionamento entre as informações de um mesmo grupo, sendo que, a partir desta classificação, pode-se gerar um banco de dados relacional.

Neste último exemplo, cada grupo de assunto da base de dados de saída, define um atributo (coluna) da relação (tabela), e as informações formam as tuplas (linhas).

É importante salientar que as informações resultantes do processo de extração estarão armazenadas em um arquivo texto, facilitando a exportação dos dados para outros sistemas, além de permitir sua leitura direta pelo usuário do sistema.

Arquivo	Grupo	Informação
Arq.1	Gru. A	Inform. 1
Arq.1	Gru. B	Inform. 2
Arq.2	Gru. A	Inform. 1
Arq.2	Gru. B	Inform. 2
Arq.2	Gru. B	Inform. 3
Arq.3	Gru. A	Inform. 1

FIGURA 4.4 - Estrutura do Arquivo de Saída

Arquivo	Grupo	Informação
Mail 1	Remet.	Carlos
Mail 1	Ass.	Dissertação
Mail 2	Remet.	João
Mail 2	Ass.	Cronograma
Mail 3	Remet.	Maria
Mail 3	Ass.	Congresso

FIGURA 4.5 - Arquivo de Saída Exemplo para *Mails*

5 Configuração da Extração

A geração das bases de conhecimento a serem utilizadas pelo sistema de extração semântica é um importante passo para o funcionamento do sistema como um todo. Um bom processo de extração depende fundamentalmente da quantidade, qualidade, estrutura e complexidade dos conhecimento relevantes para este fim, colocados à disposição do sistema. Sendo assim, quanto maior for o tempo e esforço dedicados à elaboração, organização e construção da base de conhecimento, melhores e mais completos serão os resultados fornecidos aos usuários.

Neste sentido, definiu-se o módulo Configurador, que apesar de não estar diretamente relacionado ao módulo de extração semântica, sendo um módulo separado do núcleo do sistema proposto, apresenta uma importante função quanto ao processo de extração de informações. O Configurador realiza a geração das bases de conhecimento.

Sendo as bases de conhecimento responsáveis pela configuração do processo de extração, o módulo de configuração deve apresentar algumas características importantes, as quais influenciam no funcionamento do sistema como um todo. Tais características são descritas a seguir.

5.1 Sistema Aberto

Conforme apresentado nos capítulos anteriores, o sistema proposto é conceitualmente definido como um sistema baseado em conhecimento, ou seja, os conhecimentos armazenados são independentes da máquina de inferência (extrator). Desta forma, diferentes bases de conhecimento podem ser utilizadas com a mesma máquina. Esta característica de independência, no entanto, não teria valor sem o módulo de configuração.

O Configurador atribui ao sistema de extração semântica uma qualidade de sistema aberto, ou seja, este pode ser alterado conforme as necessidades do usuário, obtendo um alto grau de flexibilidade, evolutibilidade e abrangência de utilização. Isto porque é através do Configurador que o sistema permite a geração de diversas bases de conhecimento, as quais serão utilizadas conforme especificação do próprio usuário. Neste sentido, é possível o uso de bases independentes por assunto, língua, tipo de informação a ser extraída, além de outras características; bem como a geração e uso de diferentes versões da mesma base.

5.2 Realimentação

Uma das qualidades mais importantes do Configurador é permitir a realimentação das informações utilizadas na extração. O processo de geração de uma base de conhecimento não será transparente para a maioria dos usuários do sistema, pois exigirá uma formalização dos conhecimentos a serem armazenados.

O usuário deverá incluir na base de conhecimento dados que possam ser aplicados para a extração das informações desejadas, sem, é claro, conhecer os arquivos de entrada de forma detalhada. Isto dificulta a definição do domínio de extração e faz com que o processo de configuração apresente um alto grau de abstração.

Esta dificuldade quanto a configuração das bases de conhecimento sugere que as primeiras versões de uma determinada base de conhecimento sejam consideradas como versões de teste ou de melhoria. Isto porque os resultados das extrações iniciais podem não ser os esperados, necessitando de modificações na versão inicial da base de conhecimento para que o extrator recupere as informações desejadas.

Desta forma, visando aprimorar os conhecimentos formalizados na base de conhecimento e, conseqüentemente, melhorar a qualidade das informações extraídas durante as subseqüentes operações de extração, os resultados das extrações iniciais devem ser examinados detalhadamente. As conclusões desta análise por parte do usuário irão gerar novas alterações da base de conhecimento sob criação, gerando um importante processo de realimentação do sistema de extração semântica. Assim, quando os resultados obtidos com o uso de uma determinada base forem bons, esta poderá ser utilizada para diversos arquivos de entrada semelhantes aos inicialmente usados para teste sem novas alterações. Como exemplo, pode-se ter uma base de conhecimento para a extração de informações a partir de correspondências eletrônicas sobre congressos.

A idéia central consiste na escolha de termos importantes ou expressões contidas em certos documentos previamente processados e que tenham sido identificadas como relevantes pelo usuário, e cuja importância destes termos em uma nova extração seja válida. Analogamente, termos, previamente incluídos para a recuperação de informações não relevantes, podem ser desenfatizados em uma futura alteração da base de conhecimento. O efeito deste processo é levar a extração em direção dos itens relevantes e contrariamente aos não relevantes, ou seja, melhorar o *recall* e a *precision*.

Cabe salientar que bases de conhecimento que se mostraram relativamente pobres na extração inicial de informações podem ser melhoradas razoavelmente pelo processo de realimentação. Já bases que produziram saídas satisfatórias na pesquisa inicial são mais difíceis de serem melhoradas, exigindo a alteração de pequenos detalhes do conhecimento armazenado, o que torna-se perigoso no momento que dados anteriormente corretos podem ser modificados.

No entanto, ao contrário do que é apresentado em [SAL 88a], fundamento deste item, o processo de realimentação utilizado no sistema proposto não é automatizado por completo. A avaliação dos resultados extraídos, bem como a alteração das bases de conhecimento, são efetuadas pelo próprio usuário.

Em conclusão, o processo de realimentação provê um método relativamente simples de reformular bases de conhecimento conforme a relevância (ou não relevância) das informações recuperadas.

5.3 Interface

Quando uma pessoa procura o auxílio do computador, normalmente ela o faz objetivamente, buscando a solução de um problema prático. Isto ocorre porque a interação com máquinas não é natural ao homem, já que este, acostumado a tratar com pessoas, geralmente não desenvolve capacidade para manusear os dispositivos físicos e lógicos usados na interação com o computador.

É nesse sentido que deve-se buscar uma interface Homem-Máquina amigável, capaz de interagir com o usuário sem amedrontá-lo, tornando-se, desta forma, a porta de entrada do sistema em desenvolvimento. Deve-se observar, ainda, que um sistema, por melhor que seja, não tem sentido se não puder ser utilizado, por esta razão é que existe, neste trabalho, tanta preocupação com a interface.

O sistema de extração será utilizado principalmente por usuários não especializados na área de informática. O projetista que se propõe a construir uma aplicação, cujo público alvo principal é constituído por pessoas com este perfil, deve ter uma atenção redobrada com o desenho e a funcionalidade da interface utilizada pelos usuários do sistema [FER 94]. Em suma, devemos construir um modelo de interface que, além de possibilitar o aproveitamento de todo o potencial do sistema, também seja de fácil utilização e compreensão por parte dos usuários. Desta maneira, acreditamos que o modelo de interface a ser adotado será fundamental para que os usuários possam explorar e aproveitar todas as facilidades implementadas pelo sistema.

Neste sentido, o módulo de configuração deve apresentar uma interface amigável, ou seja, uma interface capaz de responder ao que o usuário espera, de forma a ser aceita por ele, satisfazê-lo e adequar-se às suas necessidades. Segundo [BER 94], isto implica no uso de regras bem definidas e conhecidas pelo usuário, apresentando o mesmo tipo de informação sempre da mesma forma, e apresentar o mesmo comportamento sempre que operações de um mesmo tipo forem evocadas.

Além disso, a interface deve indicar o andamento do processo e deve permitir que o usuário obtenha ajuda ou cancele uma operação a qualquer momento, de forma a não frustrar suas expectativas.

5.3.1 Conceito

Interface Homem-Máquina é a forma como se estabelece a comunicação entre o usuário e o computador. Neste sentido, esta pode ser vista como a forma através da qual o *software* e o *hardware* suportam o diálogo Homem-Máquina, que é a troca de mensagens, ou seja, a interação propriamente dita. O restante do programa é chamado de aplicação.

Todos os processos que têm como finalidade principal realizar a comunicação entre a aplicação e o usuário pertencem à interface, bem como, todos os processos que visam solucionar problemas particulares de um usuário ou problemas genéricos de uma classe de usuários, pertencem à aplicação.

A principal tarefa de uma interface é prover o usuário com informações e facilidades que o auxiliam a fazer uso da aplicação.

5.3.2 Características Desejáveis

O usuário tende a avaliar o *software* através da sua interface, assim, uma ótima aplicação com uma interface inadequada ou insuficiente é mais facilmente rejeitada pelo usuário, que prefere utilizar um produto de baixa qualidade, mas com uma excelente interface.

Visando obter-se uma boa interface, serão aqui apresentadas as características desejáveis em uma interface segundo [FRA 93]. Tais características devem ser atingidas através da interface a ser utilizada no Configurador do sistema SES:

- **Diversidade:** A interface deve ser utilizada, de maneira conveniente, por todos os tipos de usuários, conforme suas características.
- **Complacência:** A interface deve ser tolerante com o usuário, permitindo que este se refaça de situações de erro e deve levar em consideração o esquecimento dele a informações já apresentadas.
- **Eficiência:** A execução de uma tarefa pela interface deve despender pouco esforço do usuário.
- **Conveniência:** O acesso e manuseio das operações da interface devem ser fáceis para o usuário.
- **Flexibilidade:** A interface deve oferecer várias maneiras para o usuário efetuar uma mesma operação.
- **Consistência:** A interface deve apresentar-se e comportar-se segundo regras fixas, bem definidas e conhecidas do usuário.
- **Prestimosidade:** Uma boa interface deve oferecer ajuda ao usuário, bem como alertá-lo de situações de erro que eventualmente ocorram.
- **Imitação:** O uso de exemplos, explanações, analogias, comparações, e descrições, é útil no sentido de a interface familiarizar-se ao usuário através da imitação do diálogo humano.
- **Naturalidade:** A comunicação com o usuário deve ser a mais natural possível, sem a exigência de conhecimento de termos específicos, fora de seu cotidiano.
- **Satisfação:** O usuário deve ficar, ao usar a interface, sem frustrações quanto à espera, dificuldade, falta de assistência, etc.

- **Passividade:** Preferencialmente, o controle da interação deve ficar por conta do usuário, isto implica em a interface assumir um papel passivo neste diálogo.

5.3.3 Windows 95

A fim de atingir as características apresentadas no item anterior, definiu-se que o sistema de extração semântica seria desenvolvido sobre o ambiente Windows 95, usando todas as facilidades de interface oferecidas por este sistema operacional.

Desde seu lançamento, em novembro de 1985, o Microsoft Windows vem se destacando como um dos mais conhecidos ambientes de interface gráfica [PET 92]. Milhares de cópias do programa já foram vendidas e centenas de aplicativos Windows estão disponíveis. Atualmente sua versão é a 95, lançada em agosto de 1995.

Este programa executa em 32 bits e requer o microprocessador 80486 DX ou superiores com pelo menos 8Mb de memória. Atualmente, a versão 95 é um sistema operacional independente, diferentemente das versões anteriores que executavam sobre o MS-DOS. Uma de principais características do Windows 95 é permitir o acesso a toda a memória do computador, terminando com a limitação dos 640Kb dos programas que executam diretamente sobre o MS-DOS [MIC 95a].

O Windows 95 apresenta um ambiente gráfico simulando uma mesa de trabalho. Os gráficos fazem uma melhor utilização da tela, transmitindo informações de uma maneira visual mais rica, e possibilitando o uso de uma interface dita do tipo "O que Você Vê é o que Você Obtém" (*What You See Is What You Get - WYSIWYG*) [PER 96]. Este estilo de interface demonstra especificamente ao usuário o que será obtido após a execução de uma ação. Por exemplo, muitos editores de texto exibem caracteres do tipo negrito, itálico e sublinhado, de acordo com as opções do usuário, enquanto que em outros o usuário vê apenas os códigos de controle destes. Ou seja, em caso de impressão, o resultado obtido será exatamente o que se visualiza na tela. A interface gráfica não é somente mais atraente em aparência, mas também pode transmitir um grande número de informações ao usuário.

Em uma interface gráfica, o monitor de vídeo mostra vários objetos gráficos na forma de ícones e dispositivos de entrada, como botões e barras de rolagem. Um ícone é uma representação de um objeto, ação ou propriedade. Apresentam muitas vantagens em relação a textos, pois um ícone bem projetado pode ser reconhecido mais rapidamente do que uma palavra e também ocupar menos espaço em vídeo. Ainda, se bem projetado, um ícone é independente da língua, permitindo o uso da interface em diferentes países, ou mesmo por pessoas não alfabetizadas. Usando o *mouse*, o usuário pode diretamente manipular esses objetos na tela: os objetos gráficos podem ser arrastados, os botões podem ser pressionados e as barras de rolagem podem ser ativadas.

O ambiente Windows oferece facilidades para que cada usuário ajuste seu ambiente conforme suas preferências. O usuário pode determinar as cores de sua preferência, a velocidade do teclado e do *mouse*, etc. Tudo isto pode ser feito através do Painel de Controle, programa disponível no próprio ambiente Windows e

automaticamente será válido para todos os programas que forem executados a partir deste momento.

Um programa Windows ocupa uma janela - uma área retangular na tela. Ele é identificado por uma barra de título. A maioria das funções do programa são iniciadas por meio do menu do programa. A FIGURA 5.1 mostra um típico programa Windows 95.

Neste sentido, a padronização é uma importante característica do Windows. Os programas desenvolvidos para este ambiente têm uma mesma aparência e modos de operação semelhantes, sendo portanto mais fáceis de aprender e usar do que os programas convencionais para o MS-DOS, por exemplo. Desta forma, os usuários não precisam mais gastar um longo período de tempo aprendendo a usar o computador ou a dominar um novo programa.

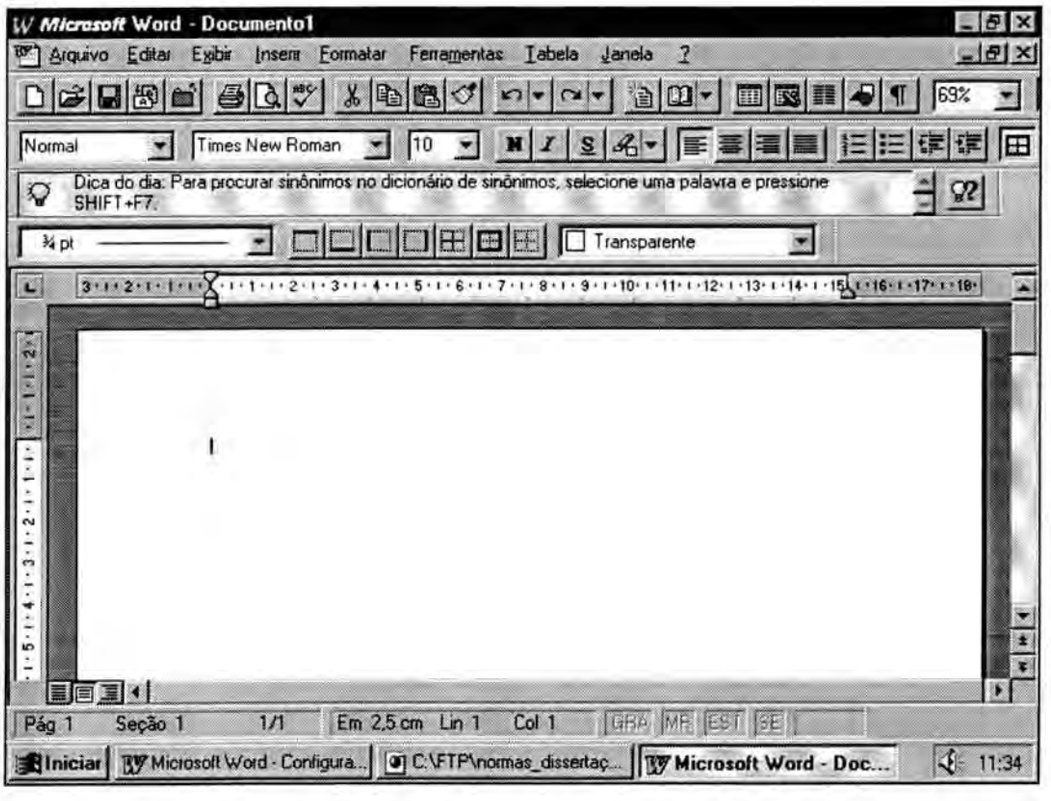

FIGURA 5.1 - Típico Programa Windows

Além das qualidades de interface, o Windows 95 é um ambiente multitarefa, onde vários programas podem ser executados e mostrados ao mesmo tempo, cada um deles ocupando uma janela diferente na tela do usuário. A comunicação entre os programas que executam sobre o Windows também é muito facilitada. Os usuários podem facilmente alternar entre os diferentes programas e transferir dados de um para outros.

Adicionalmente, o uso de características multimídia fortalece ainda mais o desenvolvimento de interfaces amigáveis, auxiliadas pelo uso de som e imagem.

5.3.4 *Parser* de Entrada

Complementarmente as características de interface oferecidas pelo ambiente Windows 95, o sistema de extração semântica deve implementar um *parser* de entrada para as regras de extração buscando atingir as características desejáveis de uma interface.

Neste sentido, propõe-se o uso de uma interface organizada através de um conjunto de formulários. O usuário preenche cada um dos campos de informação existentes para a geração das regras conforme um conjunto de opções pré-definidas no formulário. Sempre que uma entrada não prevista for realizada o sistema impede o avanço para o próximo campo e emite uma mensagem.

Desta forma, não existe a possibilidade de regras digitadas de forma errada, fugindo da gramática definida pelo sistema. Isto ocorre muito quando o usuário escreve, por exemplo, um programa fonte, onde erros de digitação ou de gramática são comuns e só são detectados normalmente no processo de compilação. Desta forma, o uso de um formulário diminui as possibilidades de erros, permitindo que o sistema consiga prever e alertar o usuário quando estes ocorrem.

No entanto, cabe salientar que esta estrutura de formulários só é bem empregada neste sistema devido à simplicidade da gramática das regras de extração de conhecimento definidas. Com isto, a eficiência de geração das bases de conhecimento é maior, sendo realizada em menor tempo e esforço do usuário.

Além disso, é importante destacar que a forte estruturação de uma interface organizada em forma de formulário, permitirá uma melhor visualização das regras sob geração, bem como uma visão global de como o extrator irá funcionar posteriormente a partir das regras geradas.

6 SES - Sistema de Extração Semântica

A fim de atingir os objetivos desejados e descritos neste trabalho, pondo em prática o embasamento conceitual apresentado anteriormente, definiu-se a arquitetura do sistema proposto. Com base nesta arquitetura realizou-se a implementação do sistema SES.

6.1 Arquitetura

Conforme é demonstrado na FIGURA 6.1, o sistema proposto foi dividido em módulos, os quais refletem muito aproximadamente a estrutura trivial de um sistema baseado em conhecimento.

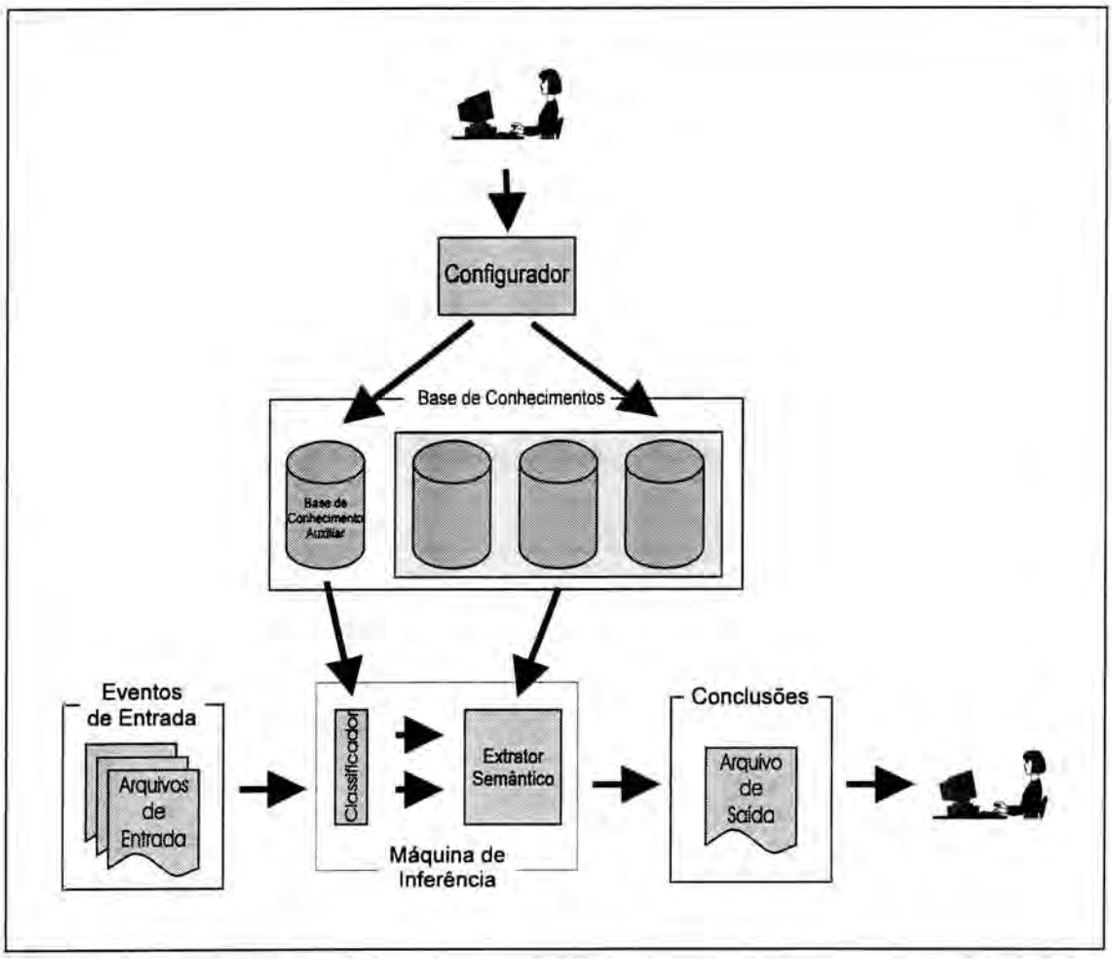

FIGURA 6.1 - Arquitetura do Sistema de Extração Semântica

A extração semântica de informações a partir de bases de dados não estruturadas e semi-estruturadas é uma tarefa de difícil e complexa solução. A modularização permitirá o tratamento do sistema como um conjunto de tarefas básicas independentes, as quais irão espelhar, posteriormente, a geração do código fonte do sistema. Desta maneira, esta

característica de modularização, além de facilitar o entendimento da arquitetura proposta, também facilitará a implementação do sistema como um todo, bem como sua manutenção e documentação.

6.1.1 Arquivos de Entrada

Os Arquivos de Entrada representam a base de dados a ser analisada e processada pelo Extrator Semântico, sendo composta por vários arquivos de dados com informações armazenadas de forma textual não estruturada ou semi-estruturada. Cabe salientar que estes arquivos são selecionados pelo próprio usuário do sistema.

Relacionando os Arquivos de Entrada com a estrutura de um Sistema Baseado em Conhecimento, observa-se que estes representam os eventos de entrada.

6.1.2 Arquivo de Saída

O Arquivo de Saída contém o resultado da análise e processamento dos Arquivos de Entrada pelo Extrator Semântico. Os dados armazenados neste arquivo estão estruturados e em formato texto, facilitando sua importação para outros sistemas que possam se valer das informações extraídas.

É importante ressaltar que o Arquivo de Saída dentro da estrutura de um Sistema Baseado em Conhecimento representa a conclusão gerada pelo processamento realizado pela máquina de inferência.

6.1.3 Extrator Semântico

Conforme a FIGURA 6.1, o módulo de Extração Semântica recebe os Arquivos de Entrada, processa-os e gera o Arquivo de Saída. Este módulo representa o núcleo do sistema proposto, sendo responsável pela extração dos dados armazenados em cada um dos arquivos que compõem a base de dados de entrada.

A extração é realizada conforme as regras armazenadas na Base de Conhecimento e configuradas pelo Usuário do sistema. Desta forma, o Extrator Semântico é um módulo do sistema cuja a forma de execução de sua função é configurável, conforme as necessidades do usuário e as características dos Arquivos de Entrada.

Juntamente com o Classificador, o Extrator Semântico forma a máquina de inferência do sistema proposto. Esta divisão foi realizada a fim de clarear o entendimento da arquitetura aqui apresentada com base nos tópicos conceituais expostos nos capítulos anteriores.

6.1.4 Classificador

Todos os arquivos de entrada, antes de serem processados pelo Extrator Semântico, passam por um prévio processamento realizado pelo Classificador. Este

processamento verifica a extensão do arquivo de entrada, identificando o tipo deste e selecionando o conjunto de regras de extração a ser aplicado a este arquivo.

Desta forma, fica clara a ligação existente entre este módulo e o Extrator Semântico, para que estes dois juntos formem a máquina de inferência.

6.1.5 Base de Conhecimento

A Base de Conhecimento é formada por um conjunto de arquivos nos quais são armazenadas as regras de extração. Tais regras são configuradas pelo Usuário e definem como o módulo de Extração Semântica irá recuperar as informações existentes nos Arquivos de Entrada.

Cada um dos arquivos da Base de Conhecimento se aplica a um determinado tipo de arquivo de entrada, como, por exemplo, arquivos de correspondência eletrônica (*Mails*). Além disso, os diversos tipos de interesse do usuário (informações a serem extraídas), também podem ser divididos em arquivos de conhecimento distintos, sendo estes utilizados conforme a necessidade do usuário. Neste último caso, por exemplo, pode-se ter um arquivo de conhecimento para *mails* sobre viagens turísticas, e outro para *mails* sobre congressos.

A seleção de qual arquivo da Base de Conhecimento utilizar é realizada pelo classificador.

6.1.6 Base de Conhecimento Auxiliar

A Base de Conhecimento Auxiliar é formada apenas por um arquivo, no qual é armazenado o relacionamento entre os arquivos de regras que também compõem a Base de Conhecimento e os Arquivos de Entrada. Este relacionamento é usado pelo Classificador na seleção do arquivo de regras propício ao processo de extração, conforme a extensão do arquivo de entrada.

Juntamente com os arquivos de regras, a Base de Conhecimento Auxiliar forma a Base de Conhecimento do sistema proposto. Contudo, conforme a divisão realizada na máquina de inferência entre o Classificador e o Extrator Semântico, o conhecimento utilizado por estes também foi dividido para fins de clareza da arquitetura proposta.

Cabe lembrar que o próprio usuário, através do Configurador, fornece os dados armazenados na Base de Conhecimento.

6.1.7 Configurador

O sistema descrito apresenta uma Base de Conhecimento com o conjunto de regras utilizadas para a extração das informações pelo Extrator Semântico. Contudo, a geração da Base de Conhecimento é realizada por um módulo auxiliar, externo à estrutura convencional de um sistema baseado em conhecimento: o módulo Configurador. Este módulo viabiliza a operação de configuração da Base de Conhecimento, provendo uma

interface de acesso a tal conhecimento, bem como um *parser* de entrada para as regras usadas na especificação deste.

6.1.8 Usuário

O Usuário é caracterizado pela pessoa que configura o sistema ou usufrui dos resultados do mesmo, conforme demonstra a FIGURA 6.1. O uso de Bases de Conhecimento, no entanto, permite que o sistema proposto possa ser manipulado por dois tipos de usuário, correspondendo a papéis distintos, os quais podem ser assumidos pela mesma pessoa. Esta qualidade, possibilita uma maior flexibilidade de uso quanto às características exigidas do usuário pelo sistema.

O primeiro tipo de Usuário é chamado de Configurador do Sistema, pois é caracterizado pela pessoa responsável pela geração da Base de Conhecimento. Este tipo de Usuário utiliza o módulo Configurador, o que exige o entendimento da forma de representação do conhecimento usada no sistema, bem como um raciocínio formal com relação à estruturação deste.

No entanto, cabe salientar que diferentes usuários podem trocar entre si Bases de Conhecimento para diversos tipos de arquivos de entrada, facilitando o processo de configuração, já que tais bases são independentes do resto do sistema.

O segundo tipo de usuário é chamado de Usuário Final, pois é caracterizado pela pessoa que somente ativa o Extrator Semântico, indicando quais arquivos devem ser analisados. Como resultado, ele obterá o Arquivo de Saída. É importante destacar que este tipo de Usuário não terá necessidade de gerar a Base de Conhecimento, utilizando bases já existentes, criadas anteriormente por Usuários Configuradores.

Desta forma, o Usuário informará ao sistema como detectar e extrair os dados relevantes referentes ao assunto sob pesquisa conforme um conjunto de regras pré-determinadas, as quais serão estruturadas e combinadas a fim de atingir o objetivo desejado.

6.2 Visão Geral

O processo de extração semântica projetado e descrito acima pode ser dividido em quatro fases básicas:

- **Primeira Fase**

Nesta fase é realizada a definição das Bases de Conhecimento necessárias para o módulo de extração. Tais bases permitem a configuração do Extrator Semântico, viabilizando sua adaptação aos diversos tipos de Arquivo de Entrada. Para tanto, o usuário faz uso do Configurador que provê uma interface onde é realizada a definição do conjunto de regras que formam as Bases de Conhecimento.

- **Segunda Fase**

Neste momento, é feita a configuração da Base de Conhecimento Auxiliar. O usuário utiliza-se do módulo Configurador para relacionar os tipos de Arquivo de Entrada com os arquivos que formam a Base de Conhecimento.

Deve-se observar que as duas primeiras fases não são realizadas freqüentemente, uma vez que as regras e os relacionamentos podem ser reaproveitados para diversos Arquivos de Entrada.

- **Terceira Fase**

Neste terceiro passo, o Extrator Semântico e o Classificador são ativados pelo usuário, ocorrendo a entrada dos arquivos a serem analisados. Desta forma, as informações de configuração (regras e dados sobre as extensões de arquivos) das fases anteriores são utilizadas.

Primeiramente, o Classificador deve identificar o tipo do Arquivo de Entrada, pesquisando na tabela de relacionamentos armazenada na Base de Conhecimento Auxiliar. Isto permiti a seleção das configurações a serem utilizadas na extração. Após, é passado ao Extrator Semântico o Arquivo de Entrada e o nome do arquivo de regras a ser utilizado, para que então seja realizada a extração das informações. Desta maneira, a extração somente é possível com o uso das regras anteriormente definidas pelo usuário e que se encontram nas Bases de Conhecimento. Com isto, por exemplo, arquivos com extensão TXT sofrem um processo de extração diferenciado de arquivos com extensão HTM.

- **Quarta Fase**

A última fase é representada pela obtenção dos dados de saída pelo Usuário Final, ou seja, a leitura do Arquivo de Saída. Esta leitura pode ser diretamente realizada sobre o Arquivo de Saída ou através de um outro sistema que utilize-se do arquivo gerado.

Neste sentido, como pode-se observar, a terceira e quarta fases serão executadas sempre que se alterarem os Arquivos de Entrada.

6.3 Implementação do Protótipo

A implementação do protótipo foi baseada totalmente na arquitetura acima apresentada e, conseqüentemente, na estrutura de um Sistema Baseado em Conhecimento, como pode ser observado na FIGURA 6.2.

Desde o início do desenvolvimento do protótipo ora em estudo, buscou-se projetar não um sistema monolítico, mas sim um ambiente orientado a expansão; ou seja, um conjunto integrado de ferramentas ao qual é possível, sem grande esforço de programação, agregar outros módulos ou então adicionar novas funcionalidades aos já existentes.

A construção de um protótipo composto por várias funções independentes e encapsuladas, além de responder satisfatoriamente ao objetivo acima citado, também

apresenta uma série de vantagens no que se concerne a seu desenvolvimento, implementação e uso. Tais características podem ser avaliadas na própria estrutura do código fonte produzido, devendo-se destacar algumas vantagens importantes:

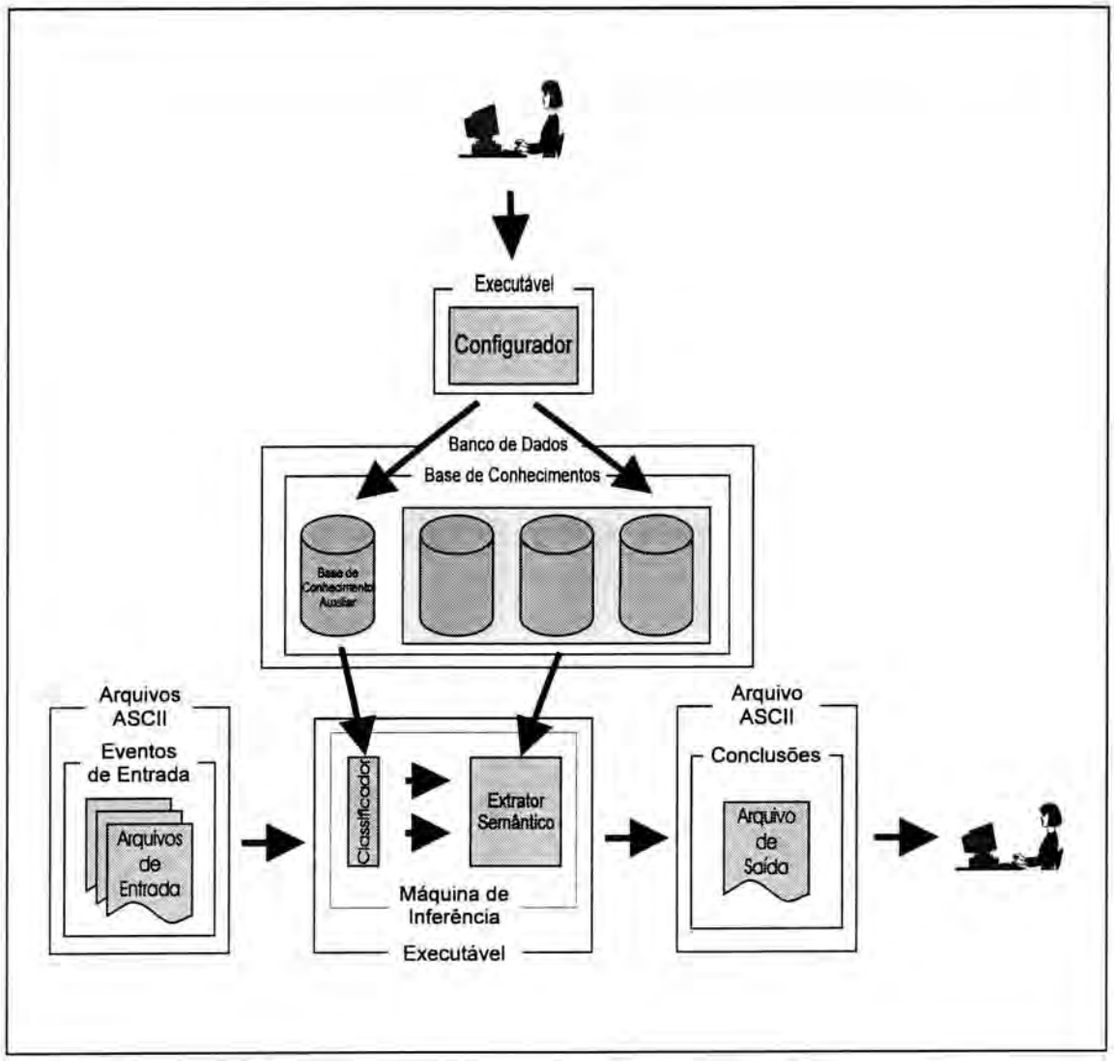

FIGURA 6.2 - Protótipo do Sistema de Extração Semântica

- Possibilidade de projetar, implementar e testar individualmente cada um dos módulos do sistema, o que, sem dúvida, diminui tanto o tempo quanto a complexidade envolvida na integração final dos módulos/funções constituintes.

- Pouca incidência de erros quando da integração em virtude dos testes preliminares realizados em cada uma das funções e módulos.

- Mesmo após a integração, seguindo as diretrizes de sistemas baseados em conhecimento, onde a base de conhecimento é independente da máquina de inferência, decidiu-se manter dois executáveis independentes (máquina de inferência e Configurador). Esta característica facilitou a existência de dois tipos de usuário com papéis distintos: Configurador e Usuário Final.

- O processo de manutenção e evolução é bastante facilitado pela estrutura do programa gerado, permitindo a integração de novas características ao processo de extração semântica, como, por exemplo, a adição de novas funções de extração as já implementadas e apresentadas no capítulo 4.

6.3.1 Módulos do Sistema

O protótipo do sistema está dividido em um conjunto de três módulos principais, que após compilados geram dois executáveis, como mostra a FIGURA 6.2. Contudo, cada um destes módulos pode estar dividido em um ou mais sub-módulos menores, visando uma melhor organização e clareza do código fonte gerado. Além disso, cada um destes sub-módulos pode conter uma ou mais funções que juntas executam uma tarefa independente encapsulada no sub-módulo.

Desta forma, existe uma hierarquia organizacional que foi usada para a geração do código fonte e executável. A FIGURA 6.3 mostra cada um dos níveis hierárquicos, desde o mais globalizador (executáveis) até o mais básico (funções).

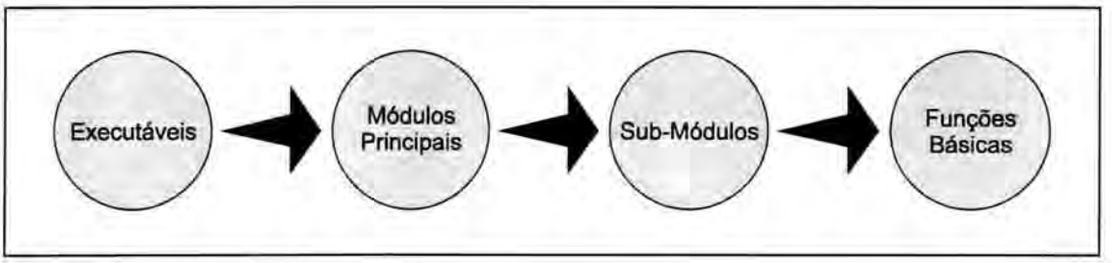

FIGURA 6.3 - Hierarquia Organizacional do SES

6.3.1.1 Módulo de Extração

A seguir serão descritos os vários sub-módulos existentes no módulo de extração do sistema SES:

- **Leitor de Arquivos**

O Leitor de Arquivos é um sub-módulo do sistema de extração. O usuário, através do Configurador, seleciona quais tipos de arquivo devem ser lidos pelo leitor quando identifica na base de conhecimento auxiliar as extensões de arquivos a serem analisados. Sempre que requisitado, o Leitor de Arquivos lê um novo arquivo de entrada para o Extrator.

A fim de melhorar a performance do Sistema de Extração Semântica, o arquivo de entrada, quando requisitado pelo Extrator, é lido inteiramente do disco (memória secundária) para a memória principal, até um limite de 32 Kb. Arquivos maiores serão

truncados. Isto permite um processo de pesquisa mais rápido das informações contidas no arquivo de entrada.

- **Gerador de *Tokens***

O arquivo de entrada colocado em memória principal pelo Leitor de Arquivos é lido palavra-a-palavra pelo Gerador de *Tokens*, sendo que estas palavras são na realidade os *tokens* de entrada para o Extrator.

O Gerador de *Tokens* é um módulo de uma só função que é chamada pelos sub-módulos Extrator, Verificador de Condições e Executor de Ações sempre que estes necessitam de um novo *token*.

- **Analisador de Regras**

O sub-módulo Analisador de Regras recebe um *token* do Extrator e verifica quais regras da base de conhecimento apresentam este *token* em suas condições de pesquisa. Caso esta verificação seja verdadeira, a regra é retornada ao Extrator.

- **Extrator**

O Extrator é o núcleo do Sistema de Extração Semântica, realizando a coordenação de todos os sub-módulos deste sistema. Apesar de ser um sub-módulo pequeno e simples, este executa um conjunto grande de chamadas de outros sub-módulos, coordenando o processo de extração como um todo:

  - Requisita a leitura de um novo arquivo de entrada para memória principal ao Leitor de Arquivos.
  - Verifica através do módulo Classificador qual base de conhecimento utilizar conforme a extensão do arquivo de entrada.
  - Ativa o Gerador de *Tokens*, enviando o *token* gerado ao Analisador de Regras.
  - Recebe as regras cuja as condições de pesquisa apresentam o *token* gerador como condição, ativando as funções do sub-modulo Verificador de Condições de pesquisa e verificação, respectivamente.
  - Ativa as funções do sub-módulo Executor de Ações para as regras cujas as condições de pesquisa e verificação forem verdadeiras.

- **Verificador de Condições**

Conforme a descrição realizada no item 4.3.2, o sub-módulo Verificador de Condições contém as funções que irão interpretar as condições de pesquisa e verificação das regras informadas pelo Extrator, averiguando se estas condições são verdadeiras ou falsas. Muitas vezes as funções de verificação de condição realizam chamadas ao Gerador de *Tokens*, requisitando *tokens* anteriores ou posteriores ao correntemente extraído necessários à verificação de um condição.

- **Executor de Ações**

O sub-módulo Executor de Ações comporta as funções que realizam a extração das informações contidas nos arquivos de entrada conforme as regras informadas pelo Extrator. Cabe salientar que existe uma função para cada uma das ações anteriormente descritas no item 4.3.3.

Dois outros sub-módulos são chamados a partir do sub-módulo de Executor de Ações: o Gerador de *Tokens*, para a extração de novos *tokens* (termos) do arquivo de entrada, e o Gerador do Arquivo de Saída, que acrescenta os dados extraídos ao arquivo de saída.

- **Gerador do Arquivo de Saída**

O Gerador do Arquivo de Saída é um sub-módulo que recebe em sua chamada o termo extraído do arquivo de entrada e o grupo de assunto informado pelo usuário na regra de extração. Desta forma, este sub-módulo simplesmente escreve no arquivo de saída, incluído de forma estruturada, conforme o item 4.4, o termo extraído.

6.3.1.2 Módulo de Classificação

O módulo de classificação forma junto com o módulo de extração o executável principal do SES, sendo o primeiro formado apenas por um sub-módulo: Classificador.

- **Classificador**

O Classificador recebe o nome do arquivo lido pelo Leitor de Arquivos do Extrator. Deste nome é retirada a extensão do arquivo de entrada, sendo esta pesquisada dentro da base de conhecimento auxiliar. O resultado da pesquisa é o nome da base de conhecimento que contém as regras de extração a serem utilizadas com o arquivo lido.

6.3.1.3 Módulo de Configuração

O módulo de configuração é composto por três sub-módulos menores e, após compilado, resulta em um executável independente do executável formado pelos módulos de classificação e extração (FIGURA 6.2).

- **Gerador de Base de Conhecimento**

O Gerador de Base de Conhecimento é o sub-módulo que apresenta a interface para a geração e manutenção das bases de conhecimento envolvidas no processo de extração do sistema SES. Esta interface facilita a manipulação e a geração das regras que compõem cada base de conhecimento, provendo também um *parser* de entrada que consiste as regras informadas pelo usuário configurador.

As regras são armazenadas em um banco de dados que representa a base de conhecimento (item 6.3.2).

- **Gerador de Base de Conhecimento Auxiliar**

O Gerador de Base de Conhecimento Auxiliar executa o mesmo papel do gerador anterior, porém sobre bases de conhecimento auxiliares. Igualmente ao item acima, os dados informados pelo usuários configurador também são armazenados em um banco de dados.

- **Gerador de Dicionário de Termos**

O Gerador de Dicionário de Termos é uma interface para a formação de arquivos de dados contendo termos relacionados entre si. Através deste sub-módulo, o usuário configurador pode criar e atualizar vários destes dicionários, que, posteriormente, serão usados no processo de extração.

6.3.2 Bases de Conhecimento

As Bases de Conhecimento armazenam as regras de configuração do extrator de informações. Devido à estrutura simples de tais regras, estas foram facilmente estruturadas e guardadas em uma base de dados relacional (FIGURA 6.4).

Contudo, apesar de logicamente a base de conhecimento estar dividida em um conjunto de 6 arquivos, fisicamente esta é armazenada em um único arquivo. Neste sentido, os arquivos que formam a base podem ser facilmente trocados entre os usuários do SES.

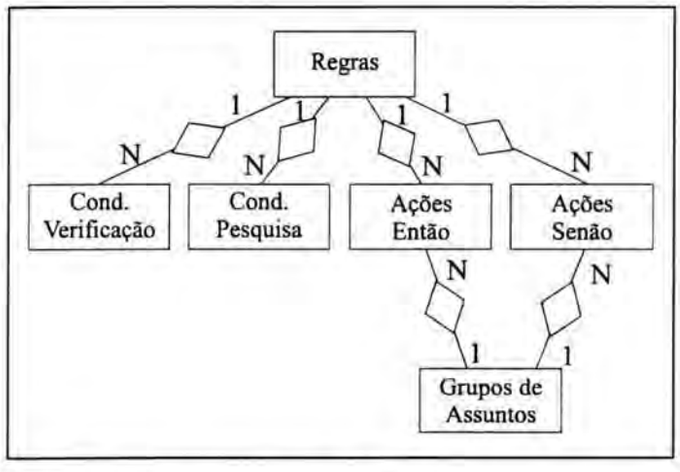

FIGURA 6.4 - Esquema da Base de Dados do SES

6.3.3 Ferramenta de Programação

O desenvolvimento do protótipo do sistema SES proposto e descrito nesta dissertação foi realizado sobre a ferramenta de programação Clarion 1.5 *for Windows* [TOP 95] [TOP 95a] [TOP 95b]. O uso desta ferramenta facilitou o processo de programação e aumentou a qualidade do protótipo gerado.

O Clarion é um gerenciador de banco de dados com uma linguagem de programação nativa. Contudo, a ferramenta como um todo é composta por vários módulos interrelacionados: Editor de Dicionário de Dados, Gerador de Programas, Compilador e *Debbuger* .

O gerador de programas automático juntamente com o editor de dicionário de dados, permite que o usuário da ferramenta descreva o dicionário de dados da aplicação sob desenvolvimento, montando a estrutura do banco de dados a ser utilizado [HEL 96]. A partir desta descrição o gerador de código gera o esqueleto da aplicação sob criação. O uso desta característica atribuiu ao protótipo uma maior velocidade de desenvolvimento, bem como maior facilidade de manutenção e realização de alterações.

Gerando programas para o ambiente Windows 95, o Clarion permitiu que o protótipo final se beneficiasse ao máximo das características deste sistema operacional, principalmente quanto à interface. O uso deste ambiente gráfico já difundido mundialmente, irá facilitar a manipulação do sistema por usuários sem conhecimento profundos de informática.

Por outro lado, o uso da linguagem de programação nativa do Clarion resultou em um sistema com boa performance de execução, pois o código executável gerado é tão rápido quanto o código executável gerado a partir de um compilador C [COF 95]. Esta qualidade de ser uma linguagem de alto nível com boa performance de execução foi a característica mais importante na escolha deste compilador, a fim de atingir-se os objetivos deste trabalho. Isto porque, como apresentado no capítulo 2, a velocidade de extração é um ponto relevante dentro das características desejáveis de um sistema deste tipo.

7 Usando o SES

O aproveitamento de todas as potencialidades e facilidades oferecidas por uma aplicação, depende fundamentalmente do nível de conhecimento que o usuário possui a respeito do sistema, bem como da experiência prática por ele adquirida com a contínua e freqüente utilização do mesmo.

Uma vez que o público alvo do SES é constituído, em grande parte, por pessoas não especializadas na área de informática, devemos apresentar de forma clara o funcionamento do sistema, sem deter-nos excessivamente em particularidades. Assim, neste capítulo, faremos uma descrição detalhada das facilidades, potencialidades e maneiras de usar adequadamente o sistema SES.

Além de satisfazer aos usuários iniciantes, que buscam neste texto uma orientação precisa no que concerne ao uso do sistema, também escrevemos este capítulo de forma que o leitor especializado possa analisar com clareza a interface do SES, discutida anteriormente no capitulo 5, e a filosofia de trabalho e funcionamento do mesmo.

7.1 Instalação

O procedimento necessário a instalação do SES é bastante simples, possibilitando que o próprio usuário o execute. A seguir são apresentados os passos para a instalação do sistema sobre o ambiente Windows 95:

1. Coloque o disquete do SES no *drive*.

2. Pressione o botão iniciar e selecione a opção executar.

3. Digite no campo abrir o seguinte comando: A:\Instala A: C:. Se o disquete estiver no *drive* B, digite B:\Instala B: C:.

O programa de instalação irá copiar os arquivos do SES para o seu disco rígido, criando o diretório \SES. A partir deste momento, o usuário necessita somente executar um dos dois executáveis que compõem o SES: Extrator e Configurador.

7.2 Executando o Configurador

O Configurador é um dos executáveis que compõem o sistema SES. Seu objetivo é realizar a geração e configuração das bases de conhecimento utilizadas pelo sistema e, conseqüentemente, definir todas as características do processo de extração.

Para inicializar o Configurador, o usuário deverá dar um *double-click* sobre o ícone que representa este executável (FIGURA 7.1). Realizada esta operação o usuário se encontrará dentro do ambiente de trabalho do módulo Configurador (FIGURA 7.2).

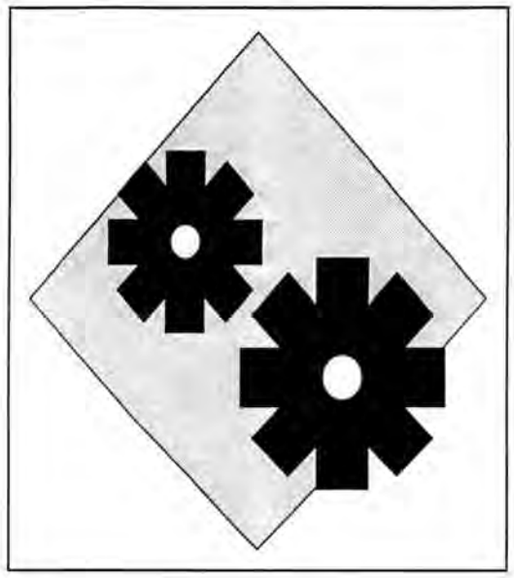

FIGURA 7.1 - Ícone do Configurador

7.2.1 Ambiente do Configurador

O usuário do Sistema de Extração Semântica de Informações ao entrar no módulo Configurador irá posicionar-se dentro do ambiente de trabalho deste módulo (FIGURA 7.2). É neste ambiente que é realizada a configuração do processo de extração.

Cabe salientar que, sendo um programa desenvolvido para o ambiente Windows 95, o usuário tem a possibilidade de realizar algumas alterações sobre o ambiente de trabalho do SES. Neste sentido, sempre que alteradas as características do Windows quanto ao ambiente, o SES irá adaptar-se as alterações definidas pelo usuário.

O ambiente de trabalho apresentado após a execução do Configurador pode ser dividido em cinco partes:

- Barra de Título: informa ao usuário o nome do sistema de extração (SES), apresentando seu ícone (FIGURA 7.1). Além disso, permite que este usuário minimize ou maximize a janela da aplicação, bem como finalize a mesma (botões no canto superior direito definidos no ambiente Windows 95).

- Menu Principal: contém o conjunto de operações possíveis de serem utilizadas na configuração das bases de conhecimento.

- Barra de Botões: abaixo do menu principal, a barra de botões contém um conjunto de botões para acesso rápido as operações do Configurador.

- Área de Trabalho: local onde o usuário irá gerar e configurar as bases de conhecimento do SES.

- Barra de *Status*: posicionada na parte inferior da tela, a barra de *status* informa ao usuário o andamento de cada uma de suas ações dentro do sistema, bem como sua localização entre as diversas tabelas que fazem parte do módulo de configuração.

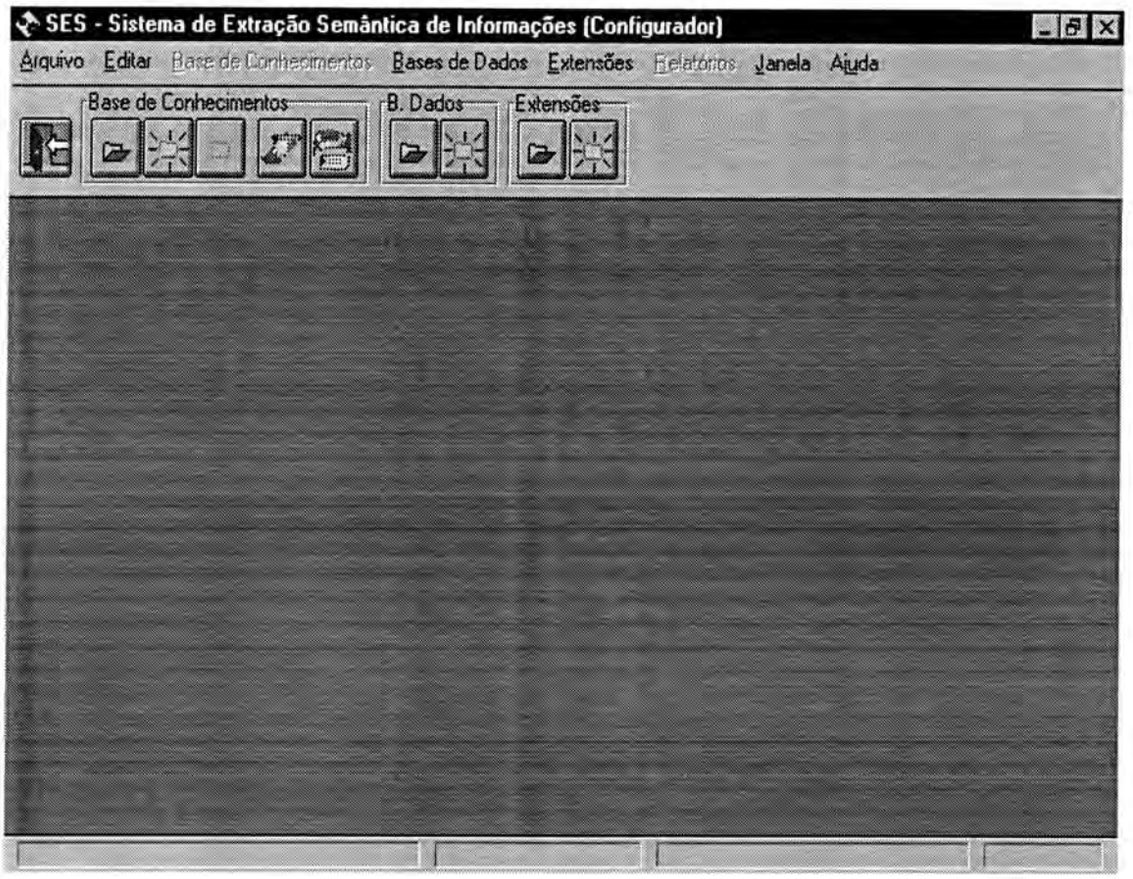

FIGURA 7.2 - Ambiente de Trabalho do Configurador

7.2.2 Abrindo uma Base de Conhecimento

Para a configuração de uma base de conhecimento, o usuário deve indicar ao sistema com qual base irá trabalhar. O acesso às bases de conhecimento é realizado através do item de menu **Arquivo/Abrir** ou do **segundo botão** da barra de botões. Neste caso, o usuário está abrindo uma base de conhecimento já existente para alteração e/ou visualização da mesma.

Selecionada a opção Abrir do menu ou pressionado o segundo botão, será aberta uma caixa de diálogo que permite a navegação entre os discos e diretórios existentes no computador do usuário, para que este possa localizar e selecionar a base de conhecimentos que deseja manipular (FIGURA 7.3). É importante lembrar que os arquivos físicos que contêm as bases de conhecimento utilizam, normalmente, a extensão **.CON**.

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

Após selecionada a base de conhecimento e pressionado o botão OK da caixa de diálogo, as opções do menu relacionadas com a configuração de bases de conhecimento serão habilitadas, passando da cor cinza clara para preto e permitindo a edição da base selecionada e aberta.

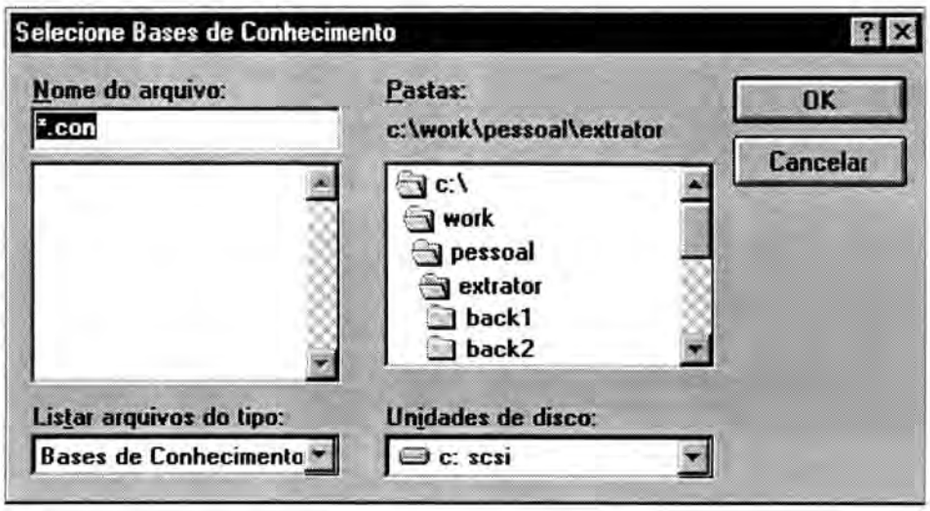

FIGURA 7.3 - Abrindo uma Base de Conhecimento

7.2.3 Criando uma Nova Base de Conhecimento

Quando o usuário do sistema SES deseja criar uma nova base de conhecimento sem que esta contenha previamente alguma regra de extração, ele deverá utilizar o menu **Arquivo/Novo** ou o **terceiro botão** da barra de botões. Neste caso, o usuário está gerando uma nova base de conhecimento vazia (sem regras).

Selecionada a opção Novo do menu ou pressionado o terceiro botão, será aberta uma caixa de diálogo (FIGURA 7.4) que permite a navegação entre os discos e diretórios existentes no computador do usuário, para que este possa localizar onde a nova base de conhecimento será gravada. Além da localização desejada, o usuário deve informar o nome da nova base a ser criada. Deve-se observar que os arquivos físicos que contém as bases de conhecimento utilizam, normalmente, a extensão **.CON**.

Após a criação da nova base de conhecimento, esta é automaticamente aberta para configuração, e as opções do menu relacionadas com a configuração de bases de conhecimento serão habilitadas, passando da cor cinza clara para preto e permitindo a edição da nova base.

Por outro lado, é importante lembrar que o usuário do SES tem a possibilidade de gerar uma nova base de conhecimento a partir de uma base já existente. Neste caso, deve-se abrir a base de conhecimento original, conforme o procedimento explicado em 7.2.2, e selecionar o menu **Arquivo/Salvar Como**. Esta operação irá criar uma nova base de conhecimento a partir da originalmente aberta, copiando todo seu conteúdo de regras.

Sempre que a opção **Arquivo/Salvar Como** for utilizada, será aberta uma caixa de diálogo onde o usuário irá informar o diretório e o nome da nova base a ser criada(FIGURA 7.4).

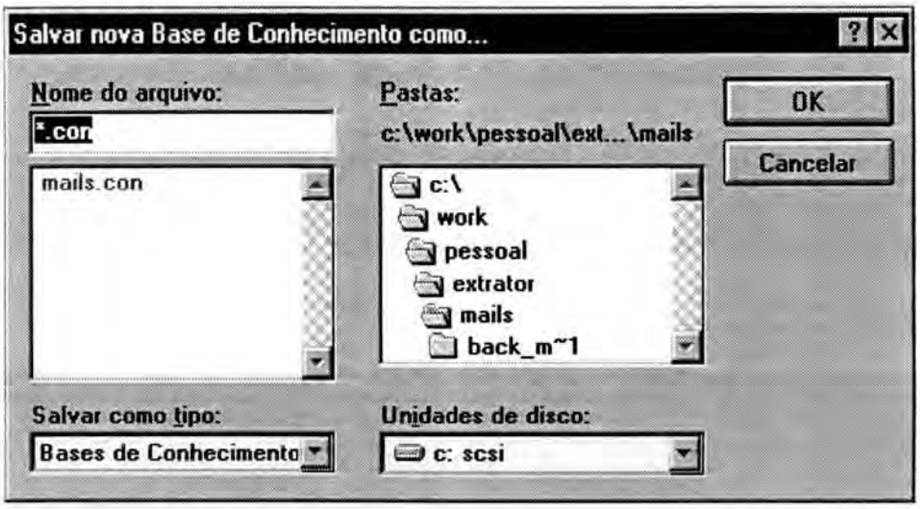

FIGURA 7.4 - Salvando uma Nova Base de Conhecimento

7.2.4 Grupos de Assuntos

Como primeiro passo para a configuração de uma base de conhecimento, o usuário deverá definir os grupos de assuntos que deseja extrair a partir dos arquivos de entrada.

Associado a cada função de extração de uma base de conhecimento existe a informação de um grupo de assunto que deve ser definido previamente pelo usuário. Estes grupos de assuntos têm por objetivo classificar e organizar a informação extraída, melhorando a qualidade resultante no arquivo de saída.

Para configurar os grupos, o usuário deve selecionar o item de menu **Base de Conhecimento/Grupos** ou pressionar o **sexto botão** da barra de botões. Sendo assim, será apresentada na área de trabalho a tabela de grupos da base de conhecimento correntemente aberta (FIGURA 7.5)

Para o manuseio desta tabela deve-se utilizar os botões existentes na mesma:

- **Botão Incluir** ou **Tecla *Insert***: Após pressionado o botão para a inclusão, será aberta a ficha de grupo, onde deve ser preenchido o título do novo grupo e pressionado o botão OK. Desta forma, será confirmada a inclusão do novo grupo de assunto, que passará a fazer parte da tabela de grupos.

- **Botão Alterar** ou **Tecla Enter**: Altera o grupo destacado pela barra de seleção da tabela de grupos. Neste sentido, quando pressionado o botão alterar, a ficha de grupo será aberta, apresentando o título do grupo a ser

alterado. Após a alteração, o usuário deve pressionar OK para confirmar a mesma, ou CANCELA desistindo do processo de alteração.

- **Botão Excluir** ou **Tecla *Delete***: Exclui o grupo destacado pela barra de seleção da tabela. Contudo, antes da exclusão efetiva, será perguntado ao usuário se este realmente deseja prosseguir com o processo de exclusão, devendo ser respondido, através dos botões, SIM ou NÃO.

- **Botão Fechar**: Fecha a tabela de grupos.

- **Botão Ajuda**: Ativa o texto de auxílio ao usuário referente à tabela de grupos.

FIGURA 7.5 - Tabela de Grupos

7.2.5 Regras de Extração

O principal processo de configuração realizado pelo Configurador é o das bases de conhecimento, ou seja, das regras contidas nestas bases. As regras de extração realizam a configuração do processo de extração. É através delas que o usuário do SES informa como deve-se extrair as informações que ele deseja a partir dos arquivos de entrada.

Para configurar as regras, o usuário deve selecionar o menu **Base de Conhecimento/Regras** ou pressionar o **quinto botão** da barra de botões. Contudo, é imprescindível que, anteriormente, tenha sido aberta a base de conhecimento a ser configurada. Sendo assim, será apresentada na área de trabalho a tabela de regras da base de conhecimento correntemente aberta (FIGURA 7.6)

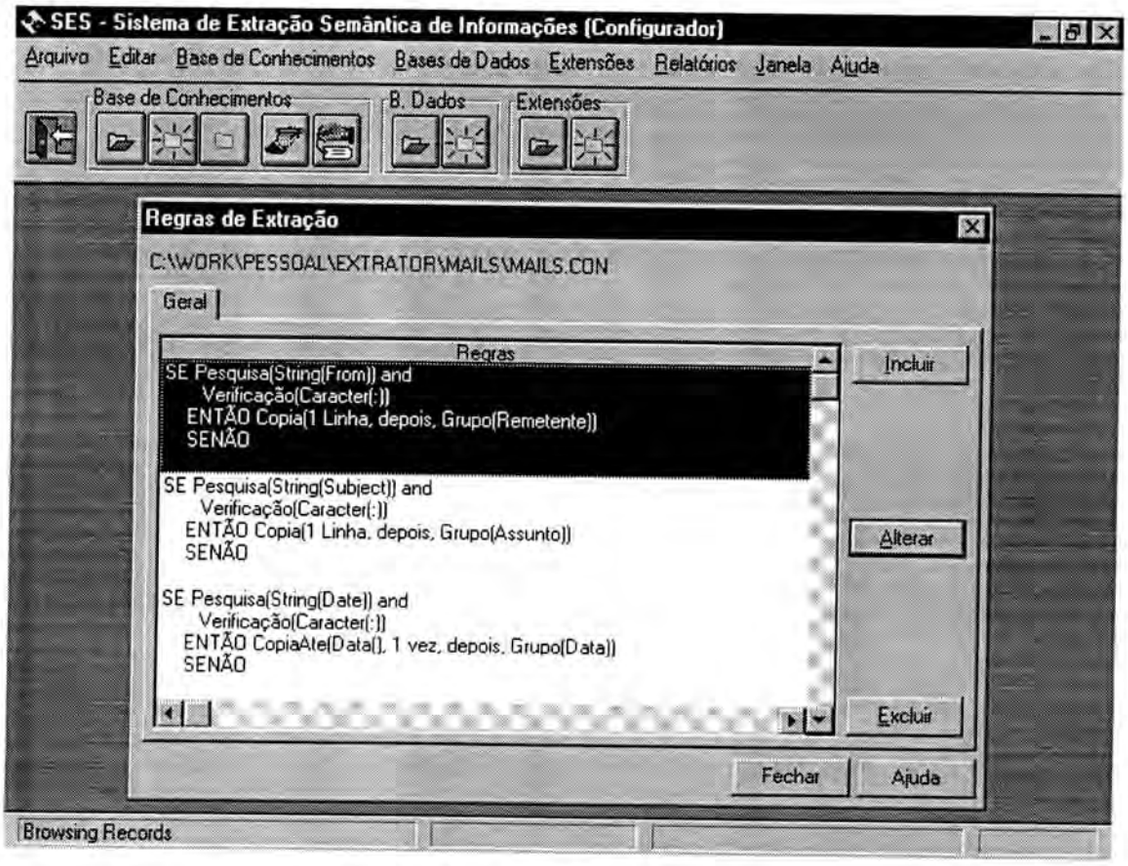

FIGURA 7.6 - Tabela de Regras

Para o manuseei-o desta tabela deve-se utilizar os botões existentes na mesma:

- **Botão Incluir** ou **Tecla *Insert***: Após pressionado o botão para a inclusão, será aberta a ficha de regra (FIGURA 7.7), onde devem ser preenchidos os campos referentes à regra a ser incluída e pressionado o botão OK. Desta forma, será confirmada a inclusão da nova regra, que passará a fazer parte da tabela de regras.

- **Botão Alterar** ou **Tecla Enter**: Altera a regra destacada pela barra de seleção da tabela de regras. Neste sentido, quando pressionado o botão alterar, a ficha de regra (FIGURA 7.7) será aberta, apresentando os dados referentes à regra a ser alterada. Após a alteração, o usuário deve pressionar OK para confirmar a mesma, ou CANCELA desistindo do processo de alteração.

- **Botão Excluir** ou **Tecla *Delete***: Exclui a regra destacada pela barra de seleção da tabela. Contudo, antes da exclusão efetiva, será perguntado ao

usuário se este realmente deseja prosseguir com o processo de exclusão, devendo ser respondido, através dos botões, SIM ou NÃO.

- **Botão Fechar**: Fecha a tabela de regras.

- **Botão Ajuda**: Ativa o texto de auxílio ao usuário referente à tabela de regras.

7.2.5.1 Ficha de Regra

A ficha de regra apresenta o conteúdo de cada regra de conhecimento dividido em quatro partes a fim de facilitar o entendimento e a visualização das mesmas. Estas partes estão divididas em três "orelhas" destacadas na parte superior da ficha de regra. Com o *mouse*, o usuário pode selecionar qual orelha visualizar.

- Condições de Pesquisa (primeira orelha)
- Condições de Verificação (primeira orelha)
- Ações Então (segunda orelha)
- Ações Senão (terceira orelha)

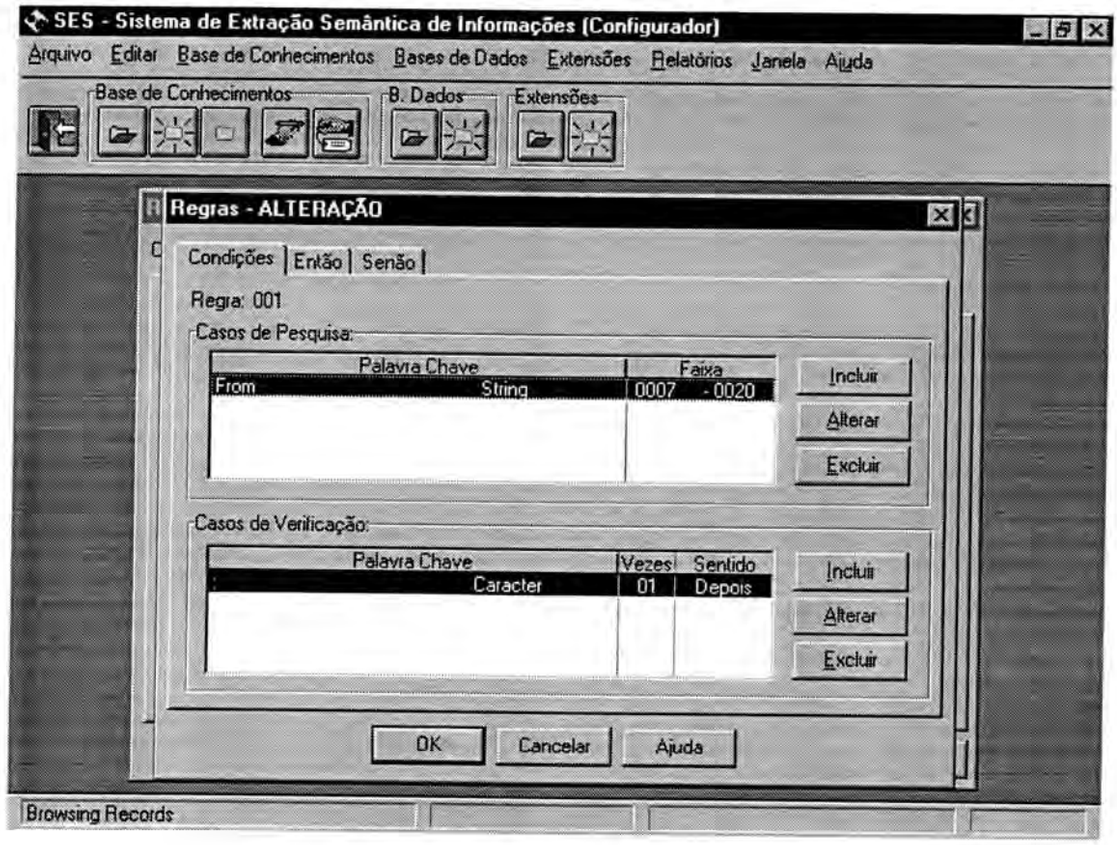

FIGURA 7.7 - Ficha de Regra

Quando a ficha de regra é aberta, ela apresenta as condições que fazem parte da regra selecionada, sendo que o usuário pode incluir, alterar e excluir tais condições,

utilizando os botões à direita da tabela correspondente ao tipo de condição (pesquisa ou verificação) que deseja manipular (FIGURA 7.7).

Para realizar a manipulação das tabelas de ações, o usuário seleciona a orelha correspondente ao tipo de ação desejado (ENTÃO ou SENÃO) e, igualmente, utiliza os botões a direita das tabelas para sua operação (FIGURA 7.8).

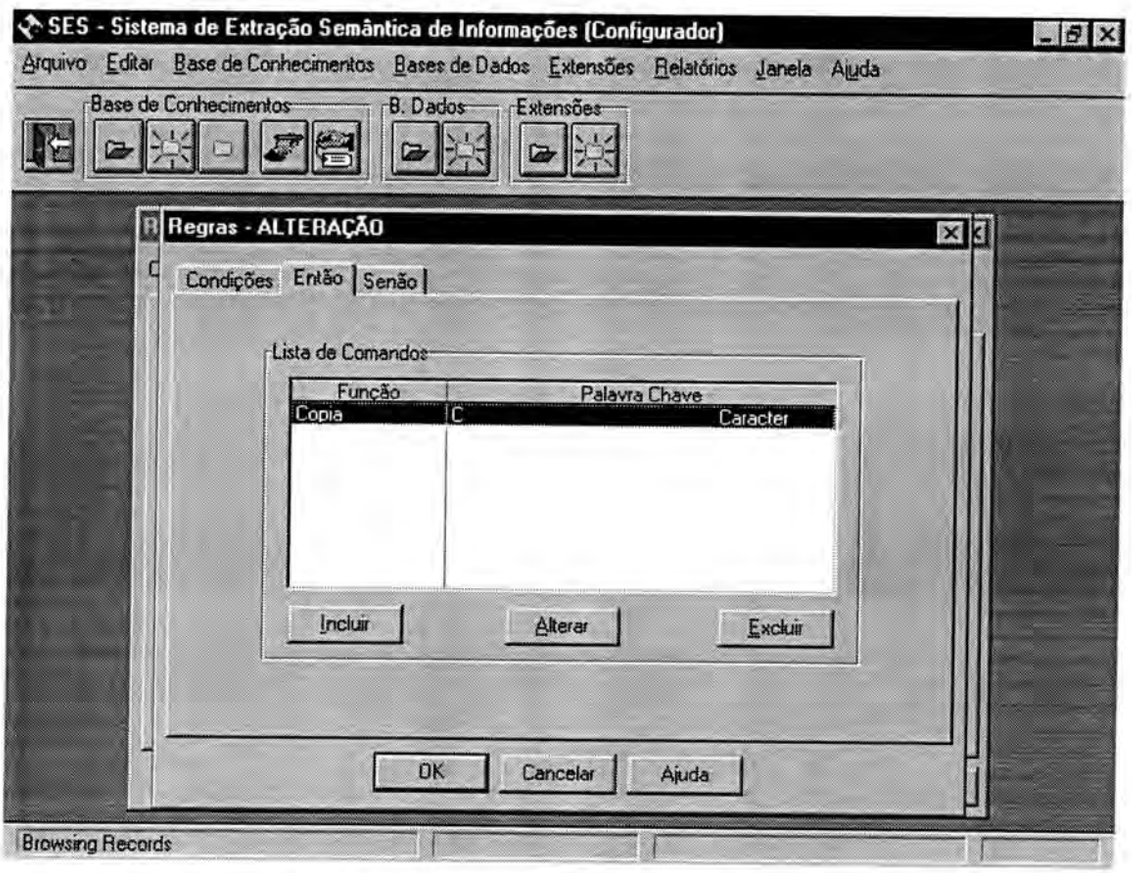

FIGURA 7.8 - Tabela de Ações de Então

7.2.5.2 Condições de Pesquisa

As condições de pesquisa também podem ser chamadas de "cabeças" de pesquisa, pois todo o termo advindo do texto de entrada é comparado com todas as condições de pesquisa existentes nas regras da base de conhecimento, verificando-se as verdadeiras, para que somente estas sejam testadas quanto à condição de verificação. A configuração de uma condição de pesquisa é realizada através de um formulário (FIGURA 7.9), contendo os seguintes campos:

- **Ordem**: Indica a ordem de pesquisa entre as várias condições de pesquisa definidas em uma mesma regra, ou seja, quais serão testadas primeiramente.

- **Tipo**: Identifica o tipo de função de pesquisa a ser utilizado. Dependendo do tipo selecionado pelo usuário, alguns outros campos da ficha de condição de

pesquisa serão desabilitados, passando da cor preta para a cor cinza claro. Os campos desabilitados não poderão ser preenchidos, pois não são válidos para a função escolhida.

A seguir são descritas as funções de condição de pesquisa:

- *String*: Termos declarados diretamente na condição de pesquisa para comparação com os termos de entrada.

- Base de Dados ou Dicionário: Os termos para comparação com o texto de entrada estão declarados em um dicionário de termos (item 0), que pode conter um ou mais termos, sendo que cada termo pode ser composto de uma ou mais palavras. Caso o termo de entrada exista no dicionário de termos a condição é verdadeira.

- Data: Esta função verifica se um termo do texto de entrada é uma data válida, conforme um conjunto de padrões de formatos de datas pré-definidos no extrator semântico (item 7.2.9).

- Livre: Neste caso, a condição de pesquisa sempre é dada como verdadeira independentemente do termo advindo do texto de entrada. Sendo assim, a regra é somente avaliada quanto à condição de verificação.

- **Caixa**: Define se, na comparação entre os caracteres que compõem o termo de pesquisa e os termos contidos no texto de entrada, a caixa dos caracteres é levada em conta para fins de comparação. Quando esta característica não é relevante, seleciona-se a opção **Livre**; ou seja, a caixa não influi no processo de comparação; caso contrário, seleciona-se a opção **Especificada**, indicando que a caixa influi no processo de comparação.

- **Faixas de Pesquisa**: Delimita o escopo de validade da regra de extração em linhas. Caso a faixa não seja informada, assume-se uma validade da condição de pesquisa sobre todo o texto, ou seja, da linha 1 à linha 9999.

Cabe salientar que uma condição de pesquisa pode ser composta, ou seja, podem ser especificadas varias condições para a mesma ação. Desta forma, as diferentes condições são ligadas por um operador do tipo OU. Esta característica faz com que se uma entre todas as condições de pesquisa for verdadeira, a regra estará validada quanto a este grupo de condições, passando ao teste das condições de verificação.

Para utilizar este recurso de várias condições de pesquisa em uma mesma regra, o usuário deve incluir as várias condições através da tabela de condições de pesquisa, usando o **botão Incluir**.

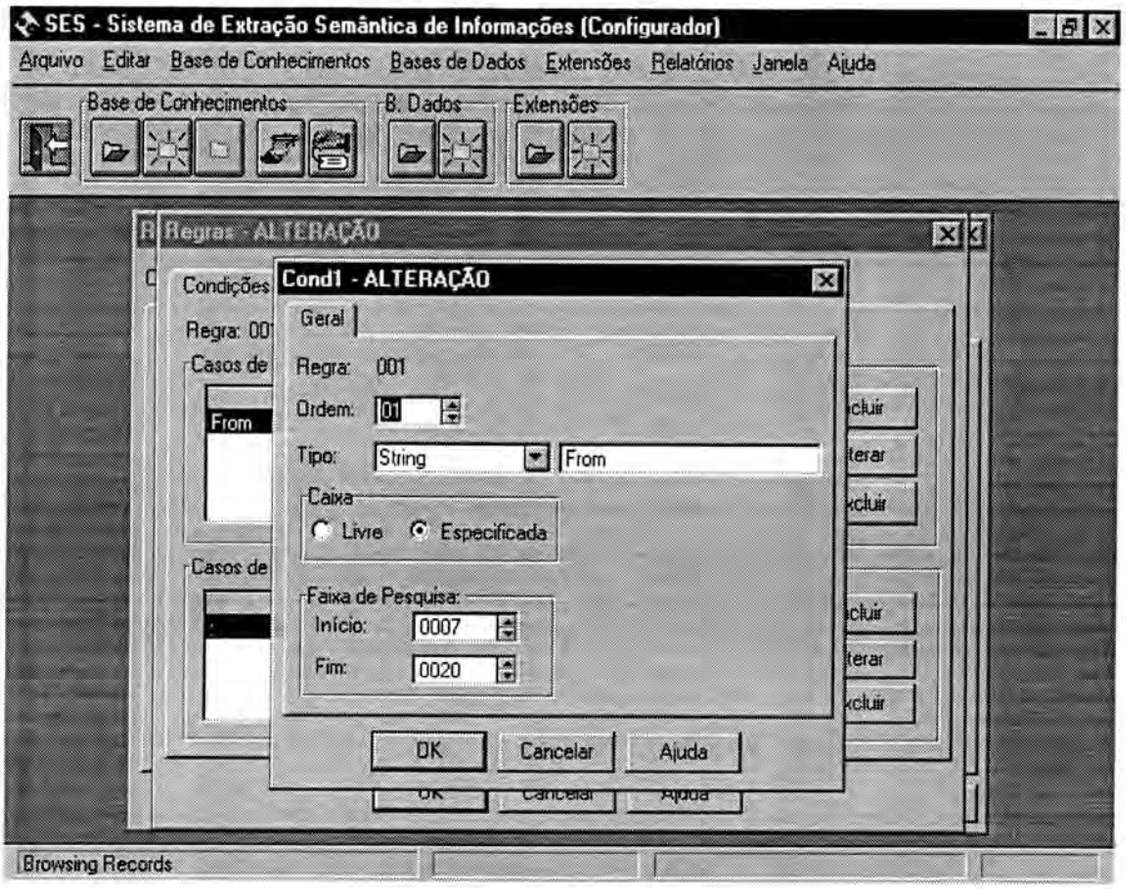

FIGURA 7.9 - Formulário de Condição de Pesquisa

7.2.5.3 Condições de Verificação

As condições de verificação servem como complemento das condições de pesquisa, sendo testadas somente quando as condições de pesquisa forem verdadeiras.

O objetivo é permitir uma melhor avaliação do arquivo de entrada quanto a outros termos relacionados ao termo testado na condição de pesquisa. Desta forma, são analisadas as características sintáticas que envolvem um termo do arquivo de entrada. A configuração de uma condição de verificação é realizada através de um formulário (FIGURA 7.10), contendo os seguintes campos:

- **Ordem**: Indica a ordem de verificação entre as várias condições de verificação definidas em uma mesma regra, ou seja, quais serão testadas primeiramente.

- **Tipo**: Identifica o tipo de função de verificação a ser utilizado. Dependendo do tipo selecionado pelo usuário, alguns outros campos da ficha de condição de verificação serão desabilitados, passando da cor preta para a cor cinza claro. Os campos desabilitados não poderão ser preenchidos, pois não são válidos para a função escolhida.

A seguir são descritas as funções de condição de verificação:

- Caracter: Caracteres declarados diretamente na condição de verificação para comparação com os caracteres advindos do arquivo de entrada.

- *String*: termos declarados diretamente na condição de verificação para comparação com os termos advindos do arquivo de entrada.

- Base de Dados ou Dicionário: Os termos para comparação com o texto de entrada estão declarados em um dicionário de termos, que pode conter um ou mais termos, sendo que cada termo pode ser composto de uma ou mais palavras. Caso o termo de entrada exista no dicionário de termos a condição é verdadeira.

- Data: Esta função verifica se um termo do texto de entrada é uma data válida, conforme um conjunto de padrões de formatos de datas pré-definidos no extrator semântico (item 7.2.9).

- **Vezes**: número de vezes que uma determinada condição deve ocorrer dentro da distância estabelecida igualmente na ficha de condição de verificação.

- **Caixa**: Define se, na comparação entre os caracteres que compõem o termo de verificação e os termos contidos no texto de entrada, a caixa dos caracteres é levada em conta para fins de comparação. Quando esta característica não é relevante, seleciona-se a opção **Livre**; ou seja, a caixa não influi no processo de comparação; caso contrário, seleciona-se a opção **Especificada**, indicando que a caixa influi no processo de comparação.

- **Distância**: delimita a distância de validade da condição de verificação, tendo como referência o ponto (termo) em que a condição de pesquisa associada (da mesma regra) foi dada como verdadeira. A distância pode ser medida em palavras, linhas (até um caractere de quebra de linha), frases (até um caractere '.') ou parágrafos (até encontrar dois ou mais caracteres de quebra de linha subseqüentes).

- **Sentido**: caracteriza se a condição de verificação deve ser testada na parte do texto de entrada anterior ou posterior a condição de pesquisa avaliada como verdadeira.

Cabe salientar que uma condição de verificação pode ser combinada com outras, ou seja, podem ser especificadas varias condições de verificação para uma mesma condição de pesquisa. Desta forma, as diferentes condições são ligadas por um operador do tipo AND. Esta característica faz com que se uma entre todas as condições de verificação for falsa, a regra estará invalidada quanto a este grupo de condições, não sendo executadas as ações declaradas nesta mesma regra.

Para utilizar este recurso de várias condições de verificação em uma mesma regra, o usuário deve incluir as várias condições através da tabela de condições de verificação, usando o **botão Incluir**.

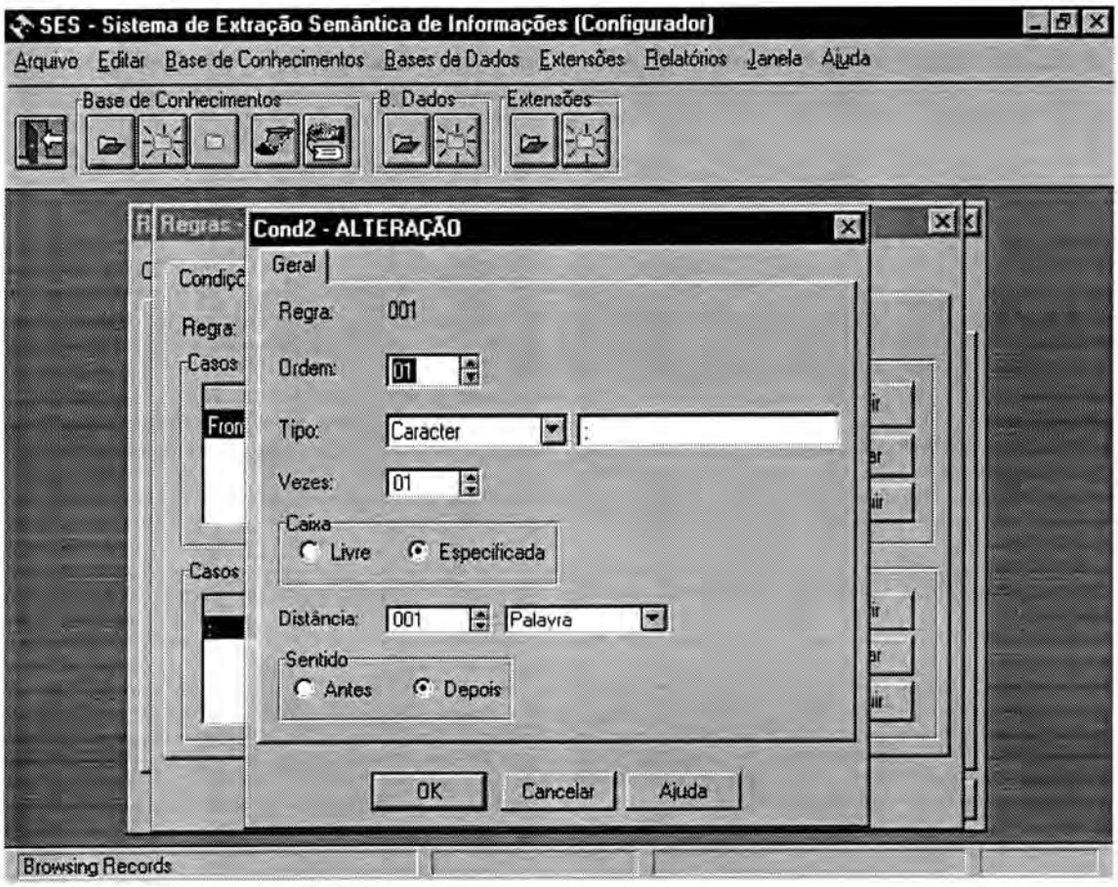

FIGURA 7.10 - Formulário de Condição de Verificação

7.2.5.4 Ações

Após a configuração das condições (pesquisa e verificação), o usuário deverá informar como será extraída a informação desejada e contida no arquivo de entrada quando tais condições se verificarem. Isto é realizado através das duas tabela de ações que fazem parte de uma regra (FIGURA 7.8), e podem ser selecionadas a partir das "orelhas" na parte superior da ficha de regra.

Conforme o modelo condicional de regra utilizado neste trabalho, SE [Condição] ENTÃO [Ação A] SENÃO [Ação B], uma ação A é executada quando a condição que compõe a regra for verdadeira; ou, caso contrário, uma ação B pode ser executada quando esta condição for falsa.

Contudo, a condição que compõe uma regra está dividida em duas partes: pesquisa e verificação. Neste sentido, definiu-se que tanto a ação positiva (então) quanto a negativa (senão) só seriam executadas quando a condição de pesquisa fosse verdadeira. Desta forma, caso esta primeira condição seja verdadeira, fica a cargo da condição de verificação definir qual das ações será realizada, conforme o resultado do teste condicional de verificação.

Tanto as ações positivas, como as negativas, utilizam o mesmo conjunto de funções, as quais servirão para extrair a informação do texto, conforme definição do usuário. A configuração de uma ação é realizada através de um formulário (FIGURA 7.11) que permite a inclusão, alteração e exclusão de ações, conforme os botões à direita das tabelas de ações. Este formulário contém os seguintes campos:

- **Ordem**: Indica a ordem de execução entre as várias ações definidas em uma mesma regra, ou seja, quais serão executadas primeiramente.

- **Função**: Seleciona qual função de extração será utilizada para extração. Dependendo da função selecionada pelo usuário, alguns outros campos da ficha de ação serão desabilitados, passando da cor preta para a cor cinza claro. Os campos desabilitados não poderão ser preenchidos, pois não são válidos para a função escolhida.

A seguir são apresentadas as funções de extração existentes no SES:

- Desloca: Avança ou retrocede o ponteiro de leitura do texto de entrada baseado em uma distância informada pelo usuário, permitido controlar o andamento do processo de extração.

- DeslocaAte: Realiza o deslocamento do ponteiro de leitura baseado em um ponto de referência informado pelo usuário do sistema. Este ponto de referência pode ser um caractere, um termo (*string*), uma data ou um termo de uma base de dados. Desta forma, o ponteiro é deslocado do ponto atual até o ponto definido pelo usuário.

- Copia: Extrai o conteúdo do arquivo de entrada baseado em uma distância informada pelo usuário do sistema. Desta forma, todo o conteúdo entre a posição atual do ponteiro de leitura e o limite da distância informada são copiados para o arquivo de saída.

- CopiaAte: Efetua a extração do conteúdo do arquivo de entrada baseado em um ponto de referência informado pelo usuário do sistema. Este ponto de referência pode ser um caractere, um termo (*string*), uma data ou um termo de uma base de dados (dicionário). Desta forma, é extraído o trecho do arquivo de entrada entre a posição atual do ponteiro e o ponto definido pelo usuário.

- CopiaCond: Realiza a cópia do termo testado e validado na condição de pesquisa para o arquivo de saída.

- CopiaTexto: Permite que o usuário coloque no arquivo de saída informações (textos) não provenientes do arquivo de entrada sob extração.

- **Tipo**: Identifica o tipo do parâmetro associado à função de extração a ser utilizada:

- Caracter: Caracteres declarados diretamente na ação para comparação com os caracteres advindos do arquivo de entrada.

- *String*: termos declarados diretamente na ação para comparação com os termos advindos do arquivo de entrada.

- Base de Dados ou Dicionário: Os termos para comparação com o texto de entrada estão declarados em um dicionário de termos, que pode conter um ou mais termos, sendo que cada termo pode ser composto de uma ou mais palavras. Caso o termo de entrada exista no dicionário de termos a condição é verdadeira.

- Data: Esta função verifica um termo do texto de entrada é uma data válida, conforme um conjunto de padrões de formatos de datas pré-definidos no extrator semântico (item 7.2.9).

- **Vezes**: número de vezes que uma determinada condição deve ocorrer dentro da distância estabelecida igualmente na função.

- **Caixa**: Define se, na comparação entre os caracteres que compõem o termo de comparação e os termos contidos no texto de entrada, a caixa dos caracteres é levada em conta para fins de comparação. Quando esta característica não é relevante, seleciona-se a opção **Livre**; ou seja, a caixa não influi no processo de comparação; caso contrário, seleciona-se a opção **Especificada**, indicando que a caixa influi no processo de comparação.

- **Distância**: delimita a distância de validade da ação, tendo como referência o ponto (termo) em que a condição de pesquisa associada foi dada como verdadeira. A distância pode ser medida em palavras, linhas (até um caractere de quebra de linha), frases (até um caractere '.') ou parágrafos (até encontrar dois ou mais caracteres de quebra de linha subseqüentes).

- **Altera Ponteiro de Pesquisa**: Indica se alterações realizadas sobre o ponteiro de pesquisa por uma ação são válidas somente para a regra corrente ou para as demais a serem executadas após a alteração.

- **Sentido**: caracteriza se a extração deve ser realizada sobre os termos anteriores ou posteriores à condição de pesquisa avaliada como verdadeira.

- **Grupo**: Associa um grupo de assunto a uma ação de extração. Assim, os dados extraídos a partir desta ação serão armazenados em um determinado grupo. Caso o usuário não saiba o código do grupo de assunto, ele pode pressionar o botão ao lado do campo grupo de assunto, abrindo a tabela de grupos para a seleção do grupo desejado.

Cabe salientar que várias ações podem ser combinadas em uma mesma regra. Esta característica permite a execução de uma seqüência de ações baseadas em um mesmo conjunto de condições, conforme a ordenação indicada no campo ordem.

Para utilizar este recurso de várias ações, o usuário pode incluir mais de uma ação na mesma regra através da tabela de ações ENTÃO ou da tabela de ações SENÃO, usando o botão **Incluir**.

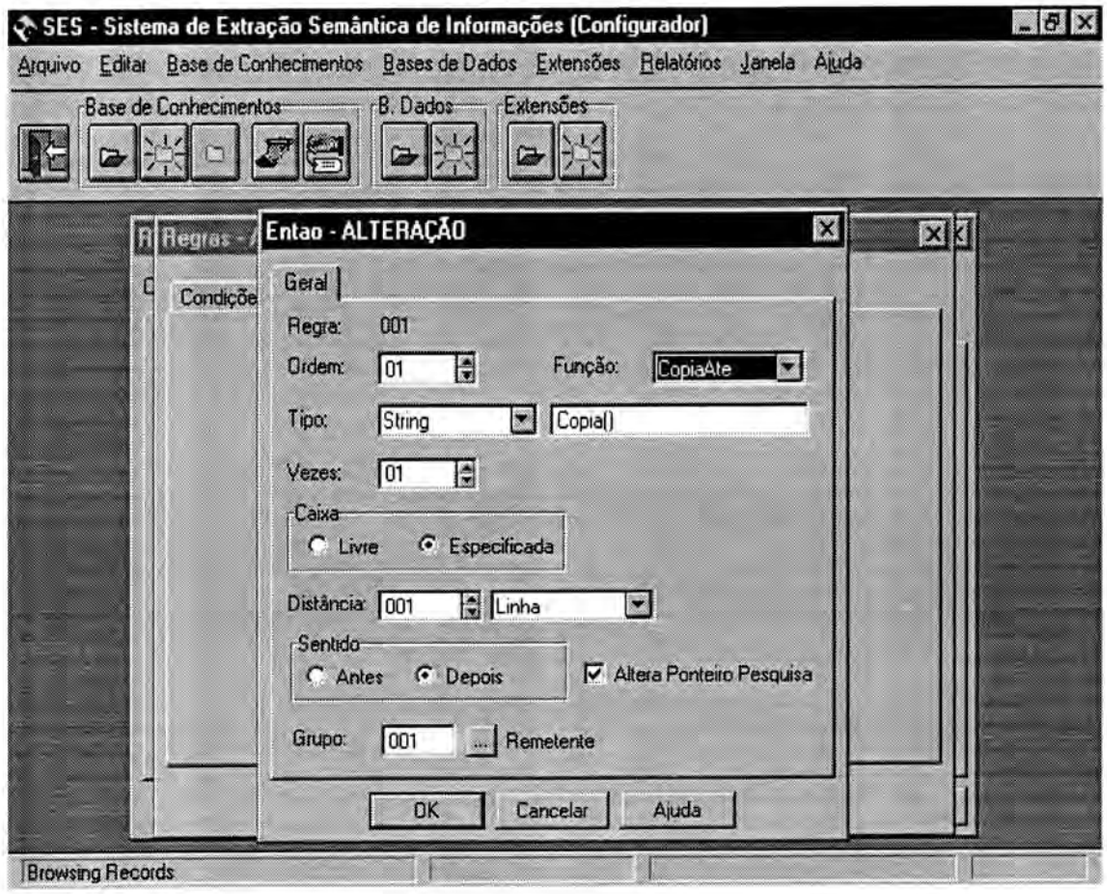

FIGURA 7.11 - Formulário de Ação

7.2.5.5 Separadores de Termos

A configuração dos separadores de termos é a fase final de configuração de uma base de conhecimento. Para configurar os separadores, o usuário deve selecionar o item de menu **Base de Conhecimento/Separadores**. Contudo, é imprescindível que, anteriormente, tenha sido aberta a base de conhecimento a ser configurada.

Os separadores são um conjunto de caracteres associados a uma base de conhecimento. Tais caracteres são entendidos pelo sistema extrator como identificadores de determinadas situações (quebras de linha e separadores de palavras). Além disso, existem caracteres que não devem ser considerados pelo SES, pois não são importantes ao processo de extração.

A configuração dos separadores é realizada pelo usuário do sistema através das tabelas de separadores (FIGURA 7.12) contida na base de conhecimento correntemente aberta; ou seja, cada base tem associada a si um conjunto independente de separadores.

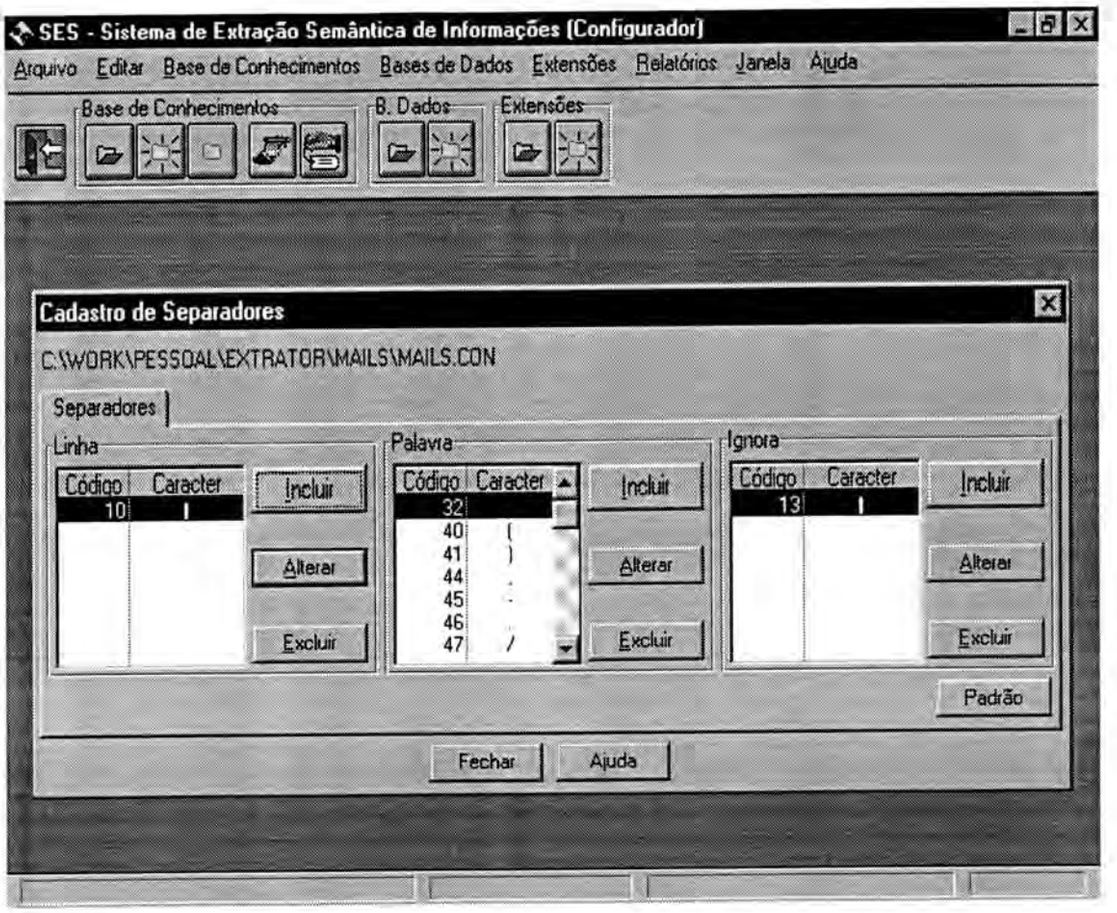

FIGURA 7.12 - Tabela de Separadores

Na realidade, existem três tabelas independentes: caracteres de quebra de linha, caracteres de separadores de palavras e caracteres a serem ignorados pelo SES. Cada um desses conjuntos de caracteres tem uma tabela e um conjunto de botões à direita desta para sua manipulação:

- **Botão Incluir** ou **Tecla *Insert***: Após pressionado o botão para a inclusão, será aberta a ficha de caracter, que deve ser preenchida com o novo caracter, pressionando-se, a seguir, o botão OK.

- **Botão Alterar** ou **Tecla Enter**: Altera o caracter destacado pela barra de seleção da tabela à esquerda do botão selecionado. Neste sentido, quando pressionado o botão alterar, a ficha de caracter será aberta para alteração. Após a alteração, o usuário deve pressionar OK para confirmar a mesma, ou CANCELA desistindo do processo de alteração.

- **Botão Excluir** ou **Tecla *Delete***: Exclui o caracter destacada pela barra de seleção da tabela à esquerda do botão selecionado. Contudo, antes da exclusão efetiva, será perguntado ao usuário se este realmente deseja prosseguir com o processo de exclusão, devendo ser respondido, através dos botões, SIM ou NÃO.

- **Botão Padrão**: Inicializa a tabela de caracteres com dados pré-definidos no sistema SES

- **Botão Fechar**: Fecha a tabela de caracteres.

7.2.6 Configurando Bases de Dados - Dicionários

Independentes das bases de conhecimento, as bases de dados ou dicionários são uma forma de definição das palavras a serem pesquisadas pelo extrator. O dicionário é composto por um conjunto de arquivos onde estão armazenadas todas as palavras identificáveis pelo analisador léxico. Cada um destes arquivos pode conter um grupo de palavras relacionadas entre si, como, por exemplo, nomes de países, formando um *thesaurus*.

7.2.6.1 Abrindo uma Base de Dados

Para a configuração de uma base de dados, o usuário deve indicar ao sistema com qual base irá trabalhar. O acesso a estas bases é realizado através do menu **Base de Dados/Abrir** ou do **sétimo botão** da barra de botões. Neste caso, o usuário esta abrindo uma base de dados já existente para alteração e/ou visualização.

Selecionada a opção Abrir do menu ou pressionado o segundo botão, será aberta uma caixa de diálogo que permite a navegação entre os discos e diretórios existentes no computador do usuário, para que este possa localizar e selecionar a base de dados que deseja manipular. É importante lembrar que os arquivos físicos que contêm as bases de dados utilizam, normalmente, a extensão **.DAD**.

Após selecionada a base de dados e pressionado o botão OK da caixa de diálogo, as opções do menu relacionadas com a configuração de bases de dados serão habilitadas, passando da cor cinza clara para preto e permitindo a edição da base selecionada e aberta.

7.2.6.2 Criando uma Nova Base de Dados

Quando o usuário do sistema SES deseja criar uma nova base de dados sem que esta contenha previamente termos já incluídos, ele deverá utilizar o item de menu **Base de Dados/Novo** ou o **oitavo botão** da barra de botões. Neste caso, o usuário está gerando uma nova base de dados vazia (sem termos).

Selecionada a opção Novo do menu ou pressionado o oitavo botão, será aberta uma caixa de diálogo que permite a navegação entre os discos e diretórios existentes no computador do usuário, para que este possa localizar onde a nova base de dados será gravada. Além da localização desejada, o usuário deve informar o nome da nova base a ser criada. Deve-se observar que os arquivos físicos que contêm as bases de dados utilizam, normalmente, a extensão **.DAD**.

Após a criação da nova base de dados, esta é automaticamente aberta para configuração, e as opções do menu relacionadas com a configuração de bases de dados

serão habilitadas, passando da cor cinza clara para preto e permitindo a edição da nova base.

Por outro lado, é importante lembrar que o usuário do SES tem a possibilidade de gerar uma nova base de dados a partir de uma base já existente. Neste caso, deve-se abrir a base de dados original, conforme o procedimento explicado em 7.2.6.1, e selecionar o menu **Base de Dados/Salvar Como**. Esta operação irá criar uma nova base de dados a partir da originalmente aberta, copiando todo seu conteúdo de termos.

Sempre que a opção **Base de Dados/Salvar Como** for utilizada, será aberta uma caixa de diálogo onde o usuário irá informar o diretório e o nome da nova base a ser criada.

7.2.6.3 Tabela de Termos

A tabela de termos (FIGURA 7.13) permite a configuração da base de dados, apresentando os termos que compõem a base correntemente aberta em ordem alfabética dos mesmos.

FIGURA 7.13 - Tabela de Termos

Para o manuseio desta tabela deve-se utilizar os botões existentes na mesma:

- **Botão Incluir** ou **Tecla *Insert***: Após pressionado o botão para a inclusão, será aberta a ficha de termo, onde deve ser preenchido o termo a ser incluído, pressionando-se o botão OK a seguir. Desta forma, será confirmada a inclusão do termo, que passará a fazer parte da tabela de termos.

- **Botão Alterar** ou **Tecla Enter**: Altera o termo destacado pela barra de seleção da tabela de termos. Neste sentido, quando pressionado o botão alterar, a ficha de termo será aberta, apresentando o termo a ser alterado. Após a alteração, o usuário deve pressionar OK para confirmar a mesma, ou CANCELA desistindo do processo de alteração.

- **Botão Excluir** ou **Tecla *Delete***: Exclui o termo destacado pela barra de seleção da tabela de termos. Contudo, antes da exclusão efetiva, será perguntado ao usuário se este realmente deseja prosseguir com o processo de exclusão, devendo ser respondido, através dos botões, SIM ou NÃO.

- **Botão Fechar**: Fecha a tabela de termos.

- **Botão Ajuda**: Ativa o texto de auxílio ao usuário referente a tabela de termos.

7.2.7 Configurando Bases de Extensões - Bases de Conhecimento Auxiliares

A base de conhecimento auxiliar ou bases de extensões têm o objetivo de guardar informações que associam as extensões dos nomes dos arquivos de entrada com as bases de conhecimento correspondentes, necessárias ao processo de extração. Este relacionamento é usado para a classificação dos arquivos de entrada, selecionando o conjunto de regras, ou seja, a base de conhecimento propícia ao processo de extração semântica de informações, conforme a extensão do arquivo de entrada.

7.2.7.1 Abrindo uma Base de Extensões

Para a configuração de uma base de extensões, o usuário deve indicar ao sistema com qual base irá trabalhar. O acesso a estas bases é realizado através do item de menu **Extensões/Abrir** ou do **nono botão** da barra de botões. Neste caso, o usuário está abrindo uma base de extensões já existente para alteração e/ou visualização.

Selecionada a opção Abrir do menu ou pressionado o nono botão, será aberta uma caixa de diálogo que permite a navegação entre os discos e diretórios existentes no computador do usuário, para que este possa localizar e selecionar a base de extensões que deseja manipular. É importante lembrar que os arquivos físicos que contêm as bases de extensões utilizam, normalmente, a extensão **.EXT**.

Após selecionada a base de extensões e pressionado o botão OK da caixa de diálogo, as opções do menu relacionadas com a configuração de bases de extensões

serão habilitadas, passando da cor cinza clara para preto e permitindo a edição da base selecionada e aberta.

7.2.7.2 Criando uma Nova Base de Extensões

Quando o usuário do sistema SES deseja criar uma nova base de extensões sem que esta contenha previamente informações incluídas, este deverá utilizar o menu **Extensões/Novo** ou o **décimo botão** da barra de botões. Neste caso, o usuário está gerando uma nova base de extensões vazia (sem informações).

Selecionada a opção Novo do menu ou pressionado o décimo botão, será aberta uma caixa de diálogo que permite a navegação entre os discos e diretórios existentes no computador do usuário, para que este possa localizar onde a nova base de extensões será gravada. Além da localização desejada, o usuário deve informar o nome da nova base a ser criada. Deve-se observar que os arquivos físicos que contêm as bases de dados utilizam, normalmente, a extensão **.EXT**.

Após a criação da nova base de extensões, esta é automaticamente aberta para configuração, e as opções do menu relacionadas com a configuração de bases de extensões serão habilitadas, passando da cor cinza clara para preto e permitindo a edição da nova base.

Contudo, é importante lembrar que o usuário do SES tem a possibilidade de gerar uma nova base de extensões a partir de uma base já existente. Neste caso, deve-se abrir a base de extensões original, conforme o procedimento explicado em 7.2.7.1, e selecionar o menu **Extensões/Salvar Como**. Esta operação irá criar uma nova base de extensões a partir da originalmente aberta, copiando todo seu conteúdo.

Sempre que a opção **Extensões/Salvar Como** for utilizada, será aberta uma caixa de diálogo onde o usuário irá informar o diretório e o nome da nova base a ser criada.

7.2.7.3 Tabela de Extensões

A tabela de extensões (FIGURA 7.14) permite a configuração da base de extensões, apresentando as extensões que compõem a base correntemente aberta.

Para o manuseio desta tabela deve-se utilizar os botões existentes na mesma:

- **Botão Incluir** ou **Tecla *Insert***: Após pressionado o botão para a inclusão, será aberta a ficha de extensão (FIGURA 7.15), onde devem ser preenchidos os campos referentes a extensão a ser incluída e pressionado o botão OK. Desta forma, será confirmada a inclusão da nova extensão, que passará a fazer parte da tabela de extensões.

- **Botão Alterar** ou **Tecla Enter**: Altera a extensão destacada pela barra de seleção da tabela de extensões. Neste sentido, quando pressionado o botão alterar, a ficha de extensão (FIGURA 7.15) será aberta, apresentando os

dados referentes a extensão a ser alterada. Após a alteração, o usuário deve pressionar OK para confirmar a mesma, ou CANCELA desistindo do processo de alteração.

- **Botão Excluir** ou **Tecla *Delete***: Exclui a extensão destacada pela barra de seleção da tabela. Contudo, antes da exclusão efetiva, será perguntado ao usuário se este realmente deseja prosseguir com o processo de exclusão, devendo ser respondido, através dos botões, SIM ou NÃO.

- **Botão Fechar**: Fecha a tabela de extensões.

- **Botão Ajuda**: Ativa o texto de auxílio ao usuário referente à tabela de extensões.

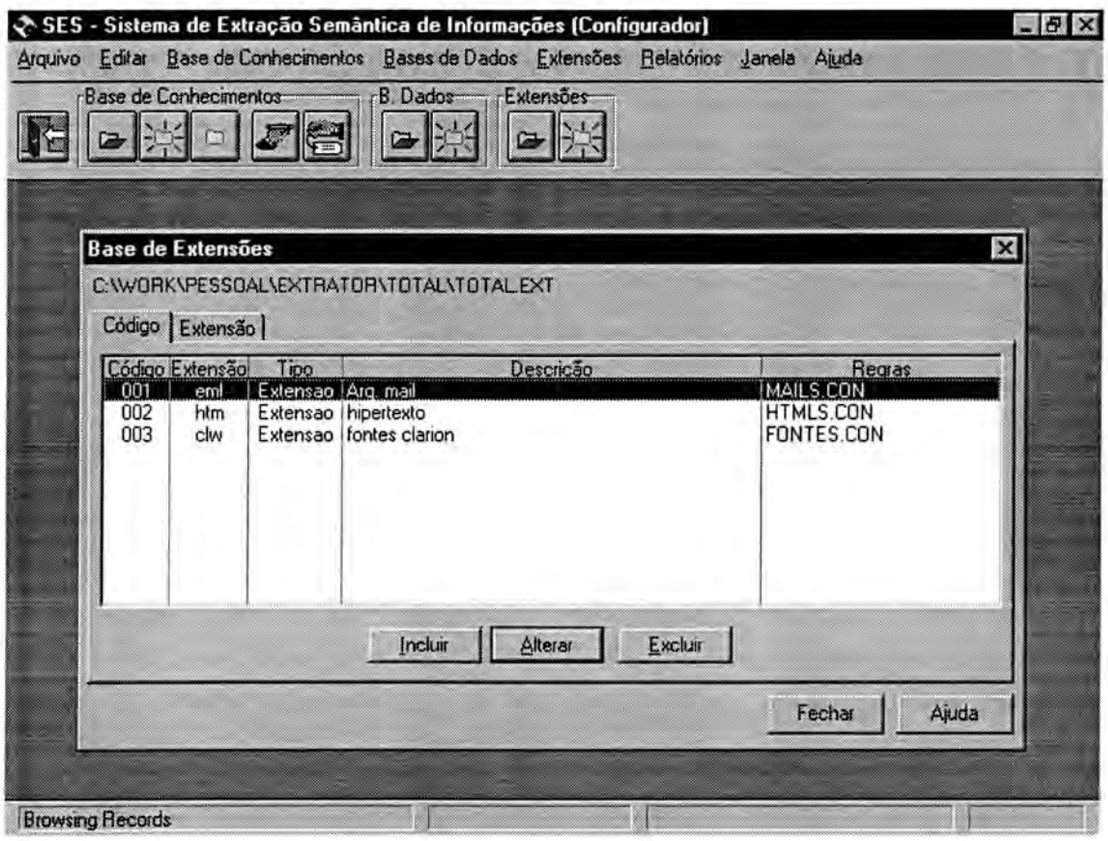

FIGURA 7.14 - Tabela de Extensões

7.2.7.4 Ficha de Extensão

A ficha de extensão permite a configuração dos dados associados a uma extensão, ou seja, a seleção da base de conhecimento que contém as regras de extração para os arquivos de entrada com esta extensão. Esta ficha (FIGURA 7.15) contém os seguintes campos:

- **Extensão**: caracteres que formam a extensão a ser analisada até o tamanho máximo de três caracteres.

- **Tipo**: Identifica o tipo de extensão informada pelo usuário. Dependendo do tipo selecionado pelo usuário, alguns outros campos da ficha de extensão serão desabilitados, passando da cor preta para a cor cinza claro. Os campos desabilitados não poderão ser preenchidos, pois não são válidos para o tipo escolhido. A seguir são descritos os tipo de extensão definidos no SES:

- Extensão: Seleciona somente os arquivos de entrada cuja extensão é igual à informada no campo anterior.

- Nula: Seleciona os arquivos de entrada sem extensão.

- Outras: Seleciona todos os arquivos de entrada cuja extensão não foi especificada na tabela de extensões.

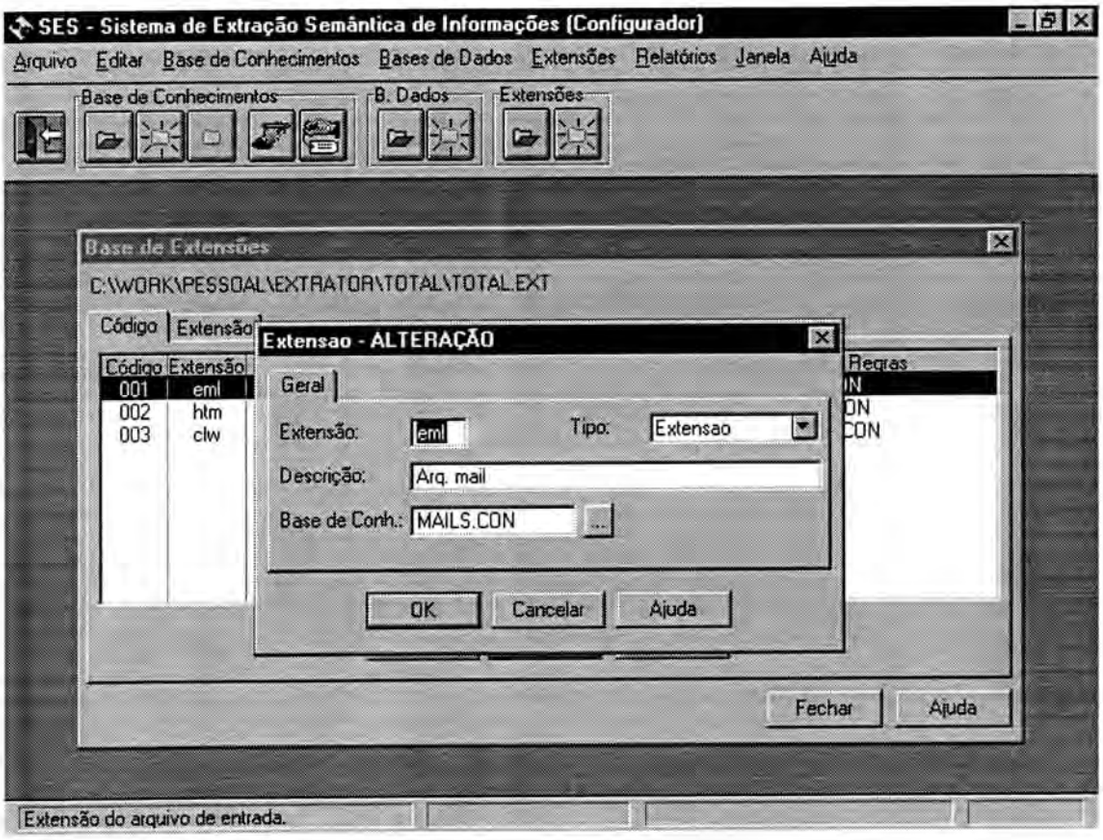

FIGURA 7.15 - Ficha de Extensão

- **Descrição**: Descrição livre informada pelo usuário do sistema, identificando o tipo de arquivo relacionado com a extensão especificada na ficha.

- **Base de Conhecimento**: nome da base de conhecimento associada à extensão sob manipulação. Todas as bases de conhecimento utilizadas por uma base de

extensão devem estar no mesmo diretório. Caso o usuário não saiba o nome do arquivo físico da base de conhecimento que deseja associar à extensão, ele pode pressionar o botão ao lado do campo base de conhecimento, abrindo uma caixa de diálogo para a procura da localização desta base em seu computador.

7.2.8 Relatórios

O módulo Configurador permite a impressão de alguns relatórios para a melhor visualização do conteúdo das bases de conhecimento.

- **Relatório de Regras**: Este relatório é gerado quando selecionada a opção de menu **Relatórios/Regras**. O objetivo é apresentar o conteúdo (regras) que compõe a base de conhecimento correntemente aberta.

- **Relatório de Grupos**: Este relatório é gerado quando selecionada a opção de menu **Relatórios/Grupos**. O objetivo é apresentar o conteúdo (grupos de assuntos) que compõe a base de conhecimento correntemente aberta.

7.2.9 Formatos de Datas Pré-Definidos no SES

A fim de realizar a extração de informações temporais a partir dos arquivos de entrada, foram definidos vários formatos de datas que são identificáveis pelo sistema SES. A seguir serão apresentados os formatos definidos:

- DD/MM/AA
- DD/MM/AAAA
- DD-MM-AA
- DD-MM-AAAA
- MM/DD/AA
- MM/DD/AAAA
- MM-DD-AA
- MM-DD-AAAA
- Dia_Sem, DD Mês_Ext AAAA
- Dia_Sem, DD de Mês_Ext de AAAA
- Mês_Ext DD, AAAA
- Mês_Ext, AAAA
- Mês_Ext/AA
- Mês_Ext-AA
- DD de Mês_Ext de AAAA
- DD/Mês_Ext/AA
- DD/Mês_Ext/AAAA
- DD-Mês_Ext-AA
- DD-Mês_Ext-AAAA
- Mês_Ext DD - DD, AAAA

DD → Dia do mês (numérico).
MM → Mês (numérico).
AA → Ano (numérico - 2 dígitos).
AAAA → Ano (numérico - 4 dígitos).
Mês_Ext → Mês do ano por extenso.
Dia_Sem → Dia da semana por extenso.

É importante salientar que os dias da semana por extenso, bem como os meses do ano por extenso, estão armazenados em duas bases de dados (dicionários), respectivamente, as quais deverão ser configuradas pelo usuário do sistema SES. Na utilização do módulo extrator será necessário informar a localização de tais bases para a extração de informações temporais.

7.2.10 Saindo do Módulo Configurador

Caso o usuário deseje sair do módulo Configurador, ele pode utilizar a opção de menu **Arquivo/Sair** ou o **primeiro botão** da barra de botões.

7.3 Extrator

O Extrator é um dos executáveis que compõem o sistema SES. Seu objetivo é realizar a extração das informações armazenadas nos arquivos de entrada. Este processo de extração é realizado conforme as regras configuradas nas bases de conhecimento.

FIGURA 7.16 - Ícone do Extrator

Para inicializar o Extrator, o usuário deverá dar um *double-click* sobre o ícone que representa este executável (FIGURA 7.16). Realizada esta operação o usuário se encontrará dentro do ambiente de trabalho do módulo Extrator (FIGURA 7.17).

7.3.1 Ambiente do Extrator

O usuário do Sistema de Extração Semântica de Informações ao entrar no módulo Extrator irá posicionar-se dentro do ambiente de trabalho deste módulo (FIGURA 7.17). É neste ambiente que é realizado o processo de extração.

Cabe salientar que, sendo um programa desenvolvido para o ambiente Windows 95, o usuário tem a possibilidade de realizar algumas alterações sobre o ambiente de trabalho do SES. Neste sentido, sempre que alteradas as características do Windows quanto ao ambiente, o SES irá adaptar-se as alterações definidas pelo usuário.

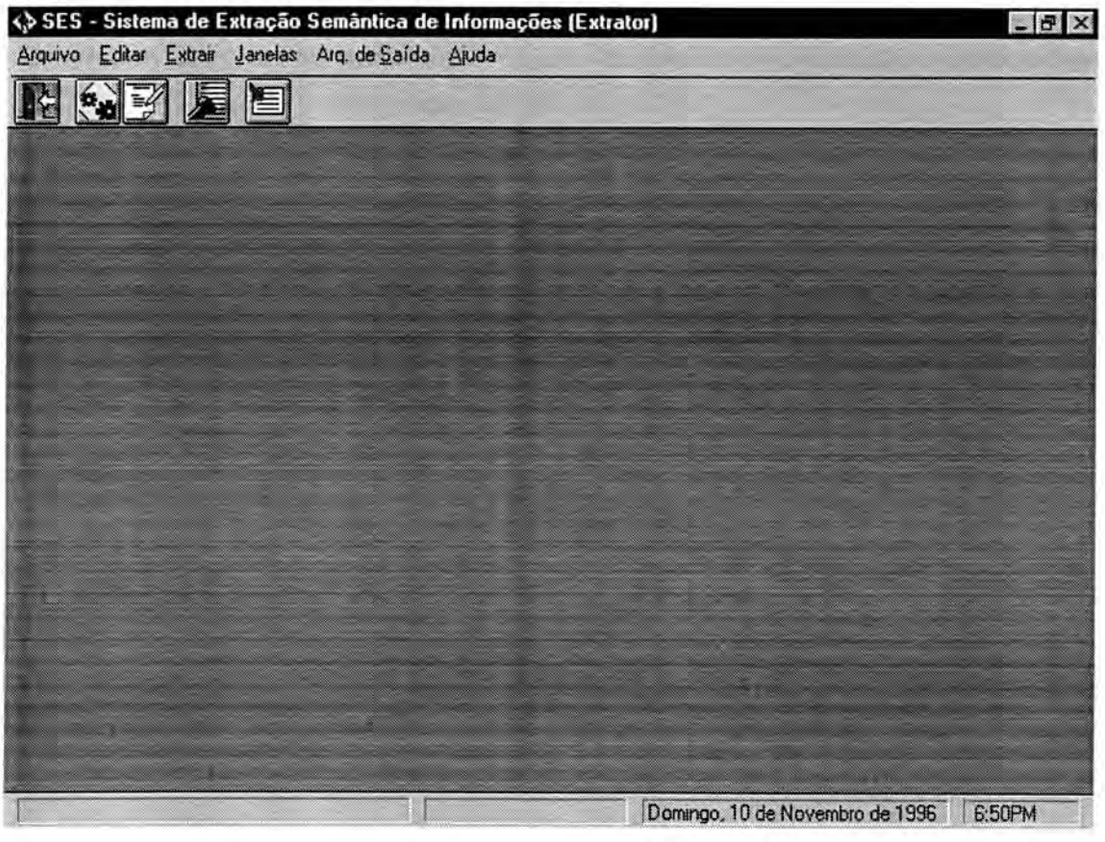

FIGURA 7.17 - Ambiente de Trabalho do Configurador

O ambiente de trabalho apresentado após a execução do Extrator pode ser dividido em cinco partes:

- Barra de Título: informa ao usuário o nome do sistema de extração (SES), apresentando seu ícone (FIGURA 7.16). Além disso, permite que este usuário minimize ou maximize a janela da aplicação, bem como finalize a mesma (botões no canto superior direito definidos no ambiente Windows 95).

- Menu Principal: contém o conjunto de operações possíveis de serem utilizadas na extração das informações.

- Barra de Botões: abaixo do menu principal, a barra de botões contém um conjunto de botões para acesso rápido as operações do Extrator.

- Área de Trabalho: local onde o usuário irá acompanhar e controlar o processo de extração semântica.

- Barra de *Status*: posicionada na parte inferior da tela, a barra de *status* informa ao usuário o andamento de cada uma de suas ações dentro do sistema, bem como sua localização entre as diversas tabelas que fazem parte do módulo de extração.

7.3.2 Chamando o Módulo Configurador Através do Extrator

A fim de facilitar a configuração do processo de extração, o módulo Extrator, através do item de menu **Arquivo/Configurador** ou do **segundo botão** da barra de botões, permite a ativação do módulo de configuração. O objetivo é viabilizar que o usuário do sistema SES possa de forma rápida alternar de um módulo para outro, o que pode ser necessário, principalmente durante o processo de configuração das bases de conhecimento.

No momento da configuração, o usuário tem de executar várias vezes o processo de extração, verificando os resultados obtidos para cada configuração realizada da base de conhecimento. Conforme o resultado, novas alterações e chamadas ao Configurador serão necessárias, até que o resultado satisfatório seja atingido.

7.3.3 Inicialização do Processo de Extração

O módulo Extrator necessita de um conjunto de dados para a execução do processo de extração. Estes dados são informados pelo usuário do sistema SES através da opção de menu **Arquivo/Inicialização** ou pressionando o **terceiro botão** da barra de botões.

Sempre que a opção Inicialização for utilizada, será aberta uma ficha de inicialização (FIGURA 7.18), contendo os campos cujos dados deverão ser informados pelo usuário.

A seguir serão descritos os campos que compõem a ficha de inicialização:

- **Arquivo de Extensões**: Informação de qual base de extensões utilizar no processo de extração. Caso o usuário não saiba o nome do arquivo físico da base de extensões que deseja selecionar para a extração, ele pode pressionar o botão ao lado do campo arquivo de extensões, abrindo uma caixa de diálogo para a procura da localização desta base em seu computador.

- **Diretório de Regras**: Informação do diretório onde se encontram as bases de conhecimento que irão ser utilizadas pelo processo de extração, conforme definição realizada na configuração da base de extensões. É importante salientar que todas as bases de conhecimento utilizadas por uma base de extensão devem estar no mesmo diretório. Caso o usuário não saiba a localização das bases de conhecimento, ele pode pressionar o botão ao lado do campo diretório de regras, abrindo uma caixa de diálogo para a procura do diretório desejado.

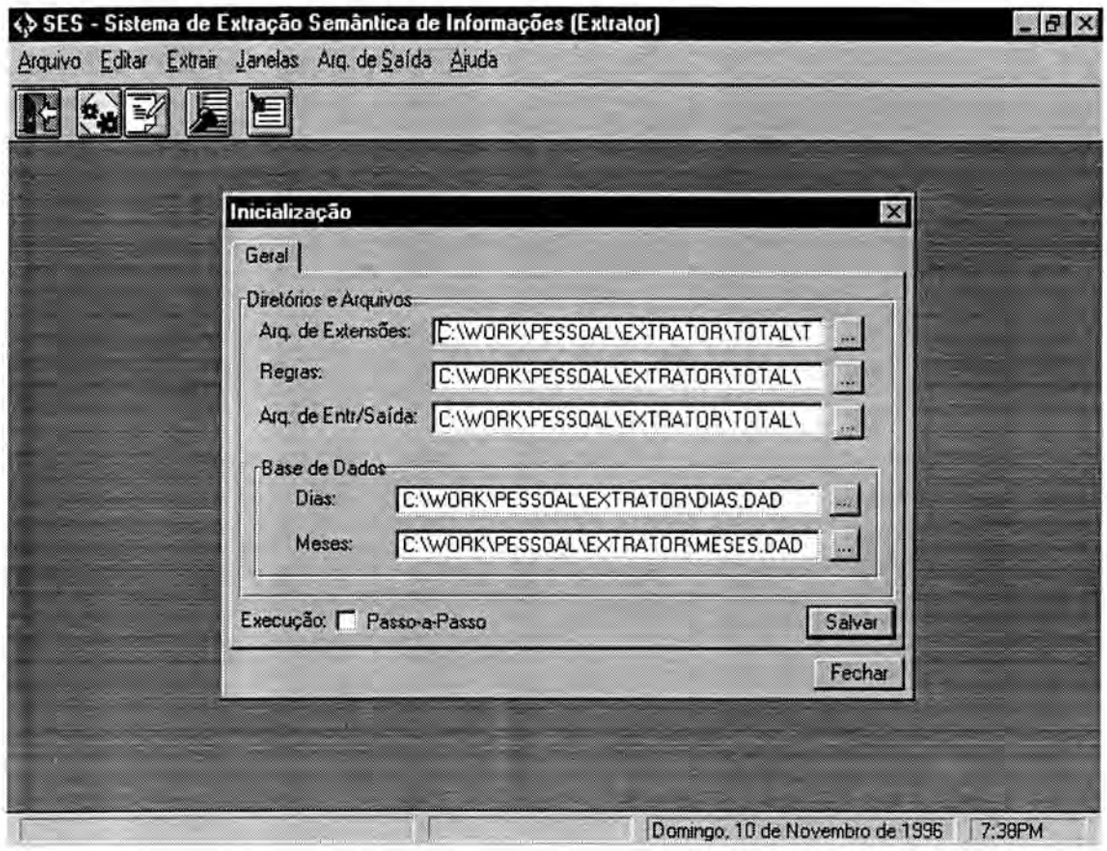

FIGURA 7.18 - Ficha de Inicialização

- **Arquivos de Entrada e Saída**: Informação do diretório onde se encontram os arquivos de entrada a serem processados pelo extrator. Além disso, neste mesmo diretório será gravado o arquivo de saída, com os resultados do processo de extração, e o arquivo de *log*, com os passos executados durante o processo de extração. Caso o usuário não saiba a localização dos arquivos de entrada, ele pode pressionar o botão ao lado do campo arquivos de entrada e saída, abrindo uma caixa de diálogo para a procura do diretório desejado.

- **Base de Dados de Dias**: Neste campo da ficha de inicialização, deve ser informado qual base de dados (dicionário) com os dias da semana por extenso será utilizada para a extração dos dados temporais (item 7.2.9). Caso o usuário não saiba o nome do arquivo físico da base de dados que deseja selecionar, ele pode pressionar o botão ao lado do campo base de dados de

dias, abrindo uma caixa de diálogo para a procura da localização desta base em seu computador.

- **Base de Dados de Meses**: Informação da base de dados (dicionário) com os meses do ano por extenso, a qual será utilizada para a extração dos dados temporais (item 7.2.9). Caso o usuário não saiba o nome do arquivo físico da base de dados que deseja selecionar, ele pode pressionar o botão ao lado do campo base de dados de meses, abrindo uma caixa de diálogo para a procura da localização desta base em seu computador.

- **Execução passo-a-passo**: Caso esta opção seja selecionada, os dados que estão sendo gravados no arquivo de *Log* do extrator são mostrados em tela durante o processamento dos arquivos de entrada. Tais dados informam todos os testes realizados na extração semântica das informações, bem como seus resultados e as informações extraídas.

- **Botão Salvar**: Salva os dados informados pelo usuário. Assim, na próxima execução do módulo Extrator, os dados informados anteriormente já estarão presentes na ficha de inicialização.

- **Botão Fechar**: Finaliza a edição da ficha de inicialização.

7.3.4 Extraindo Informações

A ativação do processo de extração é realizada através do item de menu **Extrair** ou pressionando-se o **quarto botão** da barra de botões. Neste momento, é aberta a janela de controle de extração (FIGURA 7.19), onde pode-se acompanhar o andamento de todo o processo a ser realizado.

Para iniciar o processo de extração, o usuário deve pressionar o **botão extrair**. A partir deste momento, o módulo Extrator começa a realizar a extração semântica de informações. Os dados extraídos são colocados no arquivo de saída (item 7.3.5).

O processamento iniciado ficará executando até o fim da leitura e extração dos dados a partir de todos os possíveis arquivos de entrada encontrados no diretório informado pelo usuário na ficha de inicialização. Ao fim deste processo, será enviada uma mensagem, através de uma caixa de diálogo, informando ao usuário o final do processamento, bem como o tempo total de extração.

No entanto, caso desejado, o usuário pode interromper o processo de extração. Para tal, deve ser pressionada a tecla ESC durante o andamento da extração. Pressionada esta tecla, será perguntado ao usuário se ele realmente deseja cancelar a extração, devendo ser respondido SIM ou NÃO.

Durante o processamento da extração, são apresentadas na janela de controle de extração algumas informações sobre o andamento da mesma:

- **Extensões**: base de extensão que esta sendo utilizada no processamento dos arquivos de entrada.

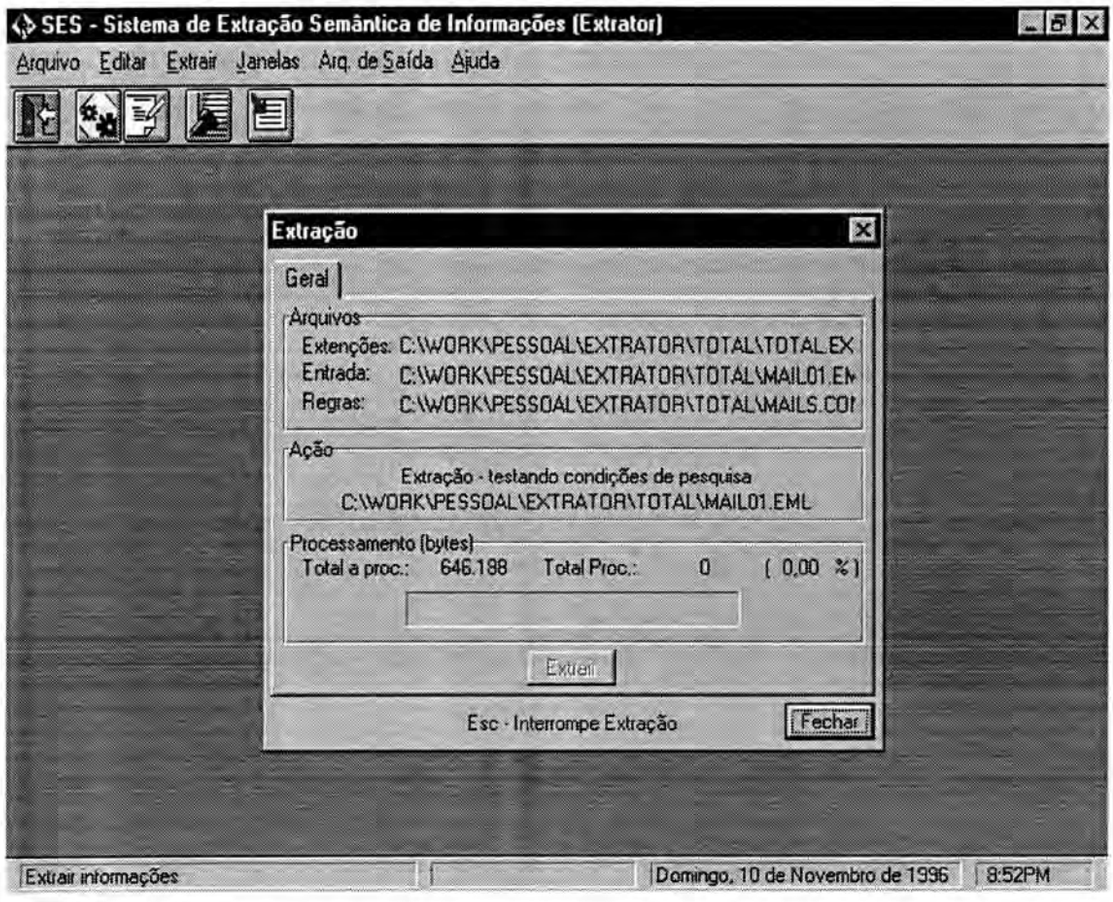

FIGURA 7.19 - Janela de Controle de Extração

- **Entrada**: Arquivo de entrada em processamento.

- **Regras**: Base de conhecimento sendo usada para a extração das informações contidas no arquivo de entrada especificado acima.

- **Ação**: Testes realizados pelo extrator na busca das informações armazenadas nos arquivos de entrada, bem como as ações de extração.

- **Processamento**: Permite acompanhar o andamento do processamento, apresentando o total de bytes a serem processados, o total já processado e a percentagem de bytes já processados em relação aos não processados.

- **Barra de Andamento**: Barra azul que indica graficamente o andamento do processo de extração, conforme o número de bytes já processados.

- **Botão Fechar**: Após o termino da extração, permite o fechamento da janela de controle de extração.

7.3.5 Arquivo de Saída

Os resultados do processo de extração, ou seja, as informações extraídas, são todas armazenadas em um arquivo ASCII de forma estruturada. Este arquivo de nome **ArqSaida.EXT** é gravado no mesmo diretório dos arquivos de entrada, e pode ser editado em um editor de textos, ou utilizado por outro sistema.

Cabe lembrar que se já existe um arquivo com este mesmo nome no diretório dos arquivos de entrada, este será substituído pelo novo arquivo de saída, e os dados anteriores serão perdidos.

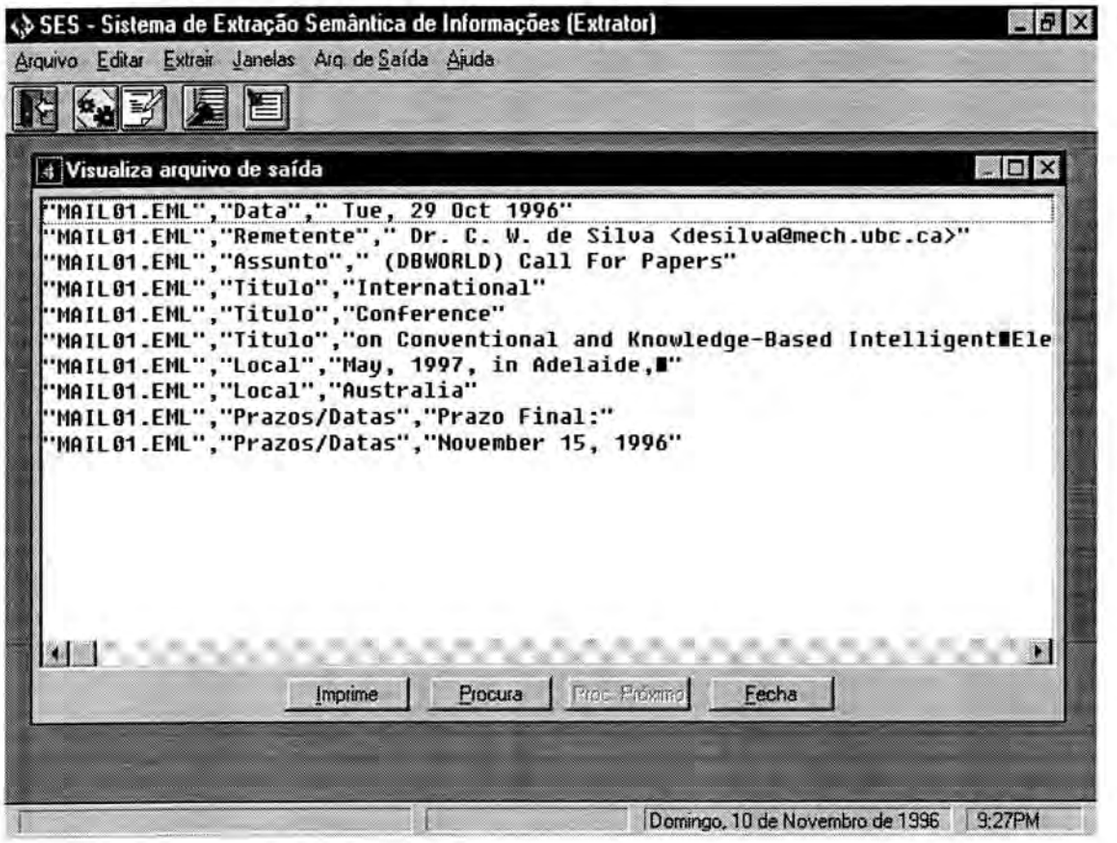

FIGURA 7.20 - Visualizador do Arquivo de Saída

O módulo Extrator facilita a visualização do arquivo de saída através do item de menu **Arq. De Saída**, ou pressionando-se o **último botão** da barra de botões. Este item do menu ativa o visualizador do arquivo de saída (FIGURA 7.20), permitindo ao usuário ter acesso as informações extraídas dentro do próprio Extrator. Além disso, o visualizador contém alguns botões com operações para o manuseio dos dados extraídos:

- **Imprime**: Imprime o arquivo de saída.
- **Procura**: Realiza a pesquisa de uma cadeia de caracteres dentro do arquivo de saída.

- **Procura Próximo**: Reitera a última pesquisa de uma cadeia de caracteres dentro do arquivo de saída, buscando uma nova ocorrência da mesma.

- **Fecha**: Fecha a janela do visualizador do arquivo de saída.

7.3.6 Arquivo de Log

Adicionalmente ao arquivo de saída gerado pelo Extrator, também é criado um arquivo de *Log* com o nome **ArqLog.EXT**. Este arquivo contém dados que informam todos os testes realizados na extração semântica das informações, bem como seus resultados e as informações extraídas. Ou seja, apresenta, passo-a-passo, o andamento do processo de extração ocorrido.

O arquivo de *Log* é um arquivo ASCII que será gravado no mesmo diretório do arquivo de saída, e pode ser editado em um editor de textos. Cabe lembrar que se já existe um arquivo com este mesmo nome no diretório do arquivo de saída, este será substituído pelo novo arquivo de *Log*, e os dados anteriores serão perdidos.

7.3.7 Saindo do Módulo Extrator

Caso o usuário deseje sair do módulo Configurador, ele pode utilizar a opção de menu **Arquivo/Sair** ou o **primeiro botão** da barra de botões.

7.4 Menu Editar

O menu editar existe nos dois executáveis do sistema SES e apresenta um conjunto de operações válidas em vários pontos do módulo Configurador ou do módulo Extrator. Sempre que tais operações forem válidas, elas serão habilitadas, passando da cor cinza claro para a preta, indicando sua ativação.

As operações possíveis são as seguintes:

- **Recortar**: copia o texto selecionado para a área de armazenamento do Windows, apagando-o do local original.

- **Copiar**: copia o texto selecionado para a área de armazenamento do Windows.

- **Colar**: Copia para a posição indicada pelo cursor o conteúdo da área de armazenamento do Windows.

7.5 Menu Janela

O item de menu janela existe nos dois executáveis do sistema SES e permite ao usuário organizar seu ambiente trabalho, posicionando de diferentes formas as janelas abertas: lado-a-lado ou em cascata.

Além disso, existe a opção de organizar os ícones das janelas que foram minimizadas pelo usuário.

7.6 Menu Ajuda

Através desta opção de menu existente nos dois executáveis do sistema SES, o usuário obtém a ajuda necessária para compreender o funcionamento do SES - Sistema de Extração Semântica de Informações, bem como a forma de navegação dentro do sistema e as regras e recursos utilizados para configurar e extrair informações.

8 Estudo de Caso - Validação

O desenvolvimento de um trabalho de pesquisa, como é o caso desta dissertação, deve apresentar pontos de validação dos resultados atingidos durante seu andamento. São tais validações que permitem verificar o desenrolar de um projeto de pesquisa, realizando a correção de falhas e desvios do objetivo final.

Neste capítulo, apresenta-se um estudo de caso aplicado do sistema SES, cujo embasamento conceitual foi descrito anteriormente. O objetivo é verificar e validar o estudo realizado no decorrer de dois anos de pesquisa, bem como de todos os conhecimentos adquiridos.

8.1 Casos de Validação

O processo de validação de um Sistema de Extração Semântica de Informações exige o uso de um conjunto de arquivos de entrada, a partir dos quais serão realizadas as extrações. Visto que o SES permite a extração de informações encontradas em arquivos com diferentes estruturas internas ou, até mesmo, arquivos não estruturados, definiu-se o uso de três tipos heterogêneos de arquivo de entrada. Para cada tipo definido foi realizada uma coletânea de arquivos, formando três grupos com trinta arquivos de entrada cada.

8.1.1 Arquivos de *Mail*

Arquivos de *Mail* são um tipo de arquivo de dados especial, pois apresentam parte de seus dados de forma estruturada e parte de forma não estruturada. O cabeçalho de um *mail* contém uma organização pré-definida e estruturada que facilita o processo de extração das informações ali contidas. Contudo, após o cabeçalho, é apresentado um texto livre, onde o autor do mesmo pode colocar um conjunto de informações organizadas conforme sua livre decisão.

8.1.1.1 Domínio dos *Mails*

Os *mails* processados pelo sistema extrator foram obtidos através de listas da rede Internet. Todos os arquivos selecionados tratavam sobre congressos na área de informática, sendo a maioria sobre chamadas de artigos para tais congressos (*Call for Paper*). Adicionalmente, cabe salientar que a maioria dos textos eram escritos em inglês, sendo somente dois entre os trinta escritos em português.

8.1.1.2 Objetivo do uso dos *Mails*

Os arquivos de *mail* utilizados nos testes realizados com o SES permitiram verificar e validar algumas das características deste sistema de extração:

- Avaliação do SES quanto ao seu uso junto a dados estruturados (cabeçalho).
- Atuação do SES junto a domínios de extração não estruturados (corpo do *mail*).
- Adaptação do extrator a diferentes tipos de línguas de origem, no caso específico, ao inglês.
- Extração de dados temporais.
- Tempo de extração.

Como complemento dos testes realizados sobre arquivos de *mail*, cabe salientar que os *mails* em português foram utilizados na base de testes para verificar os resultados de extração sobre arquivos estranhos ao domínio de extração especificado através das regras de configuração do módulo extrator do SES.

8.1.1.3 Dados a serem Extraídos dos *Mails*

A partir dos arquivos de *mail* pretende-se extrair as seguintes informações:

- Remetente do *Mail*.
- Assunto (*subject*).
- Data de envio do arquivo.
- Título do congresso sendo tratado dentro do texto.
- Local de realização do congresso.
- Data limite de envio de artigos.

8.1.1.4 Base de Conhecimento Criada para *Mails*

Grupos de Assuntos:

- Remetente
- Assunto
- Data Envio
- Título
- Local
- Prazos/Datas Importantes

Regras de Configuração:

```
Se
  CP(String("FROM", CaixaEspecificada, 0007, 0020) ) and
  CV(Caracter(":", 1 vez, CaixaEspecificada, 1, palavra, depois) )
Então
  Copia(1, linha, depois, Grupo(Remetente), AlteraPonteiro)
```

```
Se
 CP(String("SUBJECT", CaixaEspecificada, 0008, 0020) ) and
 CV(Caracter(":", 1 vez, CaixaEspecificada, 1, palavra, depois) )
Então
 Copia(1, linha, depois, Grupo(Assunto), AlteraPonteiro)

Se
 CP(String("DATE", CaixaEspecificada, 0005, 0020) ) and
 CV(Caracter(":", 1 vez, CaixaEspecificada, 1, palavra, depois) )
Então
    CopiaAté(Data(), 1 vez, CaixaLivre, 1, linha, Grupo(Data Envio),
AlteraPonteiro)

Se
 CP(String("Conference", CaixaEspecificada, 0001, 0030) ) or
 CP(String("Workshop", CaixaEspecificada, 0001, 0030) ) or
 CP(String("Symposium", CaixaEspecificada, 0001, 0030) ) or
 CP(String("Journal of,", CaixaEspecificada, 0001, 0030) )
Então
 Copia(1, linha, antes, Grupo(Título) )
 CopiaCond(Grupo(Título) )
 Copia(2, linhas, depois, Grupo(Título), AlteraPonteiro)

Se
 CP(BaseDados(Paises), CaixaLivre, 0001, 9999 ) and
 CV(Date(), 1 vez, CaixaLivre), 4, linhas, antes )
Então
 CopiaAté(Data(), 1 vez, CaixaLivre), 1, frase, antes, Grupo(Local) )
 CopiaCond(Grupo(Local) )

Se
 (CP(String("Submitted", CaixaLivre, 0001, 9999) ) or
 CP(String("Deadline"), CaixaLivre, 0001, 9999) ) or
 CP(String("Submissions"), CaixaLivre, 0001, 9999) ) or
 CP(String("Deadlines of"), CaixaLivre, 0001, 9999) ) ) and
 CV(Date(), 1 vez, CaixaLivre, 20, palavras, antes )
Então
 Copia(1, frase, depois, Grupo(Prazos/Datas Importantes) )

Se
 CP(String("IMPORTANTE DATES", CaixaLivre, 0001, 9999) ) and
 CV(Date(), 1 vez, CaixaLivre), 4, linhas, depois )
Então
 Copia(2, parágrafos, depois, Grupo(Prazos/Datas Importantes), AlteraPonteiro)
```

Se
(CP(String("later than", CaixaLivre, 0001, 9999)) or
CP(String("submitted"), CaixaLivre, 0001, 9999))) and
CV(Date(), 1 vez, CaixaLivre, 1, frase, depois)
Então
CopiaTexto(String("Prazo Final:", Grupo(Prazos/Datas Importantes))
CopiaAte(Data(), 1 vez, CaixaLivre, 1, frase, depois, Grupo(Prazos/Datas Importantes), AlteraPonteiro)

Se
CP(BaseDados(Países), CaixaLivre, 0001, 0030) and
CV(Date(), 1 vez, CaixaLivre, 2, linhas, depois)
Então
CopiaCond(Grupo(Local))
CopiaAté(Data(), 1 vez, CaixaLivre, 2, linhas, depois, Grupo(Local))

8.1.2 Arquivos HTML

Arquivos HTML, parecidos com arquivos *mail*, podem apresentar dados de forma estruturada e de forma não estruturada. Arquivos deste tipo contêm um conjunto de delimitadores pré-definidos que permitem a identificação e qualificação de alguns dados contidos no arquivo, tornando-os estruturados.

Por outro lado, grande parte do "corpo" de um arquivo HTML é formada por informações em texto livre, onde o autor organiza os dados conforme sua livre decisão.

8.1.2.1 Domínio dos Arquivos HTML Utilizados

Os arquivos HTML processados pelo sistema extrator foram obtidos através de páginas disponíveis na rede Internet. Todos os arquivos selecionados tratavam sobre monitores de computadores, mais especificamente, suas características técnicas. Neste sentido, observa-se a grande semelhança entre o conteúdo dos arquivos, pois, além de tratarem do mesmo assunto, apresentam, normalmente, o mesmo linguajar para os dados técnicos, unidades de medida, por exemplo. Adicionalmente, cabe salientar que todos os textos eram escritos em inglês.

8.1.2.2 Objetivo do uso dos Arquivos HTML

Os arquivos HTML utilizados nos testes realizados com o SES permitiram verificar e validar algumas das características deste sistema de extração:

- Atuação do SES junto a domínios de extração semi-estruturados.
- Adaptação do extrator à identificação de comandos de programação.
- Auxiliar na exploração de dados semelhantes, ressaltando as diferenças existentes.
- Extração de dados temporais.
- Tempo de extração.

8.1.2.3 Dados a serem Extraídos dos Arquivos HTML

A partir dos arquivos HTML pretende-se extrair as seguintes informações:

- Data de atualização do arquivo.
- Título da página.
- Tamanho do monitor.
- Resolução/Freqüência.
- Marca.
- *Links* para outras páginas e figuras.

8.1.2.4 Base de Conhecimento Criada para Arquivos HTML

Grupos de Assuntos:

- Data Atualização
- Título
- Tamanho
- Resolução/Freqüência
- Marca
- *Links*

Regras de Configuração:

Se
CP(String("Last modified", CaixaLivre, 0001, 9999)) and
CV(Date(), 1 vez, CaixaLivre, 1, linha, depois)
Então
CopiaCond(Grupo(Data Atualização))
Copia(1, linha, depois, Grupo(Data Atualização), AlteraPonteiro)

Se
CP(String("TITLE", CaixaEspecificada, 0001, 0010))
Então
CopiaAté(Caracter("<", 1 vez, CaixaLivre), 2, linha, depois, Grupo(Título), AlteraPonteiro)

Se
CP(String("Hz", CaixaEspecificada), 0001, 9999)) and
CV(String("vertical", 1 vez, CaixaLivre, 1, palavra, depois))
Então
Copia(1, palavra, antes, Grupo(Resolução/Freqüência))
CopiaCond(Grupo(Resolução/Freqüência))
Copia(1, palavra, depois, Grupo(Resolução/Freqüência), AlteraPonteiro)

```
Se
  CP(String("kHz", CaixaEspecificada), 0001, 9999) and
  CV(String("horizontal", 1 vez, CaixaLivre, 1, palavra, depois) )
Então
  Copia(1, palavra, antes, Grupo(Resolução/Freqüência) )
  CopiaCond(Grupo(Resolução/Freqüência) )
  Copia(1, palavra, depois, Grupo(Resolução/Freqüência), AlteraPonteiro)

Se
  CP(String("Hz", CaixaEspecificada, 0001, 9999) ) and
  CV(String("vertical", 1 vez, CaixaLivre, 1, palavra, depois) )
Então
  Copia(1, palavra, antes, Grupo(Resolução/Freqüência) )
  CopiaCond(Grupo(Resolução/Freqüência))
  Copia(1, palavra, depois, Grupo(Resolução/Freqüência), AlteraPonteiro)

Se
  CP(String("Hz", CaixaEspecificada, 0001, 9999) ) and
  CV(String("X", 1 vez, CaixaLivre, 1, palavra, antes) )
Então
  Copia(1, linha, antes, Grupo(Resolução/Freqüência) )
  CopiaCond(Grupo(Resolução/Freqüência) )

Se
  CP(String("HREF", CaixaEspecificada), 0001, 9999) )
Então
   CopiaAté(caracter(" " ", 2 vezes, CaixaLivre, 1, linha, depois), Grupo(Links),
AlteraPonteiro)

Se
  (CP(String("14", CaixaLivre, 0001, 9999) ) or
  CP(String("15", CaixaLivre, 0001, 9999) ) or
  CP(String("17", CaixaLivre, 0001, 9999) ) or
  CP(String("20", CaixaLivre), 0001, 9999) ) ) and
  CV(String("in", 1 vez, CaixaLivre), 1, palavra, depois) )
Então
  CopiaTexto("Polegadas:", Grupo(Tamanho) )
  CopiaCond(Grupo(Tamanho) )

Se
  (CP(String("14", CaixaLivre, 0001, 9999) ) or
  CP(String("15", CaixaLivre, 0001, 9999) ) or
  CP(String("17", CaixaLivre, 0001, 9999) ) or
  CP(String("20", CaixaLivre, 0001, 9999) ) ) and
  CV(Caracter(" " ", 1 vez, CaixaLivre), 1, palavra, depois) )
Então
  CopiaTexto("Polegadas:", Grupo(Tamanho) )
  CopiaCond(Grupo(Tamanho) )
```

```
Se
 (CP(String("Sony Eletronics", CaixaLivre, 0001, 9999) ) or
 CP(String("Apple Computers", CaixaLivre, 0001, 9999) ) or
 CP(String("MarkVision", CaixaLivre, 0001, 9999) ) or
 CP(String("Philips", CaixaLivre, 0001, 9999) ) )
Então
 CopiaCond(Grupo(Marca) )
```

8.1.3 Arquivos Fontes Clarion

Arquivos de fontes da linguagem de programação Clarion apresentam dados de forma altamente estruturada. Praticamente todo o conteúdo de um arquivo deste tipo é formado por comandos pré-definidos na gramática da linguagem de programação. No entanto, os comentários, apesar de delimitados por comandos definidos, são de conteúdo e estruturação indeterminada, conforme a organização do autor.

8.1.3.1 Domínio dos Fontes

Os arquivos fontes processados pelo extrator são os arquivos em Clarion utilizados para a geração do próprio sistema SES. Desta forma, todos os arquivos selecionados continham o mesmo grupo de comandos, sendo que cada arquivo representava um sub-módulo do módulo extrator. Adicionalmente, cabe salientar que o conteúdo dos arquivos era em inglês, quanto aos comandos, e em português, quanto a comentários e outras cadeias de caracteres colocadas pelo autor.

O tamanho dos arquivos fontes analisados é maior que o dos arquivos descritos anteriormente, sendo que dez entre os trinta arquivos processados tinham tamanho superior a trinta e cinco quilo bytes.

8.1.3.2 Objetivo do uso dos Fontes Clarion

Os arquivos fontes utilizados nos testes realizados com o SES permitiram verificar e validar algumas das características deste sistema de extração:

- Atuação do SES junto a domínios de extração estruturados.
- Adaptação do extrator a diferentes tipos de língua de origem, no caso específico, ao inglês e ao português.
- Extração de dados em arquivos longos.
- Tempo de extração.

8.1.3.3 Dados a serem Extraídos dos Fontes

A partir dos arquivos fontes em Clarion pretende-se extrair as seguintes informações:

- Módulo a que pertencia o código fonte; no caso, Extrator ou Configurador.

- Nomes de rotinas, procedimentos e funções.
- Comentários do autor.

8.1.3.4 Base de Conhecimento Criada para os Fontes Clarion

Grupos de Assuntos:

- Módulo
- Rotinas/Proced./Funções
- Comentários

Regras de Configuração:

```
Se
  CP(String("MEMBER", CaixaEspecificada), 0001, 0003) )
Então
  Copia(1, linha, depois, Grupo(Módulo), AlteraPonteiro)

Se
  CP(String("ROUTINE", CaixaEspecificada, 0001, 0030) ) or
  CP(String("PROCEDURE", CaixaEspecificada, 0001, 0030) ) or
  CP(String("FUNCTION", CaixaEspecificada, 0001, 0030) )
Então
  CopiaCond(Grupo(Rotinas/Proced./Funções) )
  Copia(1, linha, antes, Grupo(Rotinas/Proced./Funções) )

Se
  CP(String("!", CaixaLivre, 0001, 9999) )
Então
  Copia(1, linha, depois, Grupo(Comentário) )
```

8.1.4 Grupo de Arquivos Distintos

Visando validar uma das mais importantes características do SES, gerou-se um grupo de arquivos de entrada formado por arquivos dos três tipos anteriores em proporções iguais. A geração deste grupo irá viabilizar a melhor avaliação do processo classificador que deverá identificar o tipo do arquivo de entrada e usar o conjunto de regras de extração correspondente a este tipo.

8.1.4.1 Domínio dos Grupos de Arquivos

O grupo de arquivos gerado para teste apresenta três domínios distintos, visto que contém os domínios apresentados nos itens anteriores: *mails*, HTMLs e fontes Clarion.

8.1.4.2 Objetivos do uso do Grupo de Arquivos

Os arquivos do grupo utilizados nos testes realizados com o SES permitiram verificar e validar algumas das características deste sistema de extração:

- Verificar o processo de classificação
- Avaliar o SES quanto a extração de informações de domínios distintos.
- Extrair dados de arquivos em diferentes línguas de origem.
- Medir Tempo de Extração.

8.1.4.3 Dados a serem Extraídos do Grupo de Arquivos

A extração dos dados é realizada conforme cada um dos diferentes domínios existentes dentro do grupo de arquivos, ou seja, o Extrator adapta-se ao tipo do arquivo. Neste sentido, as bases de conhecimento a serem usadas com cada tipo de arquivo são as definidas anteriormente.

8.1.4.4 Base de Conhecimento para o Grupo de Arquivos

No caso do grupo de arquivos o Extrator utiliza mais de uma base de conhecimento durante o processo de extração. Estas bases são as três já descritas nos itens anteriores. A seleção de qual base usar é realizada conforme a extensão do arquivo de entrada.

8.2 Ambiente de Validação

A fim de permitir uma melhor avaliação do processo de validação apresentado neste capítulo, é importante caracterizar o sistema operacional e o computador em que tais testes se realizaram.

- Sistema Operacional: Windows 95.
- Número de processos estranhos ao executável Extrator e ao sistema operacional: 0 (o processo extrator executava sozinho no computador).
- Processador: Pentium 133 Mhz
- Memória RAM: 32 Mb

8.3 Resultados dos Testes

Utilizando os quatro conjuntos de arquivos descritos anteriormente, realizou-se um conjunto de testes sobre o Sistema de Extração Semântica de Informações (SES). Tais resultados foram tabulados, gerando as tabelas a seguir apresentadas.

A TABELA 8.1 mostra o volume de dados processado (em bytes) durante os testes realizados. Como pode-se observar, os grupos de arquivos de entrada foram sendo testados gradativamente. Efetuou-se inicialmente extrações sobre somente cinco arquivos de entrada de cada tipo e, a seguir, novos testes foram sendo realizados até

atingir-se trinta arquivos de cada tipo, acrescentando-se cinco arquivos em cada novo teste. Este processo permitiu estudar algumas características importantes do processo de extração, como taxa de crescimento do tempo de extração conforme o aumento do volume dos dados de entrada (FIGURA 8.1).

A TABELA 8.2 apresenta os tempos de extração, em segundos, para cada um dos testes realizados, conforme os respectivos volumes de dados da TABELA 8.1.

A partir das duas tabelas iniciais, gerou-se uma terceira, que informa a velocidade de extração das informações em bytes por segundo (TABELA 8.3). Novamente, a realização dos testes em vários passos, incrementando-se o número de arquivos a cada passo, permitiu a geração do gráfico da FIGURA 8.2. Este gráfico apresenta as curvas de velocidade de extração conforme os diferentes tipos de arquivos de entrada.

Adicionalmente, a TABELA 8.4 apresenta a velocidade de extração por regra, conforme as bases de conhecimento definidas para cada tipo de arquivo de entrada testado.

Visando verificar a qualidade dos resultados obtidos a partir do processo de Extração Semântica de Informações proposto e implementado neste trabalho, realizaram-se algumas medições de *recall* e *precision*. Tais medidas foram geradas a partir da comparação visual (análise de conteúdo) entre os trinta arquivos de entrada dos diferentes tipos utilizados (*mails*, HTMLs e fontes) e os respectivos arquivos de saída. Durante a análise dos arquivos de entrada foram salientadas quais informações deveriam ser extraídas pelo sistema SES, e, após a execução do processo de extração, observou-se quais estavam presentes nos arquivos de saída, bem como quais informações sem relevância foram extraídas.

Através da comparação realizada, pôde-se verificar quais eram as informações contidas nos arquivos de entrada e quais eram as informações contidas nos arquivos de saída. O índice de *recall* foi atribuído observando-se a quantidade de informações relevantes recuperadas a partir dos arquivos de entrada; o índice de *precision*, verificando-se a precisão a informações extraídas; ou seja, entre as informações recuperadas, quais eram relevantes.

No entanto, é importante destacar que tais medidas devem ser aprofundadas em trabalhos futuros através de metodologias de validação mais profundas e menos subjetivas, a fim de obter resultados mais precisos. Contudo, as medidas obtidas neste teste inicial do sistema SES, permitiram um primeira visão das perspectivas abertas por este trabalho.

Os índices de *recall* e *precision* obtidos foram muito satisfatórios, tendo-se sobre arquivos de *mail* um índice de 80 % de *recall* e um índice de 60 % de *precision*. No caso de HTMLs, os valores obtidos para *recall* e *precision* foram respectivamente 90 % e 70 %. Finalmente, sobre arquivos fontes Clarion, os índices de *recall* e *precision* foram de 100%.

Em anexo a esta dissertação, são colocados alguns exemplos de arquivos de entrada e seus respectivos arquivos de saída para cada um dos tipos de arquivo utilizados (*mails*, HTMLs e fontes).

A fim de testar o sistema SES sobre um volume de dados maior, foi efetuada uma extração de dados juntando os trinta arquivos de cada tipo. O resultado é mostrado na TABELA 8.5.

TABELA 8.1 - Tabela de Volume de Dados Processados

Volume (bytes)	5 Arqs.	10 Arqs.	15 Arqs.	20 Arqs.	25 Arqs.	30 Arqs.
Mails	50.138	100.396	129.700	177.517	221.704	294.316
HTML	47.534	70.488	131.093	155.242	179.249	216.879
Fontes	43.942	51.791	93.760	102.777	116.024	133.183
Grupo	55.627	83.701	156.229	194.863	226.199	242.088
Média	49.310	76.594	127.695	157.600	185.794	221.616

TABELA 8.2 - Tabela de Tempos de Processamento

Tempo (seg.)	5 Arqs.	10 Arqs.	15 Arqs.	20 Arqs.	25 Arqs.	30 Arqs.
Mails	53	124	163	222	312	367
HTML	17	27	51	60	72	85
Fontes	9	11	22	25	27	31
Grupo	38	54	92	119	130	148
Média	29	54	82	106	135	158

TABELA 8.3 - Tabela de Razão de Volume por Tempo (velocidade)

bytes/seg	5 Arqs.	10 Arqs.	15 Arqs.	20 Arqs.	25 Arqs.	30 Arqs.
Mails	946	810	796	800	710	802
HTML	2.796	2.611	2.570	2.587	2.489	2.551
Fontes	4.882	4.708	4.262	4.111	4.297	4.296
Grupo	1.463	1.550	1.698	1.637	1.740	1.636
Média	1.700	1.418	1.557	1.487	1.376	1.403

TABELA 8.4 - Tabela de Velocidade por Regra

bytes/seg/ regra	5 Arqs.	10 Arqs.	15 Arqs.	20 Arqs.	25 Arqs.	30 Arqs.
Mails 9 regras	105	90	88	89	79	89
HTML 10 regras	280	261	257	259	249	255
Fontes 3 regras	1.627	1.569	1.421	1.370	1.432	1.432

TABELA 8.5 - Resultados para Noventa Arquivos

90 Arqs.	Volume (bytes)	Tempo (segs.)	Velocidade (bytes/seg.)
Grupo	644.378	486	1.326

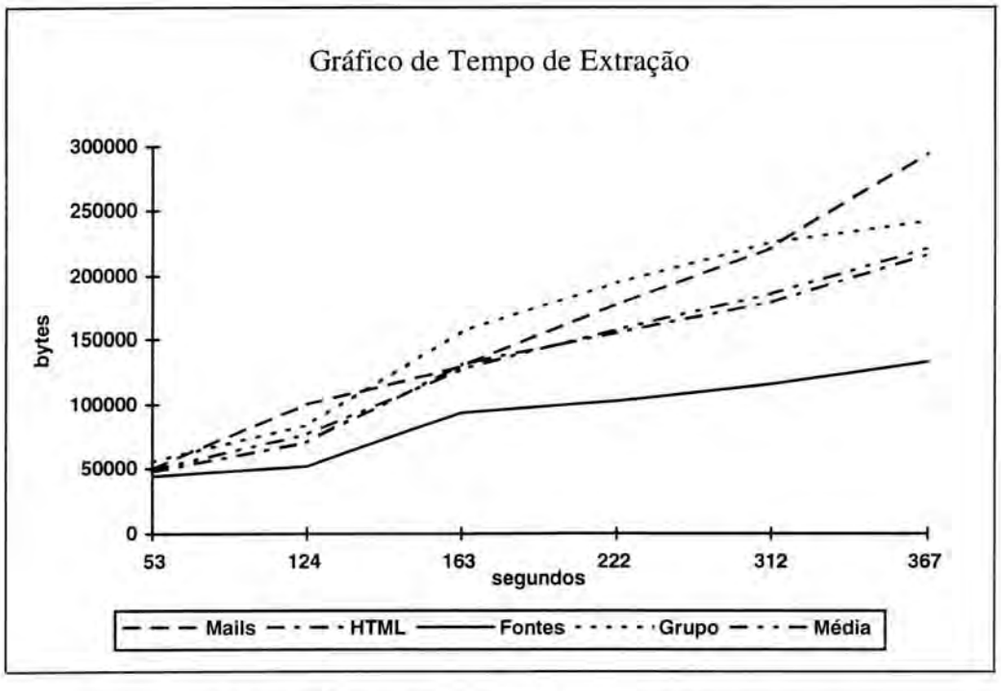

FIGURA 8.1 - Gráfico de Tempo de Extração

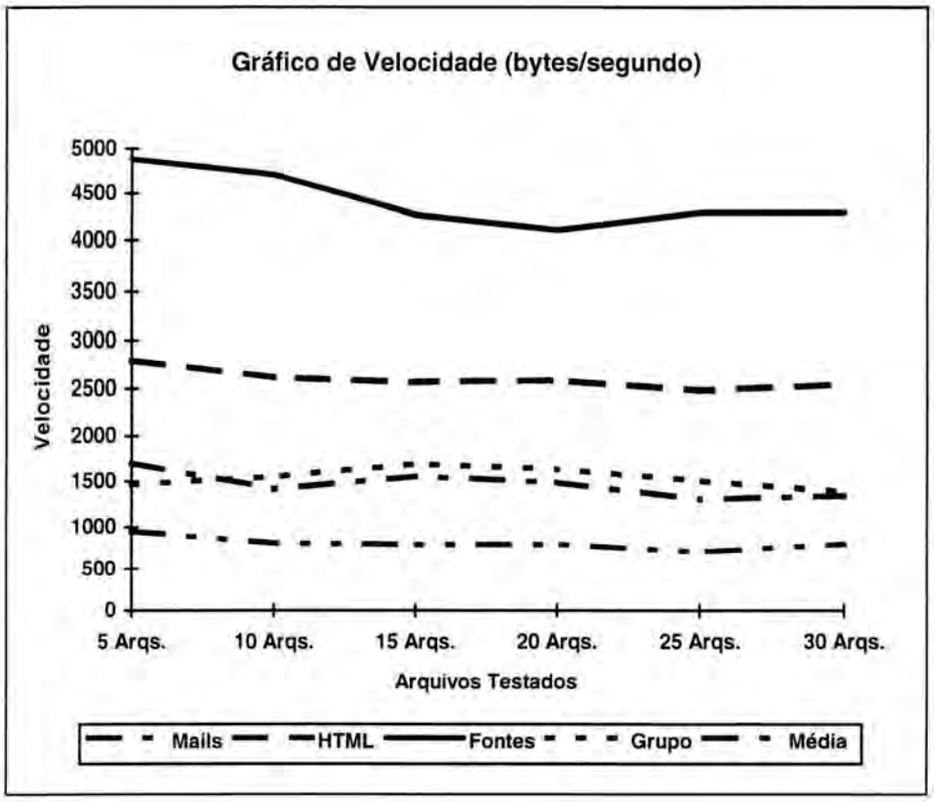

FIGURA 8.2 - Gráfico de Velocidade (bytes/segundo)

8.4 Considerações sobre o Processo de Validação

A partir dos dados resultantes dos testes realizados sobre o SES e apresentados no item anterior, pode-se observar um conjunto de conclusões e considerações:

- Conforme o gráfico da FIGURA 8.1, pode-se observar que o tempo de processamento tende a crescer linearmente em relação ao volume de dados de entrada (média).

- A velocidade de extração, independentemente do tipo de arquivo de entrada, é constante, sendo também esta independente do volume de dados de entrada.

- Conforme o gráfico da FIGURA 8.2, observa-se que a velocidade de extração aumentou conforme o grau de estruturação do arquivo de entrada, sendo os *mails* os arquivos cuja velocidade de extração é mais baixa, e os arquivos fontes em Clarion com a maior velocidade.

- É importante salientar que o número de regras da base de conhecimento, usadas para a extração das informações, influencia muito na velocidade de processamento da extração. Arquivos menos estruturados tendem a necessitar de um conjunto maior e mais complexo de regras para a extração das informações. Adicionalmente, deve-se salientar a localização da informação desejada pelo usuário do sistema dentro do arquivo de entrada, bem como a

quantidade de dados a ser extraído. Informações de maior dificuldade de extração, ou seja, com difícil caracterização para a pesquisa dentro do arquivo de entrada, exigem um maior volume de regras para a extração. Este fato fica claro através das bases de conhecimento utilizadas para os testes e pela TABELA 8.4.

- A qualidade das informações extraídas foi avaliada através de medidas de *recall* e *precision*. Estas medidas permitem validar o processo de extração através de uma comparação entre os arquivos de entrada e os resultados contidos no arquivos de saída. Conforme os dados apresentados, o SES atingiu altos graus de *recall* e *precision*, descartando as informações sem interesse para o usuário, e destacando os aspectos centrais dos arquivos de entrada, o que viabiliza o entendimento das informações de forma ágil e fácil. No entanto, é importante salientar que o processo semântico de extração depende muito da base de conhecimento envolvida na configuração do Extrator, pois a seleção e filtragem dos dados é realizada conforme o grau de precisão e qualidade desta. Adicionalmente, cabe lembrar que em [COW 96] defini-se como um bom índice de *recall* 50% e de *precision* 40%, índices superados pelos resultados dos testes aqui descritos.

- O processo de validação comprovou que o sistema SES foi capaz de manipular volumes de dados considerados grandes a nível local (TABELA 8.5), visando diminuir o tempo gasto na absorção da informação armazenada, bem como no seu entendimento.

- O SES apresentou bastante flexibilidade quanto à extração de informações a partir de arquivos de entrada distintos, variando a estrutura interna de armazenamento dos dados. Isto permitirá também sua adaptação ao surgimento de novas tecnologias e alterações nos formatos de armazenamento. Desta forma, o sistema mostrou-se flexível a tais mudanças, permitindo sua utilização nas mais diversas situações.

- Quanto ao tratamento de informações temporais, as funções de extração de dados temporais implementadas no SES e utilizadas nos testes de validação mostraram-se eficientes.

- Os resultados obtidos a partir do processo de validação foram armazenados em um arquivo texto de forma estruturada. Esta forma de arquivo de saída permitirá sua leitura e manipulação diretas pelo usuário do SES, como proposto nos objetivos desta dissertação. Além disso, o armazenamento de forma estruturada facilitará a exportação dos dados extraídos para outros sistemas.

9 Conclusão

Como capítulo final deste trabalho, apresenta-se um conjunto de considerações finais sobre o mesmo. Contudo, apesar de ser um capítulo de conclusão, seu principal objetivo é abrir novos caminhos para o aprimoramento da pesquisa e do estudo até aqui realizados.

A dissertação aqui concluída descreve um sistema para extração de dados em arquivos pessoais obtidos a partir da rede Internet, utilizando o conceito de IE (*Information Extraction*) com ênfase na análise do tempo de validade dos mesmos. Os dados processados são parcialmente estruturados ou não estruturados sendo manipulados por um extrator configurado a partir de bases de conhecimento geradas pelo usuário do sistema.

O desenvolvimento deste projeto foi realizado com o uso da ferramenta *Clarion for Windows 1.5*. A utilização desta ferramenta permitiu o desenvolvimento para ambiente Windows 95, usando todas as facilidades de interface oferecidas por este sistema operacional. O resultado final deste estudo é o referencial conceitual e a implementação de um sistema para facilitar a utilização e manipulação de dados locais através de um processo baseado na extração semântica de dados.

9.1 Considerações Finais

Considerando a proposta inicial do presente trabalho, realizar o embasamento conceitual e a descrição de um sistema de extração de informações em bases de dados não estruturadas ou semi-estruturadas, capaz de auxiliar os usuários na manipulação de dados locais, e as restrições anteriormente discutidas, principalmente nos capítulos 2 e 4, obtivemos como produto final, o protótipo apresentado neste documento. Este protótipo deverá constituir-se de um instrumento bastante útil para aquelas pessoas imbuídas da tarefa de manipular um volume de informações elevado para tratamento a nível local, bem como informações de diferentes formatos.

Acreditamos que os resultados deste e de outros trabalhos que venham a ser realizados nesta área, irão beneficiar principalmente aquele conjunto de pessoas que sentem dificuldade em utilizar os sistemas atuais de manipulação de dados locais, os quais não comportam nenhum suporte automático de tratamento da informação. Tais pessoas, por esta razão, acabam prejudicadas na absorção da vasta quantidade de conhecimento hoje disponível a todos indivíduos.

Cabe salientar que o protótipo descrito não constitui-se, por assim dizer, em uma linha de chegada, mas sim, em um ponto de partida do qual muitos outros trabalhos ou estudos poderão ser desenvolvidos. Neste aspecto, o que nos parece o mais importante a ser destacado diz respeito ao fato de que este conjunto de projetos de pesquisa potenciais não está restrito a uma única área da Ciência da Computação, envolvendo, por exemplo, pesquisadores ligados às áreas de Recuperação de Informações,

Inteligência Artificial, Bancos de Dados Temporais, Descoberta de Conhecimento e Extração de Informações.

Por outro lado, um aspecto interessante observado no decorrer deste estudo é a inexistência de um sistema similar ao projetado, no tocante à generalidade de aplicações do mesmo. O SES não se restringe à extração de informações de um determinado assunto, língua, estrutura de armazenamento ou tipo de dados a ser extraído, podendo ser totalmente configurado, conforme as necessidades do usuário. A adoção de um sistema baseado em conhecimento para a extração de informações armazenadas em bases de dados não estruturadas ou semi-estruturadas parece ser uma boa opção para satisfazer plenamente as necessidades e restrições impostas no projeto.

Gostaríamos de enfatizar ainda as vantagens advindas do fato de ter-se particionado o sistema proposto em vários processos individuais, o que irá facilitar os trabalhos futuros referentes a programação e teste de cada um dos módulos.

No entanto, devido à complexidade do processo de extração, foi detectada uma dificuldade quanto à provável velocidade de execução deste processo. Este fato exigiu o uso de técnicas de programação que visassem a otimização da velocidade de processamento, bem como a utilização de uma linguagem de programação que apresenta flexibilidade e velocidade de execução, aproveitando todo o potencial provido pelo *hardware*.

A necessidade e possibilidade de aprofundar os estudos realizados, melhorando, desta forma, os resultados, até agora obtidos, fez com que surgisse o desejo de implementar-se o protótipo descrito neste documento, a fim de servir como alvo de conclusões mais especificas, conforme as apresentadas no capítulo de validação. No entanto, espera-se gerar novas avaliações do protótipo, mediante sua contínua utilização em ambientes práticos, bem como, realizar o desenvolvimento de novas versões, visando a correção de falhas e, principalmente, a implementação de melhorias.

9.2 Trabalhos Futuros

Um dos mais importantes parâmetros que pode ser utilizado para a avaliação qualitativa de um trabalho de pesquisa, seja ele realizado em nível de graduação, mestrado, doutorado, ou até mesmo na vida profissional, tendo como referência a razão central que motivou o seu desenvolvimento, são as perspectivas, caminhos e idéias por ele inspiradas, tanto em quem o realizou efetivamente, quanto naqueles que por ventura tomaram contato com o produto final (trabalho escrito e/ou programa implementado).

Neste sentido, tendo em vista os objetivos necessariamente limitados que restringiram a abrangência do presente trabalho, a necessidade e a possibilidade de aprofundar os estudos realizados, melhorando, desta forma, os resultados até agora obtidos e, finalmente, a eventual extensão ou deslocamento do foco de atenção deste trabalho para outras áreas correlatas, cria-se um estímulo a criatividade deste e de outros pesquisadores, ensejando-lhes a concepção de projetos que podem ser propostos como trabalhos a serem desenvolvidos futuramente, tanto em nível de graduação, quanto de pós-graduação.

Neste particular, merecem destaque as seguintes propostas de trabalhos de pesquisa, os quais visam complementar ou aproveitar os resultados obtidos e os conhecimentos adquiridos durante o período de preparação desta dissertação:

- **Sistema de Descoberta de Conhecimento a Partir dos Dados Extraídos**

Como descrito no capítulo de introdução, o estudo realizado nesta dissertação faz parte de um projeto maior dividido em duas fases. Neste sentido, as informações extraídas pelo SES serão analisadas por um sistema de descoberta de conhecimentos em bases de dados temporais para a manipulação automática de arquivos de dados locais (segunda fase).

Assim sendo, o estudo até aqui realizado abre caminho para novas áreas de pesquisa, que poderão beneficiar-se do que já foi feito para o futuro desenvolvimento de um Sistema de Descoberta de Conhecimento.

- **Novas Regras de Extração**

Nesta dissertação, foi estudado e implementado um conjunto de regras de extração de informações. Considerando importante a continuidade deste trabalho e a constante aproximação do mesmo à realidade dos usuários, os quais exigem muitas vezes a extração de informações complexas, observa-se a necessidade da expansão do grupo de regras do SES. Esta expansão visa ampliar o potencial de extração semântica e a aplicabilidade do sistema em ambientes reais.

- **Atualização Automática da Base de Conhecimento**

O estudo de técnicas e metodologias orientadas à aquisição automática do conhecimento, tornando possível a construção pelo próprio sistema de novas regras de extração, pode ser um ponto de ampliação e melhoria do sistema de extração semântica de informações apresentado. O objetivo é melhorar a qualidade dos resultados obtidos a partir do SES através de regras incluídas automaticamente na base de conhecimento inicialmente configurada pelo usuário do sistema, bem como a alteração ou exclusão de outras regras já existentes.

- **Formatação dos Dados Extraídos**

O estudo desenvolvido nesta dissertação enfocou de forma mais profunda as técnicas envolvidas na extração de informações. Contudo, por limitações de tempo, o arquivo de saída resultante do processo de extração não apresentou uma formatação conforme as definições apresentadas nas seis Conferência de Compreensão de Mensagens (*Message Understanding Conferences* - MUCs) [MUC 96].

Esta padronização do arquivo de saída conforme definido no MUC é de importante valia para o SES, pois permitirá a exportação dos dados resultantes do processo de extração para vários outros sistemas de forma facilitada, ampliando o uso do estudo aqui desenvolvido.

- **Pré Processamento de Arquivos de Entrada**

Apesar de fazer parte do embasamento conceitual desta dissertação, o pré-processamento dos arquivos de entrada não foi implementado no protótipo do sistema SES por motivos de prazo de tempo. Contudo, como um trabalho futuro, seria válida a implementação deste sub-módulo do Extrator, que ampliará a qualidade dos dados extraídos, bem como o uso do sistema para novos tipos de arquivos de entrada, como, por exemplo, arquivos compactados.

- **Novos Testes**

Como apresentado no capítulo 8, o sistema SES foi testado sobre um conjunto de noventa arquivos de entrada. No entanto, os testes devem continuar, visando a correção de novas possíveis falhas, bem como aprofundar a própria validação do estudo apresentado nesta dissertação. Neste sentido, novos testes permitiram realizar um estudo quantitativo e qualitativo das informações extraídas.

Anexo 1 Arquivos de Entrada e Saída

Durante o processo de validação foram utilizados vários arquivos de entrada para testes do sistema SES. A fim de melhor descrever os resultados obtidos no uso do sistema proposto, a seguir são apresentados alguns dos arquivos de entrada, com seus respectivos arquivos de saída. Tais arquivos estão aqui transcritos fielmente aos originais.

- **Arquivo *Mail* de Entrada**

Return-Path: <owner-dbworld@cs.wisc.edu>
Received: from sea.cs.wisc.edu by castor.inf.ufrgs.br (4.1/SMI-4.1)
id AA03388; Tue, 29 Oct 96 21:12:26 EDT
Received: from localhost (localhost [127.0.0.1]) by sea.cs.wisc.edu (8.6.12/8.6.12) with SMTP id QAA01460; Tue, 29 Oct 1996 16:13:22 -0600
Received: from ricotta.cs.wisc.edu (ricotta.cs.wisc.edu [128.105.67.19]) by sea.cs.wisc.edu (8.6.12/8.6.12) with ESMTP id QAA01205 for <dbworld@sea.cs.wisc.edu>; Tue, 29 Oct 1996 16:03:52 -0600
Received: (from raghu@localhost) by ricotta.cs.wisc.edu (8.6.12/8.6.12) id QAA23724 for dbworld; Tue, 29 Oct 1996 16:03:51 -0600
Message-Id: <199610292203.QAA23724@ricotta.cs.wisc.edu>
Date: Tue, 29 Oct 1996 16:03:51 -0600
Reply-To: desilva@mech.ubc.ca
Sender: owner-dbworld@cs.wisc.edu
Precedence: bulk
From: Dr. C. W. de Silva <desilva@mech.ubc.ca>
To: Multiple recipients of list <dbworld@cs.wisc.edu>
Subject: (DBWORLD) Call For Papers
X-To: dbworld@ricotta.cs.wisc.edu
X-Listprocessor-Version: 7.1 -- ListProcessor by CREN

Dear Colleague,
Please consider the following call for papers for a conference
and for an associated special issue of a journal.
Thank you.
Clarence de Silva.
(Regional Editor, North America, for the EAAI Journal)
==========

Special Journal Issue on Intelligent Electronic Systems
==========

International Conference on Conventional and Knowledge-Based Intelligent Electronic Systems will be held from 21-23 May, 1997, in Adelaide, Australia (Conference Chair: L.C. Jain). Several high-quality papers

chosen from those accepted for the Conference will be published in a special issue of the International Journal, Engineering Applications of Artificial Intelligence (The International Journal of Intelligent Real-Time Automation). For the special issue, only papers on "Intelligent Electronic Systems" will be considered. For example, fuzzy, neural, rule-based, knowledge-based, and expert control of practical systems in areas such as biomedical engineering, consumer electronics, communication, control, production systems, electronic security, industrial electronics, mechatronics, multimedia, optical electronics, sensor and actuator technology, robotics, and virtual reality would be appropriate. Please submit your complete paper, to be simultaneously considered for both Conference and Journal issue, no later than November 15, 1996 to:

Clarence W. de Silva
NSERC Professor of Industrial Automation
Department of Mechanical Engineering
The University of British Columbia
2324 Main Mall
Vancouver, BC, CANADA V6T 1Z4
Fax: 604-822-2403
e-mail: desilva@mech.ubc.ca

==========

The dbworld alias reaches many people, and should only be used for messages of general interest to the database community.

Requests to get on or off dbworld should go to listproc@cs.wisc.edu.

to subscribe send
subscribe dbworld Your Full Name

to unsubscribe send
unsubscribe dbworld

to change your address
send an unsubscribe request from the old address
send a subscribe request from the new address

to find out more options send
help

---FOOTER-

- **Arquivo de Saída para *Mails***

```
"MAIL01.EML","Data Envio"," Tue, 29 Oct 1996"
"MAIL01.EML","Remetente"," Dr. C. W. de Silva <desilva@mech.ubc.ca>"
"MAIL01.EML","Assunto"," (DBWORLD) Call For Papers"
"MAIL01.EML","Titulo","International"
"MAIL01.EML","Titulo","Conference"
"MAIL01.EML","Titulo","on Conventional and Knowledge-Based Intelligent Electronic
Systems will be held from 21-23 May, 1997, in Adelaide,"
"MAIL01.EML","Prazos/Datas Importantes","Prazo Final:"
"MAIL01.EML","Prazos/Datas Importantes","November 15, 1996"
```

- **Arquivo HTML de Entrada**

```
<!-- Last modified on Fri September 27,1996 -->
<!-- by MPO of ID -->
<HTML>
<HEAD>
<TITLE>Multiscan Trinitron True-Color Display</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff">
<a href="/SEL/ccpg/ITA2head.map"><IMG border=0 SRC="/SEL/ccpg/ITA2head.gif"
ismap></a><br>
<a href="mstch.map"><IMG border=0 SRC="mstch.gif" ismap></a>
<H2>Multiscan<font size=-2>(R)</font>

 TC Trinitron<font size=-2>(R)</font> True-Color Display</H2>

<IMG ALIGN=LEFT HSPACE=10 BORDER=3 SRC="mstc.gif" WIDTH=200
HEIGHT=210>

Ideal color reference display for <BR>
color-critical, pre-press, and electronic <BR>
publishing applications.

<BR CLEAR=ALL><BR><FONT SIZE=+2>$3,999.99 </FONT> ESP*<BR>
<FONT SIZE=-1>*Price listed is an estimated U.S. retail selling price.
  Actual retail price may vary.</FONT>
<HR WIDTH=471PIXELS ALIGN=LEFT SIZE=3 NOSHADE>
<H3>Features and Benefits</H3>
<UL>
        <LI>20" Trinitron CRT (19.1" viewable image size) for superior image quality
        <LI>Digital Multiscan Technology to support multiple PC and Macintosh<font
size=-2>(R)</font>
         <BR>resolutions
        <LI>Digital Uniformity Compensation for color uniformity throughout the
screen
        <LI>Beam Current Feedback maintains stable color reproduction over time
```

```
        <LI>Supports resolutions up to 1600 x 1200 non-interlaced
        <LI>On-screen display of comprehensive digital controls for easy picture
adjustment
        <LI>Meets or exceeds industry standards - EPA, MPRII, TCO
</UL>

<H3>Specifications</H3>

<UL>
        <LI>19.1" Maximum Viewable Image Size
        <LI>0.30 mm Aperture Grill Pitch
        <LI>31-85 kHz Horizontal Scan Range, 50-150 Hz Vertical Scan Range
        <LI>Maximum Resolutions
<UL>  PC: 1600 x 1200 @ 60 Hz<BR>
                MAC: 1280 x 1024 @ 75 Hz
</UL>
        <LI>Color Temperature 3000 - 9300 Kelvin (User Adjustable by 100f steps)
        <LI>On Screen Display Controls
<UL>  Brightness<BR>
                Contrast<BR>
                Horizontal Size and Centering<BR>
                Vertical Size and Centering<BR>
                Horizontal and Vertical Static Convergence<BR>
                Color Temperature<BR>
                Geometry<BR>
                BCF (Beam Current Feedback)<BR>
                DUC (Digital Uniformity Compensation)
</UL></UL>

<P ALIGN=CENTER><A HREF="dcontent.html">Computer Graphic Displays</A> ||
<A HREF="../storage/scontent.html">Data Storage &amp;amp; CD-ROM</A> ||
<A HREF="../cav/ccontent.html">Computer Audio &amp;amp; Video</A></P>
<P><HR WIDTH=471PIXELS ALIGN=LEFT SIZE=3 NOSHADE><BR>
<A HREF="dnav.map"><IMG border=0 SRC="dnav.gif" ismap></a><P><P>
<A HREF="http://www.sony.com">Sony Online</A> ||
<A HREF="http://www.SEL.sony.com">Sony Electronics</A><P>
<A HREF="/SEL/consumer/ss5/trademarks.html">&#169 1996 Sony Electronics Inc.
All rights reserved.</A>
</BODY>
</HTML>
```

- **Arquivo de Saída para HTML**

```
"HTML01.HTM","Data Atualizacao","Last modified"
"HTML01.HTM","Data Atualizacao","on Fri September 27,1996 -->"
"HTML01.HTM","Titulo","Multiscan Trinitron True-Color Display"
"HTML01.HTM","Titulo",""
"HTML01.HTM","Links",""""/SEL/ccpg/ITA2head.map"
```

"HTML01.HTM","Links",""""mstch.map"
"HTML01.HTM","Tamanho","Polegadas:"
"HTML01.HTM","Tamanho","20"
"HTML01.HTM","Resolucao/Frequencia"," <LI>Supports resolutions up to 1600"
"HTML01.HTM","Resolucao/Frequencia","x"
"HTML01.HTM","Resolucao/Frequencia","1200 non-interlaced"
"HTML01.HTM","Resolucao/Frequencia","85"
"HTML01.HTM","Resolucao/Frequencia","kHz"
"HTML01.HTM","Resolucao/Frequencia","Horizontal"
"HTML01.HTM","Resolucao/Frequencia","150"
"HTML01.HTM","Resolucao/Frequencia","Hz"
"HTML01.HTM","Resolucao/Frequencia","Vertical"
"HTML01.HTM","Resolucao/Frequencia","<UL> PC: 1600"
"HTML01.HTM","Resolucao/Frequencia","x"
"HTML01.HTM","Resolucao/Frequencia","1200 @ 60 Hz
"
"HTML01.HTM","Resolucao/Frequencia"," MAC: 1280 x 1024 @ 75"
"HTML01.HTM","Resolucao/Frequencia","Hz"
"HTML01.HTM","Links",""""dcontent.html"
"HTML01.HTM","Links",""""../storage/scontent.html"
"HTML01.HTM","Links",""""../cav/ccontent.html"
"HTML01.HTM","Links",""""dnav.map"
"HTML01.HTM","Links",""""http://www.sony.com"
"HTML01.HTM","Links",""""http://www.SEL.sony.com"
"HTML01.HTM","Links",""""/SEL/consumer/ss5/trademarks.html"
"HTML01.HTM","Marca","Sony Electronics Inc"

- **Arquivos em Fontes de Clarion**

```
                MEMBER('EXTRATOR.clw')      ! This is a MEMBER module
Main PROCEDURE

LocalRequest        LONG,AUTO
OriginalRequest     LONG,AUTO
LocalResponse       LONG,AUTO
WindowOpened        LONG
WindowInitialized   LONG
ForceRefresh        LONG,AUTO
CurrentTab          STRING(80)
DisplayDayString STRING('Domingo Segunda Terça  Quarta  Quinta  Sexta
Sábado  ')
DisplayDayText   STRING(9),DIM(7),OVER(DisplayDayString)
DisplayMonthString STRING('Janeiro FevereiroMarço  Abril  Maio  Junho  Julho
Agosto  Setembro Outubro  Novembro Dezembro ')
DisplayMonthText   STRING(9),DIM(12),OVER(DisplayMonthString)
AppFrame            APPLICATION('SES - Sistema de Extração Semântica de
Informações (Extrator)'),AT(,,317,183),CENTER,ICON('EXTRATOR.ICO'),STATUS(-
1,80,120,45),SYSTEM,MAX,RESIZE,IMM
                  MENUBAR
```

```
                MENU('&Arquivo'),USE(?FileMenu)
                 ITEM('&Inicialização'),USE(?FileInicialização),MSG('Inicializa
extrator.')
                 ITEM('&Configurador'),USE(?ArquivoConfigurador),MSG('Executa
módulo configurador.')
                 ITEM,SEPARATOR
                 ITEM('&Ajuste Impressora'),USE(?PrintSetup),MSG('Setup
Printer'),STD(STD:PrintSetup)
                 ITEM,SEPARATOR
                 ITEM('&Saida'),USE(?Exit),MSG('Exit this
application'),STD(STD:Close)
                END
                MENU('&Editar'),USE(?EditMenu)
                 ITEM('&Recortar'),USE(?Cut),MSG('Remove item para a área de
transferência.'),STD(STD:Cut)
                 ITEM('&Copiar'),USE(?Copy),MSG('Copia item para a área de
transferência.'),STD(STD:Copy)
                 ITEM('C&olar'),USE(?Paste),MSG('Cola o conteúdo da área de
transferência.'),STD(STD:Paste)
                END
                ITEM('&Extrair'),USE(?Extrair),MSG('Extrai informações dos arquivos
de entrada.')
                MENU('&Janelas'),USE(?Janelas),MSG('Ajusta
janelas.'),STD(STD:WindowList)
                 ITEM('&Lado-a-Lado'),USE(?Tile),MSG('Coloca todas as janelas de
forma visível.'),STD(STD:TileWindow)
                 ITEM('&Cascata'),USE(?Cascade),MSG('Empilha todas as
janelas.'),STD(STD:CascadeWindow)
                 ITEM('&Ajusta Ïcones'),USE(?Arrange),MSG('Alinha todos os
ícones.'),STD(STD:ArrangeIcons)
                END
                ITEM('Arq. de &Saída'),USE(?ArqdeSaída),MSG('Visualizar arquivo de
saída.')
                MENU('&Ajuda'),USE(?Ajuda),MSG('Janelas de ajuda.')
                 ITEM('&Contents'),USE(?Helpindex),MSG('View the contents of the
help file'),STD(STD:HelpIndex)
                 ITEM('&Search for Help On...'),USE(?HelpSearch),MSG('Search for
help on a subject'),STD(STD:HelpSearch)
                 ITEM('&How to Use Help'),USE(?HelpOnHelp),MSG('How to use
Windows Help'),STD(STD:HelpOnHelp)
                 ITEM,SEPARATOR
                 ITEM('Sobre'),USE(?AjudaSobre),MSG('Informações sobre o
SISDECON.')
                END
               END
               TOOLBAR,AT(0,0,317,16)
                BUTTON,AT(20,2,13,13),MSG('Módulo configurador'),TIP('Módulo
configurador'),USE(?Button1),ICON('CONFIGUR.ICO')
```

```
BUTTON,AT(52,2,13,13),MSG('Extração'),TIP('Extração'),USE(?Button2),ICON('BUS
CA.ICO')
                LINE,AT(1,1,318,0),USE(?Line1),COLOR(00H)
                BUTTON,AT(70,2,13,13),MSG('Visualiza arquivo de saída.'),TIP('Arq.
de Saída'),USE(?Button5),ICON('ROLOCD2.ICO')

BUTTON,AT(2,2,13,13),MSG('Sair'),TIP('Sair'),USE(?Button4),ICON('SAIR.ICO'),ST
D(STD:Close)

BUTTON,AT(34,2,13,13),MSG('Inicialização'),TIP('Inicialização'),USE(?Button3),ICO
N('INIC01.ICO')
              END
            END
 CODE
 PUSHBIND
 LocalRequest = GlobalRequest
 OriginalRequest = GlobalRequest
 LocalResponse = RequestCancelled
 ForceRefresh = False
 CLEAR(GlobalRequest)
 CLEAR(GlobalResponse)
  GLO:DiretorioEntrada = getini('Inicialização','Diretorio Entrada',,'Extrator.INI')
  GLO:DiretorioRegras  = getini('Inicialização','Diretorio Regras',,'Extrator.INI')
  GLO:Extensoes        = getini('Inicialização','Base de Extensoes',,'Extrator.INI')
  GLO:NomeMes          = getini('Inicialização','Base de Meses',,'Extrator.INI')
  GLO:NomeDia          = getini('Inicialização','Base de Dias',,'Extrator.INI')
  GLO:Passo_a_Passo    = getini('Inicialização','Passo-a-Passo',,'Extrator.INI')

  if GLO:Extensoes <> '' and GLO:NomeMes <> '' and GLO:NomeDia <> ''
   enable(?Extrair)
   unhide(?Button2)
  else
   disable(?Extrair)
   hide(?Button2)
  end

  GLO:ThreadMain = thread()

  GLO:ArquivoSaida = clip(GLO:DiretorioEntrada) & 'ArqSaida.Ext'
  GLO:ArquivoLog   = clip(GLO:DiretorioEntrada) & 'ArqLog.Ext'
 IF KEYCODE() = MouseRight
  SETKEYCODE(0)
 END
 OPEN(AppFrame)
 WindowOpened=True
 INIRestoreWindow('Main','Extrator.INI')
 IF NOT INRANGE(AppFrame{Prop:Timer},1,100)
  AppFrame{Prop:Timer} = 100
```

```
 END
   AppFrame{Prop:StatusText,3} = CLIP(DisplayDayText[(TODAY()%7)+1]) & ', ' &
DAY(TODAY()) & ' de ' & CLIP(DisplayMonthText[MONTH(TODAY())]) & ' de ' &
YEAR(TODAY())
   AppFrame{Prop:StatusText,4} = FORMAT(CLOCK(),@T3)
 ACCEPT
    if (GLO:Extensoes <> '') and (GLO:NomeMes <> '') and (GLO:NomeDia <> '')
     if ?Extrair{prop:disable} = 1 then enable(?Extrair).
     if ?Button2{prop:hide} = 1 then unhide(?Button2).
    else
     if ?Extrair{prop:disable} = 0 then disable(?Extrair).
     if ?Button2{prop:hide} = 0 then hide(?Button2).
    end
   CASE EVENT()
   OF EVENT:OpenWindow
    IF NOT WindowInitialized
     DO InitializeWindow
     WindowInitialized = True
    END
   OF EVENT:GainFocus
    ForceRefresh = True
    IF NOT WindowInitialized
     DO InitializeWindow
     WindowInitialized = True
    ELSE
     DO RefreshWindow
    END
   OF Event:Rejected
    BEEP
    DISPLAY(?)
   ELSE
    IF EVENT() = Event:Timer
      AppFrame{Prop:StatusText,3} = CLIP(DisplayDayText[(TODAY()%7)+1]) & ', '
& DAY(TODAY()) & ' de ' & CLIP(DisplayMonthText[MONTH(TODAY())]) & ' de '
& YEAR(TODAY())
      AppFrame{Prop:StatusText,4} = FORMAT(CLOCK(),@T3)
    END
   END
   CASE ACCEPTED()
   OF ?FileInicialização
    DO SyncWindow
    START(UpdInicializacao,25000)
    LocalRequest = OriginalRequest
    DO RefreshWindow
   OF ?ArquivoConfigurador
    DO SyncWindow
    RUN('configur.exe ')
    ForceRefresh = True
    LocalRequest = OriginalRequest
```

```
    DO RefreshWindow
  OF ?PrintSetup
    DO SyncWindow
  OF ?Exit
    DO SyncWindow
  OF ?Cut
    DO SyncWindow
  OF ?Copy
    DO SyncWindow
  OF ?Paste
    DO SyncWindow
  OF ?Extrair
    DO SyncWindow
    START(ExtraiDados,25000)
    LocalRequest = OriginalRequest
    DO RefreshWindow
  OF ?Tile
    DO SyncWindow
  OF ?Cascade
    DO SyncWindow
  OF ?Arrange
    DO SyncWindow
  OF ?ArqdeSaída
    DO SyncWindow
    START(VerSaida,25000)
    LocalRequest = OriginalRequest
    DO RefreshWindow
  OF ?Helpindex
    DO SyncWindow
  OF ?HelpSearch
    DO SyncWindow
  OF ?HelpOnHelp
    DO SyncWindow
  OF ?AjudaSobre
    DO SyncWindow
    START(sobre,25000)
    LocalRequest = OriginalRequest
    DO RefreshWindow
  END
  CASE FIELD()
  OF ?Button1
   CASE EVENT()
   OF EVENT:Accepted
     DO SyncWindow
     RUN('configur.exe ')
     ForceRefresh = True
     LocalRequest = OriginalRequest
     DO RefreshWindow
   END
```

```
    OF ?Button2
      CASE EVENT()
      OF EVENT:Accepted
        DO SyncWindow
        START(ExtraiDados,25000)
        LocalRequest = OriginalRequest
        DO RefreshWindow
      END
    OF ?Button5
      CASE EVENT()
      OF EVENT:Accepted
        DO SyncWindow
        START(VerSaida,25000)
        LocalRequest = OriginalRequest
        DO RefreshWindow
      END
    OF ?Button4
      CASE EVENT()
      OF EVENT:Accepted
        DO SyncWindow
      END
    OF ?Button3
      CASE EVENT()
      OF EVENT:Accepted
        DO SyncWindow
        START(UpdInicializacao,25000)
        LocalRequest = OriginalRequest
        DO RefreshWindow
      END
    END
  END
  DO ProcedureReturn
!---------------------------------------------------------------------------
ProcedureReturn ROUTINE
  IF WindowOpened
    INISaveWindow('Main','Extrator.INI')
    CLOSE(AppFrame)
  END
  IF LocalResponse
    GlobalResponse = LocalResponse
  ELSE
    GlobalResponse = RequestCancelled
  END
  POPBIND
  RETURN
!---------------------------------------------------------------------------
InitializeWindow ROUTINE
  DO RefreshWindow
!---------------------------------------------------------------------------
```

```
RefreshWindow ROUTINE
  IF AppFrame{Prop:AcceptAll} THEN EXIT.
  DISPLAY()
  ForceRefresh = False
!---------------------------------------------------------------------------
SyncWindow ROUTINE
!---------------------------------------------------------------------------
```

- **Arquivo de Saída para Fonte Clarion**

```
"EXTRA001.CLW","Modulo","'EXTRATOR.clw')      ! This is a MEMBER module"
"EXTRA001.CLW","Rotinas/Proced./Funções","PROCEDURE"
"EXTRA001.CLW","Rotinas/Proced./Funções","Main"
"EXTRA001.CLW","Comentario","-------------------------------------------------------------
-----------"
"EXTRA001.CLW","Rotinas/Proced./Funções","ROUTINE"
"EXTRA001.CLW","Rotinas/Proced./Funções","ProcedureReturn"
"EXTRA001.CLW","Comentario","-------------------------------------------------------------
-----------"
"EXTRA001.CLW","Rotinas/Proced./Funções","ROUTINE"
"EXTRA001.CLW","Rotinas/Proced./Funções","InitializeWindow"
"EXTRA001.CLW","Comentario","-------------------------------------------------------------
-----------"
"EXTRA001.CLW","Rotinas/Proced./Funções","ROUTINE"
"EXTRA001.CLW","Rotinas/Proced./Funções","RefreshWindow"
"EXTRA001.CLW","Comentario","-------------------------------------------------------------
-----------"
"EXTRA001.CLW","Rotinas/Proced./Funções","ROUTINE"
"EXTRA001.CLW","Rotinas/Proced./Funções","SyncWindow"
"EXTRA001.CLW","Comentario","-------------------------------------------------------------
-----------"
```

Bibliografia

[ALC 95] ALCÂNTARA, Eurípedes. A Rede que Abraça Todo o Planeta. **Veja**, São Paulo, v. 28, n. 9, p. 48-54, mar. 1995.

[BER 94] BERTOLETTI, Ana Carolina; DENARDIN, Fernanda K. **SIGEI - Sistema Gerador de Interface**. Porto Alegre: PUCRS, nov. 1994. Trabalho de Conclusão.

[BOW 94] BOWMAN, C. Mic et al. **Scalable Internet Resource Discovery: Research Problems and Approaches**. Disponível por ftp://ftp.cs.colorado.edu/pub/cs/techports/bowman/doc. (fev. 1994).

[COF 95] COFFE, P. **Clarion: A Turbocharged 18-wheeler**. Disponível por http://www.topspeed.com. (nov. 1995).

[COW 96] COWIE, Jim; LEHNERT, Wendy. Information Extraction. **Communications of the ACM**, New York, v. 39. n. 1, p. 80-91, Jan. 1996.

[CRO 93] CROFT, Bruce W. Knowledge-Based and Statistical Approaches to text Retrieval. **IEEE Expert**, Los Alamitos, v. 8, n. 2, p. 8-12, Apr. 1993.

[DVO 96] DVORAK, John C. Uma Explosão Chamada Internet. **Informática Exame**, São Paulo, v. 11, n. 126, p. 13, set. 1996.

[EDE 93] EDELWEISS, Nina; PALAZZO, José M., PERNICI, B. An Object-Oriented Temporal Model. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION SYSTEMS ENGINEERING - CAISE , 5., 1993, Paris, France. **Proceedings...** Berlin: Springer-Verlag, 1993. p.397-415. (Lecture Notes in Computer Science, n. 685).

[EDE 94] EDELWEISS, Nina; PALAZZO, José M. **Modelagem de Aspectos Temporais de Sistemas de Informação**. Recife: UFPE, 1994. 163 p.

[EDE 94a] EDELWEISS, Nina; PALAZZO, José M.; PERNICI, B. An Object-Oriented Approach to a Temporal Query Language, In: DEXA CONFERENCE, 1994, Greece. **Proceedings...** Berlin: Springer-Verlag, 1994. p. 223-235. (Lecture Notes in Computer Science, n° 856)

[EMT 91] EMTAGE, Alan; DEUTSCH, Peter. **The Archie Server**. Montréal, Canada: McGill University, Computer Science Department, 1991. Technical Report.

[EMT 92] EMTAGE, Alan; DEUTSCH, Peter. Archie - An Eletronic Directory Service for the Internet. In: USENIX WINTER CONFERENCE, 1992, San Fracisco, California. **Proceedings...** Montréal: McGill, 1992. p. 93-110.

[FER 94] FERNANDEZ, Luís Fernando Nunes. **SDIP : Um ambiente Inteligente para a localização de informações na Internet**. Porto Alegre: CPGCC da UFRGS, 1994. 238 p. Dissertação de Mestrado.

[FRA 92] FRAWLEY, W. J. et al. **Knowledge Discovery in Database**: An Overview. Disponível por HTTP://alf.usask.ca/homepages/grads/cco410/ 826/summaries/sdmdd3.html. (jun. 1995).

[FRA 93] FRAINER, Antônio S. **Planos na Interação Homem-Máquina**. Porto Alegre: CPGCC da UFRGS, 1993. Dissertação Mestrado.

[FUL 94] FULLTON, M. J. et al. **WAIS over Z39.50-1988**. [S.l.:s.n.], 1994. 7p.

[GEN 86] GENARO, Sérgio. **Sistemas Especialistas:** O Conhecimento Artificial. Rio de Janeiro: LTC Livros Técnicos e Científicos, 1986. 192 p.

[GIF 91] GIFFORD, D. R. et al. **Semantic File System**. Disponível por gifford@lcs.mit.edu. (maio 1995).

[HAR 93] HARDY, Darren R.; SCHWARTZ, Michael F. Essence: A Resource Discovery System Based on Semantic File Indexing. In: USENIX WINTER CONFERENCE, 1993, San Diego, California. **Proceedings...** Boulder: University of Colorado, 1993. p. 361-374.

[HAR 95] HARDY, Darren R.; SCHWARTZ, Michael F. **Customized Information Extraction as Basis for Resource Discovery**. Disponível por http://www.cs.colorado.edu/homes/schwartz/public_html/resource_location.html. (abr. 1996).

[HED 95] HEDBERG, Sara R. The Data Gold Rush. **Byte**, Peterborough, N. H., v. 20, n. 10, p. 83-88, Oct. 1995.

[HEL 96] HELLER, Martin. **Reviews Head to Head**: Client/Server Development. Disponível por http://www.topspeed.com. (set. 1996).

[HOW 88] HOWARD, J. et al. Scale and Performance in Distribute File Systems. **ACM Transactions on Computer Systems**, New York, v. 6. n. 1, p. 51-81, Feb. 1988.

[KAH 91] KAHLE, Brewster. **An Information System for Corporate Users**: Wide Area Information Servers. Disponível por http://www.sbexpos.com/sbexpos/interop/connexions/v5_index.html. (set. 1995).

[LAN 68] LANCASTER, F. Wilfrid. **Information Retrivel Systems - Characteristics, Testing end Evaluation**. New York : John Wiley & Sons Inc., 1968.

[LEH 96] LEHNERT, Wendy. **Information Extraction**. Computer Science Department at the University of Massachusetts. Disponível por http://www-nlp.cs.umass.edu/ ~nlpgroup/nlpie.html. (out. 1996).

[LEV 88] LEVINE, Robert I.; DRANG, Diane E.; EDELSON Barry. **Inteligência Artificial e Sistemas Especialistas**. São Paulo: McGraw-Hill, 1988.

[LEW 96] LEWIS, David D.; JONES, Karen S.. Natural Language Processing for Information Retrieval. **Communications of the ACM**, New York, v. 39. n. 1, p. 92-101, Jan. 1996.

[MAR 92] MARSHALL, Peter. **WAIS: The Wide Area Information Server or Anonymous What ???**. The University of Western Ontario, June 92. Disponível por ftp://ftp.think.com/wais/ bibliography. (jun. 1992).

[MAT 93] MATHEUS, Christopher et al. **KDD-93: Progress and Challenges in Knowledge Discovery in Databases**. Disponível por http://info.gte.com/~kdd. (jul. 1995).

[MES 96] MESROBIAN, E.; SHEK, E. C.; MUNTZ, R. R. On Heterogeneous Distributed Geocientific Query Processing. In: INTERNATIONAL WORKSHOP ON RESEARSH ISSUES IN DATA ENGINEERING: INTEROPERABILITY OF NONTRADITIONAL DATABASES SYSTEMS, 1996, New Orleans. **Proceedings...** Disponível por http://nitro.cs.ucla.edu/publications/ #ride_nds_96_conquest. (mar. 1996).

[MET 89] METZLER, Douglas P.; HAAS, Stephanie W. The Constituent Object Parser: Syntactic Structure Matching for Information Retrieval. **ACM Transactions on Information Systems**, New York, v. 7, n. 3, p. 292-316, July 1989.

[MIC 82] MICHIE, Donald. **Introdutory Readings in Expert Systems**. New York: Gordon and Breach, Science, 1982.

[MIC 95] MICROSOFT CORPORATION. **Obtendo Resultados com o Microsoft Word para Windows 95**. Redmond: Microsoft Corporation, 1995.

[MIC 95a] MICROSOFT CORPORATION. **Microsoft Windows 95**. Redmond: Microsoft Corporation, 1995.

[MOU 92] MOULIN, Bernard; ROUSSEAU, Daniel. Automated Knowledge Acquisition from Regulatory Texts. **IEEE Expert**, Los Alamitos, v. 7, n. 2, p. 27-35, Oct. 1992.

[MUC 96] MUC-6, **The Sixth in a Series of Message Understanding Conferences**. Disponível por http://cs.nyu.edu/cs/faculty/grishman/muc6.html. (out. 1996).

[OVE 95] **Overview of Information Extraction Task**. Disponível por http://cs.nyu.edu/cs/faculty/grishman/IEtask15.book_2.html. (abr. 1996).

[PER 96] PERSON, Ron; et al. **Usando Windows 95**. Rio de Janeiro: Campus, 1996. 1388 p.

[PET 92] PETZOLD, Charles. **Programando para Windows.** São Paulo: Makron Books, 1993. p. 1034.

[RIC 93] RICH, Elaine; KNIGHT, Kevin. **Inteligência Artificial**. São Paulo: Makron Books, 1993.

[SAL 68] SALTON, Gerard. **Automatic Information Organization and Retrieval**. New York: McGraw-Hill, 1968.

[SAL 86] SALTON, Gerard. Another look at Automatic Text-Retrieval Systems. **Communications of ACM**, New York, v. 29, n.7, p. 648-656, July 1986.

[SAL 87] SALTON, Gerard; BUCKLEY, Chirs. **Term Weighting Approaches in Automatic Text Retrieval.** New York: Cornell University, 1987. 21 p. (Techinical Report 87-881).

[SAL 88] SALTON, Gerard; SMITH, Maria. **On the Application of Syntatic Methodologies in Automatic Text Analysis**. New York: Cornell University , 1988. 20 p. (Techinical Report 88-954).

[SAL 88a] SALTON, Gerard; BUCKLEY, Chirs. **Improving Retrieval Performance by Relevance Feedback.** New York: Cornell University, 1988. 24 p. (Techinical Report 88-898).

[SAR 95] SARAEE, M. H.; Theodoulidis, C. **Knowledge Discovery in Temporal Databases**. Disponível por http://www.co.umist.ac.uk/~timilab/ projects/tempominer.html. (maio 1996).

[SCH 92] SCHWARTZ, Michael F.. **Internet Resource Discovery at the University of Colorado**. Disponível por http://www.cs.colorado. edu/homes/schwartz/public_html/resource_location.html. (abr. 1995).

[SHA 94] SHAPIRO, Gregory P. et al. **Systems for Knowledge Discovery in Database**. Disponível por http://info.gte.com/~kdd. (jul. 1995).

[SNO 85] SNODGRASS, R; AHN, I. A Taxonomy of time in databases. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 1985, Texas. **Proceedings...** [S.l. : s.n.], 1985. p.28-31.

[TAN 93] TANSEL, A.U. et al. **Temporal Databases - Theory, Design, and Implementation**. Redwood City: Benjamin/Cummings, 1993. 633p.

[TOP 95] TOPSPEED CORPORATION. **Clarion for Windows TM**: Language Reference. Pompano Beach, Florida: TopSpeed Corporation, 1995.

[TOP 95a] TOPSPEED CORPORATION. **Clarion for Windows TM**: Users' Guide. Pompano Beach, Florida: TopSpeed Corporation: 1995.

[TOP 95b] TOPSPEED CORPORATION. **Clarion for Windows TM**: Getting Started. Pompano Beach, Florida: TopSpeed Corporation, 1995.

[TUR 91] TURTLE, Howard; CROFT, Bruce W. Evaluation of an Inference Network Based Retrieval Model. **ACM Transactions on Information Systems**, New York, v. 9, n. 3, p. 187-222, July 1991.

[WEB 96] WEBBER, Carine G. **Estudo sobre Recuperação de Informações e Bibliotecas Digitais**. Porto Alegre: CPGCC da UFRGS, 1996. 91 p. Trabalho Individual.

[WEI 88] WEISS, Sholom M.. **Guia Prático para Projetar Sistemas Especialistas**. Rio de Janeiro: LTC Livros Técnicos e Científicos, 1988. 169 p.

[WOR 93] WORDPERFECT CORPORATION. **WordPerfect for Windows Versão 6.0**. WordPerfect Corporation, 1993.

[WYA 95] WYATT, Allen L. **Sucesso com Internet**. São Paulo: Érica, 1995. 404 p.

Informática *UFRGS*

CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

SES - Sistema de Extração Semântica de Informações.

por

Rui Gureghian Scarinci

Dissertação apresentada aos Senhores:

Prof. Dr. Alberto Henrique Frade Laender (UFMG)

Prof. Dr. Clesio Saraiva dos Santos

Prof. Dr. Raul Fernando Weber

Vista e permitida a impressão.
Porto Alegre, 30/04/97.

Prof. Dr. José Palazzo Moreira de Oliveira,
Orientador.

Prof. Flávio Rech Wagner
Coordenador do Curso de Pós-Graduação em
Ciência da Computação
Instituto de Informática - UFRGS